

Identifying and Mitigating Security Risks in Multi-Level Systems-of-Systems Environments

Kirsty E. Lever BSc (Hons), AFHEA

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

November 2017

This dissertation is dedicated to the memory of my late grandmother Elizabeth, and to my grandfather William Lockett, for their love and support throughout my life, for their encouragement to continuously learn, and for introducing me to the joys and frustrations of computers as a child which established a deep lifelong passion within me.

Acknowledgments

Firstly, I would like to thank my Director of Studies, Dr Kashif Kifayat for his support and guidance throughout the duration of my studies. Kashif always believed in me as a student and saw the potential in me to be a great academic researcher which I failed to perceive, he has remained an influential part of my academic development, for which I am truly thankful. In addition, I would like to express my gratitude to the other member of my supervisory team Dr Bo Zhou, and former advisors Prof. Madjid Merabti and Dr David Llewellyn-Jones. Their support and astuteness guided and influenced my research path, which greatly contributed to its successful completion.

Eternal thanks must be given to my close family and friends. While I have been pursuing my goals, the unconditional love, support, advice and reassurance has been enormous. Special thanks to my immediate family Sue, Chris and Tony Lever, they have had to endure excessive 'geek technical babble', along with the stress of supporting me. I do not deserve such a wonderful sacrificing and patient family who went out of their way to encourage and ensure that this thesis was completed. In addition I would like to give many thanks to all the SUEvivor and PHAB Bebington members who have adopted me as part of their wider family, and who have supported me and my spouse, enabling in part for me to work and complete my research.

I would like to give thanks to the Liverpool John Moores University, Department of Computer Science, for giving me the opportunity to undertake this research degree and for their encouragement and support throughout my studies. This opportunity has allowed me to progress my academic career, by greatly enhancing my skill set.

Finally, I would also like to thank all my friends and colleagues at Liverpool John Moores for their instrumental counsel. I particularly appreciate Áine MacDermott and Paul Buck who were simply awesome with the never-ending support and conversations which made my research more enjoyable, thanks also to, David Tully, Mark Sabino, Tricia Waterson, Carol Oliver, and Lucy Tweedle.

Kirsty E. Lever, Liverpool John Moores, UK, November 2017

Abstract

In recent years, organisations, governments, and cities have taken advantage of the many benefits and automated processes Information and Communication Technology (ICT) offers, evolving their existing systems and infrastructures into highly connected and complex Systems-of-Systems (SoS). These infrastructures endeavour to increase robustness and offer some resilience against single points of failure. The Internet, Wireless Sensor Networks, the Internet of Things, critical infrastructures, the human body, etc., can all be broadly categorised as SoS, as they encompass a wide range of differing systems that collaborate to fulfil objectives that the distinct systems could not fulfil on their own.

ICT constructed SoS face the same dangers, limitations, and challenges as those of traditional cyber based networks, and while monitoring the security of small networks can be difficult, the dynamic nature, size, and complexity of SoS makes securing these infrastructures more taxing. Solutions that attempt to identify risks, vulnerabilities, and model the topologies of SoS have failed to evolve at the same pace as SoS adoption. This has resulted in attacks against these infrastructures gaining prevalence, as unidentified vulnerabilities and exploits provide unguarded opportunities for attackers to exploit. In addition, the new collaborative relations introduce new cyber interdependencies, unforeseen cascading failures, and increase complexity.

This thesis presents an innovative approach to identifying, mitigating risks, and securing SoS environments. Our security framework incorporates a number of novel techniques, which allows us to calculate the security level of the entire SoS infrastructure using vulnerability analysis, node property aspects, topology data, and other factors, and to improve and mitigate risks without adding additional resources into the SoS infrastructure. Other risk factors we examine include risks associated with different properties, and the likelihood of violating access control requirements. Extending the principals of the framework, we also apply the approach to multi-level SoS, in order to improve both SoS security and the overall robustness of the network. In addition, the identified risks, vulnerabilities, and interdependent links are modelled by extending network modelling and attack graph generation methods.

The proposed SeCurity Risk Analysis and Mitigation Framework and principal techniques have been researched, developed, implemented, and then evaluated via numerous experiments and case studies. The subsequent results accomplished ascertain that the framework can successfully observe SoS and produce an accurate security level for the entire SoS in all instances, visualising identified vulnerabilities, interdependencies, high risk nodes, data access violations, and security grades in a series of reports and undirected graphs. The framework's evolutionary approach to mitigating risks and the robustness function which can determine the appropriateness of the SoS, revealed promising

results, with the framework and principal techniques identifying SoS topologies, and quantifying their associated security levels. Distinguishing SoS that are either optimally structured (in terms of communication security), or cannot be evolved as the applied processes would negatively impede the security and robustness of the SoS. Likewise, the framework is capable via evolution methods of identifying SoS communication configurations that improve communication security and assure data as it traverses across an unsecure and unencrypted SoS. Reporting enhanced SoS configurations that mitigate risks in a series of undirected graphs and reports that visualise and detail the SoS topology and its vulnerabilities. These reported candidates and optimal solutions improve the security and SoS robustness, and will support the maintenance of acceptable and tolerable low centrality factors, should these recommended configurations be applied to the evaluated SoS infrastructure.

Published work

Some key aspects, ideas, and figures from this thesis have previously appeared in the following publications:

Journal articles

1. Lever, K.E., Kifayat, K., “The Challenge of Quantifying and Modelling Risk Elements within Collaborative Infrastructures,” *International Journal of Risk Assessment and Management (IJRAM)*, vol. 19, no. 3, pp. 167-193, 2016.
2. Lever, K.E., Kifayat, K., “Risk Assessment and Attack Graph Generation for Collaborative Infrastructures: A Survey,” *International Journal of Critical Computer-Based Systems (IJCCBS)*, vol. 6, no. 3, pp. 204-228, 2016.

Conference papers

1. Rehman, O., Yang, S., Khan, S.U., Lever, K., Kifayat, K., “A Global Modified Quantum Particle Swarm Optimizer Applied to Optimization Design of Electromagnetic Devices,” *IEEE Transactions on Magnetics*, 2018 [in review].
2. Lever, K.E., MacDermott, Á., Kifayat, K., “Evaluating Interdependencies and Cascading Failures Using Distributed Attack Graph Generation Methods for Critical Infrastructure Defence,” 2015 International Conference on Developments of E-Systems Engineering (DeSE), Dubai, United Arab Emirates, pp. 47-52, 13-14 Dec. 2015.
3. Lever, K.E., Kifayat, K., Merabti, M., “Identifying interdependencies using attack graph generation methods,” 2015 11th International Conference on Innovations in Information Technology (IIT), Dubai, pp. 80-85, 1-3 Nov. 2015.
4. Lever, K.E., Merabti, M., Kifayat, K., Llewellyn-Jones, D., “Elements of Risk Within Systems-of-Systems,” 15th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet), Liverpool, UK, pp.225-230, Jun. 2014.
5. Lever, K.E., Merabti, M., Kifayat, K., “Single Points of Failure Within Systems-of-Systems,” 14th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet), Liverpool, UK, pp. 183-188, 24-25 Jun. 2013.

Book Chapters

1. Lever, K.E., Kifayat, K., and Merabti, M., “Identifying Interdependencies Using Simulation and Attack Graph Generation Methods,” Springer [in press].

Contents

Chapter 1 Introduction	1
1.1 Foreword.....	1
1.1.1 General Issues Within Systems-of-Systems	3
1.1.2 Security Issues Within Systems-of-Systems	4
1.1.3 Risk Analysis Within Systems-of-Systems.....	5
1.2 Motivation and Research Gaps	7
1.3 Aims and Objectives	9
1.4 Novel Contributions.....	10
1.5 Research Findings	11
1.6 Thesis Structure	12
Chapter 2 Background	15
2.1 Systems-of-Systems	16
2.1.1 Systems-of-Systems Types	17
2.1.2 Systems-of-Systems Examples	18
2.1.2.1 Critical Infrastructures	19
2.1.2.2 Smart Cities.....	21
2.1.2.3 The Human Body	23
2.1.3 Systems-of-Systems Associated Rewards	25
2.2 Systems-of-Systems Challenges	26
2.2.1 Systems-of-Systems Associated Characteristic Challenges.....	26
2.2.1.1 Operational Independence.....	26
2.2.1.2 Managerial Independence	27
2.2.1.3 Evolutionary Development	28
2.2.1.4 Emergent Behaviour	28
2.2.1.5 Geographic Behaviour	29
2.2.2 Single Points of Failure Within Systems-of-Systems	30

2.3 Systems-of-Systems Security Challenges	32
2.3.1 Factors Which Can Impact Security	34
2.3.1.1 Interdependency	34
2.3.1.2 Complexity.....	37
2.3.1.3 Cascading Failure.....	39
2.3.1.4 Identified Systems-of-Systems Risks and Attacks.....	42
2.4 Systems-of-Systems Risk Analysis.....	46
2.4.1 Risk Assessment and Management.....	47
2.5 Systems-of-Systems Vulnerability Analysis.....	50
2.5.1 Network Vulnerabilities.....	51
2.5.2 Network Vulnerability Scanners.....	54
2.5.3 Vulnerability Analysis, Scoring, and Exploit Databases	56
2.6 Network Security Systems	57
2.6.1 Attack Graph Generation	57
2.6.1.1 Multi-Host Multi-Stage Vulnerability Analysis.....	59
2.6.1.2 Network Security Planning Architecture	60
2.6.1.3 Topological Analysis of Network Attack Vulnerability	61
2.6.2 Network Intrusion Detection Systems and Analysers.....	62
2.7 Summary	63
Chapter 3 Related Work.....	65
3.1 Systems-of-Systems Security.....	65
3.1.1 Securing SoS Against Malicious Attacks	66
3.1.2 Intrusion Detection and Prevention Methods.....	70
3.1.3 Securing SoS During the Development Life Cycle	72
3.1.4 Securing SoS Using Network Security Solutions	74
3.2 Risk Analysis	76
3.2.1 Risk Analysis Based Techniques	76
3.2.2 Risk Analysis: Lab Based Risk Reduction	81

3.2.3 Prevention and Detection of Single Points of Failure	83
3.2.4 Prevention and Detection of Cascading Failures	87
3.2.5 Detecting Interdependence.....	90
3.2.6 Detecting Complexity	93
3.2.7 Detecting Emergent Behaviour.....	95
3.3 Risk Management and Assessment.....	96
3.4 Risk Modelling.....	101
3.4.1 Network Modelling.....	101
3.4.2 Attack Graph Generation	103
3.5 Information Assurance.....	105
3.6 Network Optimisation.....	109
3.7 Summary of Existing Methodologies	110
3.8 Summary.....	113
Chapter 4 SeCurity Risk Analysis and Mitigation (SCRAM) Framework.....	115
4.1 Problem Analysis	116
4.1.1 Aims Analysis.....	117
4.1.2 Objectives Analysis	120
4.1.2.1 Background Literature Research.....	120
4.1.2.2 Security Risk Analysis.....	121
4.1.2.3 Robustness Analysis	122
4.1.2.4 Optimisation Evaluation	123
4.1.2.5 Case Study	124
4.1.2.6 Multi-Level SoS Analysis.....	125
4.1.3 Methodology	125
4.1.3.1 Research Method	126
4.1.3.2 Research Design.....	127
4.1.3.3 Data Collection Considerations	129
4.1.3.4 Data Analysis.....	131

4.1.3.5 Problems and Limitations	132
4.1.3.6 Method Summary.....	133
4.2 SeCurity Risk Analysis and Mitigation Framework	133
4.3 SeCurity Risk Analysis and Mitigation Framework Design Overview	137
4.4 SeCurity Risk Analysis and Mitigation Framework Operation	140
4.5 Data Access Control Problem and Management	144
4.6 Topological Vulnerabilities.....	148
4.6.1 High Connectivity Vulnerability.....	149
4.6.2 Shortest Path Vulnerability	150
4.6.3 Single Points of Failure.....	151
4.6.4 Weighted High Connectivity	152
4.6.5 Dependent Communication Vulnerability	153
4.7 Security Enhancement and Risk Mitigation Techniques	153
4.7.1 Node Security Grade Assignment.....	154
4.7.1.1 Vulnerability Analysis	154
4.7.1.2 Common Vulnerability Scoring System (CVSS).....	155
4.7.1.3 National Vulnerability Database (NVD).....	157
4.7.1.4 Calculating Node Security Grades	159
4.7.2 Network Data Flow Security Level	161
4.6.3 Robustness Function	162
4.7.4 Risk Mitigation	163
4.7.4.1 Genetic Algorithm	163
4.7.4.2 Ant Colony Optimisation Combined with Local Search	165
4.7.4.3 Tabu Search	167
4.8 Summary	168
Chapter 5 Implementation and Evaluations	170
5.1 SCRAM Framework	170
5.1.1 Network Generation.....	172

5.1.2 Topological Vulnerabilities Analysis.....	176
5.1.3 SCRAM Framework Cycle Analysis.....	177
5.1.4 Applied Network Security Risk Mitigation Principals	180
5.1.4.1 Genetic Algorithm Evaluation	180
5.1.4.2 Ant Colony Optimisation Combined with Local Search Evaluation	183
5.1.4.3 Tabu Search Optimisation Evaluation	185
5.1.4.4 Network Security Enhancement and Risk Mitigation Evaluation	186
5.1.5 Evaluating Dynamic Systems-of-Systems	189
5.1.6 Effectiveness of Simulated SCRAM Framework	197
5.2 Effectiveness of Integrating Vulnerability Identification	199
5.3 Effectiveness of Risk Mitigation Within Secure Networks	208
5.3.1 Evaluating Positive Security Risk Mitigation of Secure SoS	212
5.3.2 Evaluating Negative Security Risk Mitigation of Secure SoS	218
5.4 Case Study	221
5.4.1 Node Energy Efficiency Problem	221
5.4.2 Node Energy Efficiency Comparison	222
5.4.3 Smart City WSN Robustness	227
5.4.4 Smart City WSN Data Analysis.....	228
5.4.5 Smart City WSN Observations	232
5.4.6 Smart City Sectors Simulation Evaluation.....	234
5.5 Summary	238
Chapter 6 Multi-Level SoS Security Analysis and Evaluation	239
6.1 Multi-Level SoS Security Challenges.....	239
6.1.1 Multi-Level SoS: Calculating Connecting Node Security Grades.....	241
6.1.2 Multi-Level SoS: Connecting Node Security Analysis	242
6.1.3 Multi-Level SoS: Calculating Connecting Node Security Grades.....	246
6.1.4 Multi-Level SoS: Security Evaluation	249
6.1.5 Multi-Level SoS: Case Study.....	252

6.2 Multi-Level SoS Evaluation.....	256
6.2.1 Multi-level SoS: SCRAM Evaluation.....	257
6.2.1.1 SCRAM Positive Multi-Level SoS Vulnerability Performance	258
6.2.1.2 SCRAM Positive Multi-Level SoS Vulnerability and Data Access Performance.....	265
6.2.1.3 SCRAM Negative Multi-Level SoS Vulnerability Performance.....	273
6.2.1.4 SCRAM Negative Multi-Level SoS Vulnerability and Data Access Performance	278
6.3 Summary	283
Chapter 7 Conclusion and Future Work	284
7.1 Thesis Summary.....	285
7.2 Aims and Objectives Evaluation.....	286
7.3 Novel Contributions and Publications	288
7.4 Limitations	290
7.5 Future Work.....	292
7.6 Concluding Remarks.....	294
References.....	297
Appendix A SCRAM Positive Multi-Level SoS Vulnerability Performance.....	318
Appendix B SCRAM Positive Multi-Level SoS Vulnerability and Data Access Performance	336
Appendix C SCRAM Negative Multi-Level SoS Vulnerability Performance	354
Appendix D SCRAM Negative Multi-Level SoS Vulnerability Performance	366

List of Figures

Figure 1.1 Schematic Representation of SoS and Multi-Level SoS Classifications	1
Figure 2.1. Cyber Systems-of-Systems Architecture.....	15
Figure 2.2. Critical Infrastructure Example Architecture	20
Figure 2.3. Smart City Topology Schematic.....	22
Figure 2.4. Human Systems Topology.....	23
Figure 2.5. Schematic Representation of Single Points of Failure	30
Figure 2.6. Schematic Representation of Interdependent Relations in a Smart City	37
Figure 2.7. Complex Multi-Level Systems-of-Systems.....	38
Figure 2.8. Schematic Representation of Cascading Failures.....	40
Figure 2.9. ISO 31000:2009 Risk Management Principles, Framework, and Processes	48
Figure 2.10. Cross-Site Scripting Attack	52
Figure 2.11. Overview of Required Network Based Vulnerability Scanner Components.....	54
Figure 2.12. An Overview of the Identified Risks Visualised in Graph Form	58
Figure 2.13. MulVAL Framework.....	60
Figure 2.14. NetSPA Framework.....	61
Figure 2.15. TVA Framework.....	62
Figure 3.1. Cyber Security Evaluation Tool Assessment Process	99
Figure 4.1. SeCurity Risk Analysis and Mitigation Framework Positioning	135
Figure 4.2. Illustrated Overview of the SCRAM Framework.....	137
Figure 4.3. SeCurity Risk Analysis and Mitigation Framework Execution Flowchart	141
Figure 4.4. Composed Data Access Control Scenario of a Smart City.....	145
Figure 4.5. Composed Smart City Consisting of Sensitivity Levels and Data Flow Risk.....	148
Figure 4.6. Schematic Representation of Degree Centrality.....	149
Figure 4.7. Schematic Representation of Betweenness Centrality	150
Figure 4.8. Schematic Representation of Closeness Centrality	151
Figure 4.9. Schematic Representation of Eigenvector Centrality	152

Figure 4.10. Schematic Representation of Bridging Centrality	153
Figure 4.11. Overview of the CVSS v3.0 Metric Groups	155
Figure 5.1. SCRAM Screenshot.....	171
Figure 5.2. Primary Test SoS Network Configuration File	173
Figure 5.3. Primary Simulated Network Environment	174
Figure 5.4. Primary Simulated Network Environment User Interface Report.....	175
Figure 5.5. Code Excerpt Showing Degree Centrality Method	176
Figure 5.6. Analysis of the Network Evolutionary Process	178
Figure 5.7. Robustness Monitor for the Applied Genetic Algorithm	181
Figure 5.8. Comparison of Genetic Algorithm Improved Security Solutions	182
Figure 5.9. Comparison of ANT Algorithm Improved Security Solutions.....	183
Figure 5.10. Robustness Monitor for the Applied ANT Algorithm.....	184
Figure 5.11. SCRAM Output Window Identifying ANT Best Solutions	185
Figure 5.12. Comparison of Tabu Algorithms Improved Solutions	186
Figure 5.13. Comparison of the Applied Risk Mitigation Algorithms	188
Figure 5.14. Modified Topologies of the Primary Network	190
Figure 5.15. Evaluating Modified Network A Reconfiguration	191
Figure 5.16. Evaluating Modified Network B Failure and Reconfiguration	192
Figure 5.17. Evolution Analysis of Modified Primary Network Reconfiguration.....	194
Figure 5.18. Evolution Analysis of Modified Primary Network Failure and Reconfiguration	196
Figure 5.19. Network Comparison of Assigned Security and Simulated Vulnerabilities.....	200
Figure 5.20. Evaluating Network Vulnerability Identification	205
Figure 5.21. Evolution Analysis of Network Vulnerability Technique Comparison.....	207
Figure 5.22. Evaluating a Secure System-of-Systems	209
Figure 5.23. Analysis of the Secure Network When GA was Applied.....	211
Figure 5.24. Analysis of Node Centralities of Secure Network When GA was Applied	212
Figure 5.25. Secure 8 Node Network Comparisons with Applied GA Risk Mitigation.....	213
Figure 5.26. Secure 12 Node Network Comparisons with Applied GA Risk Mitigation.....	215

Figure 5.27. Security Analysis of Secure Networks When GA is Applied to Mitigate Risks	217
Figure 5.28. Comparison of Secure Networks Evolved that Negatively Impact Network Cost.....	218
Figure 5.29. Security Analysis of Secure Networks When GA is Applied with Negative Outcomes	220
Figure 5.30. Primary Simulated Smart City WSN Environment	223
Figure 5.31. Comparison of Smart City WSN Security Risk Mitigation.....	225
Figure 5.32. Comparison of Smart City WSN Energy Level and Security Risk Mitigation	226
Figure 5.33. Comparison of Smart City WSN Robustness Graphs	227
Figure 5.34. Network Evolution Security Results Comparison for WSN A.....	229
Figure 5.35. Network Evolution Security Results Comparison for WSN B.....	230
Figure 5.36. Simulated Smart City Networks	235
Figure 5.37. Simulated Smart City Sector Robustness and Security Comparison.....	236
Figure 6.1. Reconfigured Secure Simulated Smart City Networks	240
Figure 6.2. Simulated Smart City Networks	245
Figure 6.3. A Multi-Level SoS Example	246
Figure 6.4. Multi-Level SoS Primary Connecting Node Scores.....	247
Figure 6.5. Reconfigured Multi-Level SoS Connecting Node Scores	248
Figure 6.6. Multi-Level SoS Topological Security Vulnerabilities and Robustness Comparison.....	250
Figure 6.7. Simulated Multi-Level SoS	252
Figure 6.8. Simulated Multi-Level SoS Primary Connecting Node Scores.....	253
Figure 6.9. Security Enhanced Simulated Multi-Level SoS Connecting Node Scores.....	254
Figure 6.10. Multi-Level SoS Topological Security Vulnerabilities and Robustness Comparison...	256
Figure 6.11. Set A of Multi-Level SoS Used in the Experiments (see Appendix A)	260
Figure 6.12. Set B of Multi-Level SoS Used in the Experiments (see Appendix A).....	261
Figure 6.13. Set C of Multi-Level SoS Used in the Experiments (see Appendix A).....	262
Figure 6.14. Set A Multi-Level SoS Security Vulnerabilities and Robustness Comparison	263
Figure 6.15. Set B Multi-SoS Topological Security Vulnerabilities and Robustness Comparison ...	264
Figure 6.16. Set C Multi-SoS Topological Security Vulnerabilities and Robustness Comparison ...	265
Figure 6.17. Set D of Multi-Level SoS Used in the Experiments (see Appendix B).....	267

Figure 6.18. Set E of Multi-Level SoS Used in the Experiments (see Appendix B)	268
Figure 6.19. Set F of Multi-Level SoS Used in the Experiments (see Appendix B)	269
Figure 6.20. Set D Multi-SoS Topological Security Vulnerabilities and Robustness Comparison ...	270
Figure 6.21. Set E Multi-SoS Topological Security Vulnerabilities and Robustness Comparison ...	271
Figure 6.22. Set F Multi-SoS Topological Security Vulnerabilities and Robustness Comparison....	272
Figure 6.23. Set G of Multi-Level SoS Used in the Experiments (see Appendix C).....	273
Figure 6.24. Set H of Multi-level SoS Used in the Experiments (see Appendix C)	274
Figure 6.25. Set I of Multi-Level SoS Used in the Experiments (see Appendix C)	275
Figure 6.26. Set G-I Multi-Level SoS Topological Security Vulnerabilities Comparison	277
Figure 6.27. Sets G-I Populations Robustness Comparison	277
Figure 6.28. Set J of Multi-Level SoS Used in the Experiments (see Appendix D).....	279
Figure 6.29. Set K of Multi-Level SoS Used in the Experiments (see Appendix D)	280
Figure 6.30. Set L of Multi-Level SoS Used in the Experiments	281
Figure 6.31. Set J-L Multi-Level SoS Topological Security Vulnerabilities Comparison	282
Figure 6.32. Sets J-L Multi-Level SoS Robustness Comparison.....	283
Appendix A Figure 1. Multi-Level SoS A with Node Status	318
Appendix A Figure 2. Multi-Level SoS A Topology	319
Appendix A Figure 3. Multi-Level SoS A Optimum Candidate	319
Appendix A Figure 4. Multi-Level SoS B with Node Status.....	320
Appendix A Figure 5. Multi-Level SoS B Topology.....	321
Appendix A Figure 6. Multi-Level SoS B Optimum Candidate.....	321
Appendix A Figure 7.. Multi-Level SoS C with Node Status.....	322
Appendix A Figure 8. Multi-Level SoS C Topology.....	323
Appendix A Figure 9. Multi-Level SoS C Optimum Candidate.....	323
Appendix A Figure 10. Multi-Level SoS D with Node Status	324
Appendix A Figure 11. Multi-Level SoS D Topology	325
Appendix A Figure 12. Multi-Level SoS D Optimum Candidate	325
Appendix A Figure 13. Multi-Level SoS E with Node Status.....	326

Appendix A Figure 14. Multi-Level SoS E Topology	327
Appendix A Figure 15. Multi-Level SoS E Optimum Candidate	327
Appendix A Figure 16. Multi-Level SoS F with Node Status	328
Appendix A Figure 17. Multi-Level SoS F Topology	329
Appendix A Figure 18. Multi-Level SoS F Optimum Candidate	329
Appendix A Figure 19. Multi-Level SoS G with Node Status	330
Appendix A Figure 20. Multi-Level SoS G Topology	331
Appendix A Figure 21. Multi-Level SoS G Optimum Candidate	331
Appendix A Figure 22. Multi-Level SoS H with Node Status	332
Appendix A Figure 23. Multi-Level SoS H Topology	333
Appendix A Figure 24. Multi-Level SoS H Optimum Candidate	333
Appendix A Figure 25. Multi-Level SoS I with Node Status	334
Appendix A Figure 26. Multi-Level SoS I Topology	335
Appendix A Figure 27. Multi-Level SoS I Optimum Candidate	335
Appendix B Figure 1. Multi-Level SoS J with Node Status	336
Appendix B Figure 2. Multi-Level SoS J Topology	337
Appendix B Figure 3. Multi-Level SoS J Optimum Candidate	337
Appendix B Figure 4. Multi-Level SoS K with Node Status.....	338
Appendix B Figure 5. Multi-Level SoS K Topology.....	339
Appendix B Figure 6. Multi-Level SoS K Optimum Candidate.....	339
Appendix B Figure 7. Multi-Level SoS L with Node Status	340
Appendix B Figure 8. Multi-Level SoS L Topology	341
Appendix B Figure 9. Multi-Level SoS L Optimum Candidate	341
Appendix B Figure 10. Multi-Level SoS M with Node Status	342
Appendix B Figure 11. Multi-Level SoS M Topology	343
Appendix B Figure 12. Multi-Level SoS M Optimum Candidate	343
Appendix B Figure 13. Multi-Level SoS N with Node Status.....	344
Appendix B Figure 14. Multi-Level SoS N Topology.....	345

Appendix B Figure 15. Multi-Level SoS N Optimum Candidate.....	345
Appendix B Figure 16. Multi-Level SoS O with Node Status.....	346
Appendix B Figure 17. Multi-Level SoS O Topology.....	347
Appendix B Figure 18. Multi-Level SoS O Optimum Candidate.....	347
Appendix B Figure 19. Multi-Level SoS P with Node Status	348
Appendix B Figure 20. Multi-Level SoS P Topology	349
Appendix B Figure 21. Multi-Level SoS P Optimum Candidate	349
Appendix B Figure 22. Multi-Level SoS Q with Node Status.....	350
Appendix B Figure 23. Multi-Level SoS Q Topology.....	351
Appendix B Figure 24. Multi-Level SoS Q Optimum Candidate.....	351
Appendix B Figure 25. Multi-Level SoS R with Node Status	352
Appendix B Figure 26. Multi-Level SoS R Topology	353
Appendix B Figure 27. Multi-Level SoS R Optimum Candidate	353
Appendix C Figure 1. Multi-Level SoS S with Node Status	354
Appendix C Figure 2. Multi-Level SoS S Topology	355
Appendix C Figure 3. Multi-Level SoS S Optimum Candidate	355
Appendix C Figure 4. Multi-Level SoS T with Node Status	356
Appendix C Figure 5. Multi-Level SoS T Topology	357
Appendix C Figure 6. Multi-Level SoS T Optimum Candidate	357
Appendix C Figure 7. Multi-Level SoS U with Node Status.....	358
Appendix C Figure 8. Multi-Level SoS U Topology.....	359
Appendix C Figure 9. Multi-Level SoS U Optimum Candidate.....	359
Appendix C Figure 10. Multi-Level SoS V with Node Status.....	360
Appendix C Figure 11. Multi-Level SoS V Topology.....	361
Appendix C Figure 12. Multi-Level SoS V Optimum Candidate.....	361
Appendix C Figure 13. Multi-Level SoS W with Node Status.....	362
Appendix C Figure 14. Multi-Level SoS W Topology.....	363
Appendix C Figure 15. Multi-Level SoS W Optimum Candidate.....	363

Appendix C Figure 16. Multi-Level SoS X with Node Status.....	364
Appendix C Figure 17. Multi-Level SoS X Topology.....	365
Appendix C Figure 18. Multi-Level SoS X Optimum Candidate.....	365
Appendix D Figure 1. Multi-Level SoS Y with Node Status	366
Appendix D Figure 2. Multi-Level SoS Y Topology	367
Appendix D Figure 3. Multi-Level SoS Y Optimum Candidate	367
Appendix D Figure 4. Multi-Level SoS Z with Node Status.....	368
Appendix D Figure 5. Multi-Level SoS Z Topology.....	369
Appendix D Figure 6. Multi-Level SoS Z Optimum Candidate.....	369
Appendix D Figure 7. Multi-Level SoS AA with Node Status.....	370
Appendix D Figure 8. Multi-Level SoS AA Topology.....	371
Appendix D Figure 9. Multi-Level SoS AA Optimum Candidate.....	371
Appendix D Figure 10. Multi-Level SoS BB with Node Status	372
Appendix D Figure 11. Multi-Level SoS BB Topology	373
Appendix D Figure 12. Multi-Level SoS BB Optimum Candidate.....	373
Appendix D Figure 13. Multi-Level SoS CC with Node Status	374
Appendix D Figure 14. Multi-Level SoS CC Topology	375
Appendix D Figure 15. Multi-Level SoS CC Optimum Candidate.....	375
Appendix D Figure 16. Multi-Level SoS DD with Node Status.....	376
Appendix D Figure 17. Multi-Level SoS DD Topology.....	377
Appendix D Figure 18. Multi-Level SoS DD Optimum Candidate.....	377

List of Tables

Table 1.1. Systems-of-Systems Comparison	3
Table 2.1. USA and UK Critical Infrastructure Sector Comparison.....	20
Table 2.2. Collaborative Systems Within the Human Body	24
Table 2.3. Example Security Attacks.....	33
Table 2.4. Types of Interdependency Based Upon Their Linkages	35
Table 2.5. Types of Interrelationships Between Collaborative Infrastructure Systems	36
Table 2.6. Types of Interdependence Related Disruptions	41
Table 2.7. Risk Management and Assessment Methods.....	49
Table 2.8. Summary of Network Vulnerabilities and Attack Factors.....	51
Table 2.9. Identified Cross-Site Scripting Vulnerabilities	53
Table 3.1. DDoS and Jamming Attack Detection Methods Summary.....	67
Table 3.2. Eavesdropping, Masquerading, and Snooping Detection Methods Summary.....	69
Table 3.3. Intrusion Detection and Intrusion Prevention Methods Summary.....	71
Table 3.4. Security Methods Summary.....	73
Table 3.5. Network Security Methods Summary.....	75
Table 3.6. Singular Risk Analysis Methods Summary	78
Table 3.7. Risk Analysis Methods Summary.....	80
Table 3.8. Laboratory Based Risk Reduction Summary.....	82
Table 3.9. Eliminating SPoF and Improving Network Robustness Methods Summary.....	85
Table 3.10. Algorithms that Overcome the Limitations of Existing Methods Summary	87
Table 3.11. Cascading Failure Methods Summary	89
Table 3.12. Interdependency Methods Summary	92
Table 3.13. Complexity Methods Summary	94
Table 3.14. Emergent Behaviour Methods Summary.....	96
Table 3.15. Risk Management and Assessment Methods Summary	100
Table 3.16. Network Modelling Methods Summary	102

Table 3.17. Attack Graph Methods Summary	104
Table 3.18. Data Assurance Methods Summary	108
Table 3.19. Network Optimisation Methods Summary	110
Table 3.20. Comparison of Analysed Methods Against Solution Requirements.....	111
Table 3.21. Summary of Reviewed Theoretical and Applied Solutions.....	112
Table 4.1 National Vulnerability Database Scoring Methodology Overview	158
Table 4.2. Identified Vulnerability CVE-2016-7211 Entry	159
Table 4.3. Example Parameters and Their Associated Risk Probability Scores	160
Table 4.4. Simulated Risk Parameters and Associated Risk Probability Scores	161
Table 5.1 Initial Visualised Security Vulnerabilities and Parameters.....	174
Table 5.2. Comparing Improved Solutions Robustness During Evolutionary Process Cycles.....	179
Table 5.3. Resource Usage During Evolutionary Process Cycle.	180
Table 5.4. SCRAM Resource Usage for Applied Algorithms	189
Table 5.5. Visualised Security Graph Vulnerabilities and Parameters	200
Table 5.6. Identified NVD Vulnerabilities with CVSS v3 Scores for Android OS Devices	202
Table 5.7. Identified NVD Vulnerabilities with CVSS v3 Scores for Windows OS Devices	202
Table 5.8. Identified NVD Vulnerabilities with CVSS v3 Scores for Linux OS Devices	203
Table 5.9. Excerpt from SCRAM Security Report for Network Figure 5.19-b.....	204
Table 5.10. Secure Network A Security Evolution Results	214
Table 5.11. Secure Network B Security Evolution Results	214
Table 5.12. Network C Security Evolution Results	216
Table 5.13. Network D Security Evolution Results.....	216
Table 5.14. Network E Security Evolution Results	219
Table 5.15. Network F Security Evolution Results	219
Table 5.16. Visualised Security Graph Vulnerabilities, Parameters, and Energy Levels	224
Table 5.17. WSN A Security Evolution Results	231
Table 5.18. WSN B Security Evolution Results	231
Table 5.19. Network A Enhanced Candidates Bridging Centrality Scores	231

Table 5.20. Simulated Smart City Security Evolution Results	237
Table 6.1. Simulated Smart City Networks Security Evolution Results.....	241
Table 6.2. SoS Visualised Security Vulnerabilities and Parameters.....	244
Table 6.3. Multi-Level SoS Evolution Results	250
Table 6.4. Simulated Multi-Level SoS Evolution Results	255
Table 6.5. Multi-level SoS Unevolved Vulnerability Performance Properties Comparison	258
Table 6.6. Multi-level SoS Evolved Vulnerability Performance Comparison.....	258
Table 6.7. Multi-Level SoS Unevolved Vulnerability and Data Access Performance Comparison...	266
Table 6.8. Multi-Level SoS Evolved Vulnerability and Data Access Performance Comparison.....	266
Table 6.9. Multi-Level SoS Sets G-I Unevolved Infrastructure Properties Comparison.....	276
Table 6.10. Multi-Level SoS Sets G-I Evolved Infrastructure Properties Comparison	276
Table 6.11. Multi-Level SoS Sets J-L Unevolved Infrastructure Properties Comparison	281
Table 6.12. Multi-Level SoS Sets J-L Evolved Infrastructures Properties Comparison.....	282

List of Abbreviations

ANT:	Ant Colony Optimisation combined with Local Search Algorithm
CI:	Critical Infrastructure
CVE:	Common Vulnerabilities and Exposures
CVSS:	Common Vulnerability Scoring System
DDoS:	Distributed Denial of Service Attack
DNP3:	Distributed Network Protocol Version 3
GA:	Genetic Algorithm
HIDS:	Host-Based Intrusion Detection Systems
ICAT:	Internet Catalog
ICS:	Industrial Control System
ICT:	Information and Communication Technology
IDS:	Intrusion Detection Systems
IPS:	Intrusion Prevention Systems
IoT:	Internet-of-Things
ISO:	International Standards Organisation
LAN:	Local Area Network
MANET:	Mobile Ad Hoc Networks
MulVAL:	Multi-host Multi-stage Vulnerability Analysis
NetSPA:	Network Security Planning Architecture
NIDS:	Network Intrusion Detection Systems
NVD:	National Vulnerability Database
OVAL:	Open Vulnerability Assessment Language
SCADA:	Supervisory Control and Data Acquisition
SCRAM:	SeCurity Risk Analysis and Mitigation
SoS:	Systems-of-Systems
SPoF:	Single Point of Failure
TABU:	Tabu Search Optimisation Algorithm
USB:	Universal Serial Bus
VPN:	Virtual Private Networks
TVA:	Topological Analysis of Network Attack Vulnerability
WSN:	Wireless Sensor Network

Chapter 1

Introduction

1.1 Foreword

Academics, organisations, governments, and cities have researched and heavily invested in Information and Communication Technology (ICT) in recent years, attempting to both understand and take advantage of the many benefits ICT platforms provide, amalgamating ICT with their existing infrastructures forming larger dynamic and complex Systems-of-Systems (SoS). In part, this is due to the growth of the Internet and ICT becoming cheaper to produce, thus widely available.

We define Systems-of-Systems as ‘a collection of distinct systems, each capable of being operated and managed independently, that when integrated can collaborate together to form a much larger extended infrastructure that then functions on objectives that the distinct systems could not fulfil on their own’. These platforms allow for physical, cyber and human elements to be combined, and while SoS are formed by the integration of components they are only truly capable of collaborating upon objectives via the exchange of data.

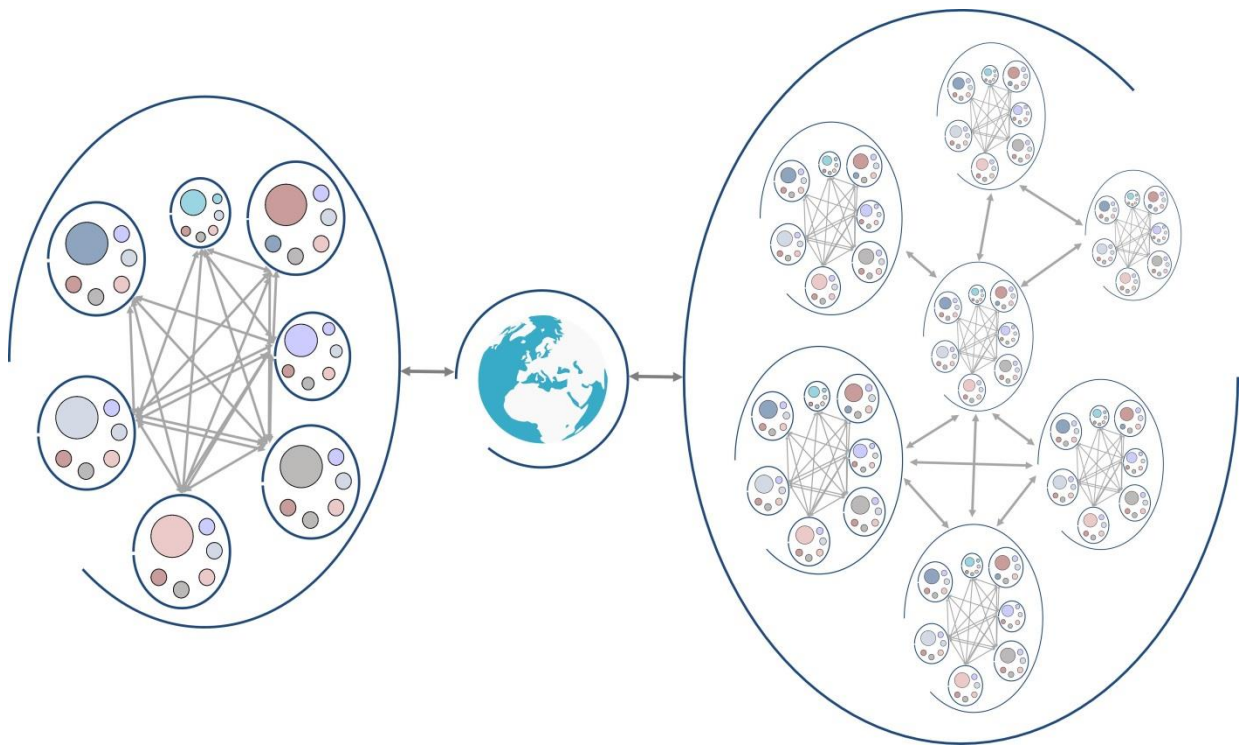


Figure 1.1 Schematic Representation of SoS and Multi-Level SoS Classifications

Furthermore, we consider multi-level Systems-of-Systems to be an assembly of SoS, reliant upon the unique assets or services provided by each of the distinct SoS in order to overcome the limitations of separate SoS, with the assembly of SoS performing as a single entity to meet the desired and often complex objectives of the multi-level SoS. Figure 1.1 provides a schematic representation of our perceived classifications of SoS and multi-level SoS.

These collaborative infrastructures are comparable to other ICT based networks, fraught with issues such as scalability, trust and security, with citizens and organisations increasingly finding themselves reliant upon the services and assets provided by these collaborating infrastructures, which have formed vast collective networks resembling ‘spider’s webs’ across towns and cities.

Systems-of-Systems Example – BT Group plc developed one of the largest telecommunication networks in the UK, which has vastly increased its products and services, by heavily investing in ICT to extend their communications network. The company has developed and is responsible for telephone exchanges, 28 million telephone lines, trunk network, local loop connections, broadband internet, Cloud computing, and other services. Countless external organisations and infrastructures deemed critical now depend on the physical and cyber assets provided by BT in order to collaborate with third parties and maintain control of their distributed systems, utilising BT’s infrastructure as their backbone. As BT’s communication network has grown across the UK, it has become increasingly difficult to effectively identify and manage risks within their infrastructure despite investment into network discovery and risk analysis methodologies, due to the sheer size, complexity, and dynamic nature of their infrastructures [1].

Features of Systems-of-Systems – The ad-hoc nature of SoS is to exploit interconnected services and infrastructures which are dispersed and interconnected via a variety of communication links, allowing for the distinct components to ‘pool resources’ and data, in order to fulfil identified objectives. Table 1 summarises some of the main features, benefits and issues of four different types of SoS.

Some of the main benefits of SoS is that they endeavour to deliver resilience against single points of failure (SPoF), increasing robustness, reliability, and performance, and can contribute to reductions in cost of operation. These platforms can combine physical, cyber, and human elements together, along with both new and aging technology. In addition, distinct systems within the collaborative environment can maintain their individual operation and management, ensuring that distinct systems can function independently, as part of the SoS, or even collaborate with other external SoS.

SoS allow for highly complex processes to be automated, that prior to developments within ICT and integration could not be achieved alone, and provide a means for processes to be continuously executed over long periods of time.

Table 1.1. Systems-of-Systems Comparison

SoS Example	Features	Benefits	Issues
Wireless Sensor Network	<ul style="list-style-type: none"> No fixed infrastructure. Self-organisation. Low cost. Dynamic network topology. Distributed sensing. Distributed processing. Deploy in remote areas. 	<ul style="list-style-type: none"> Large scale deployment. Non costly implementation. No centralised monitor. Resilient to node failure. Integrate new devices easily. Energy efficient. 	<ul style="list-style-type: none"> Restricted computation. Restricted power supply. Restricted storage. Less secure. Complicated to configure. Easily hacked/attacked. Lower network speed. Multi-hop communication.
Critical Infrastructure	<ul style="list-style-type: none"> Integrate physical and cyber infrastructures. Distinct systems can modify operation to meet objectives if malfunction occurs to support SoS. Can co-ordinate planning across sectors. Provide a restricted level of service in emergencies or during limited failures. 	<ul style="list-style-type: none"> Increase robustness. Automate processes. Increase economic development. Dynamic behaviour. Allows systems to be geographically dispersed. Delivers essential services. In the event of failure can return to operation quickly. 	<ul style="list-style-type: none"> Reliant on the assets of telecommunications. Reliant on legacy systems. Legacy systems no longer standalone or isolated. Application dependent data. Security is problematic. Vigorous security testing cannot always be applied. Failure can cause economic loss or catastrophic disaster. Difficult to replace systems. Introduces dependencies.
Internet of Things	<ul style="list-style-type: none"> Integrates large numbers of devices or 'things'. Home and industrial automation. Smart environments. Machine-to-machine communication. 	<ul style="list-style-type: none"> Allows for the automation of scheduled daily tasks. Ability to monitor large numbers of sensors. Ability to collate large amounts of data. Can save time and money. Allows for the automation of processes and control. 	<ul style="list-style-type: none"> Increases complexity. Increases incompatibilities between integrated devices. Data security and privacy. Easily hacked/attacked. Scalability. Large data sets.
Cloud Computing	<ul style="list-style-type: none"> Provides Infrastructure as a Service. Provides Platform as a Service. Provides Software as a Service. 	<ul style="list-style-type: none"> Reduce network costs. Shared collaboration. Reduces hardware and Support. Distributed access. Energy efficient. Scalability 	<ul style="list-style-type: none"> Reliant on transfer of data. Security data breaches. Data availability. Dependent on third party services. Dependent on Internet and network communications.

1.1.1 General Issues Within Systems-of-Systems

Due to the built-in redundancy and flexibility of systems, often failures are small and go relatively unnoticed as they have minor or no impact, while the more extreme failures are highly noticeable resulting in catastrophic consequences or can even cause the loss of human life.

SoS have been rapidly developed and deployed into countless environments, and due to societal dependence upon the assets of these infrastructures, organisations have been forced to implement upgrades without taking systems off-line. Collaboration depends upon the transfer of data between

systems within the SoS, and while we can simply connect systems together, it does not mean that positive integration will be achieved. Data does not instinctively flow just because systems are connected together, and incompatibilities with software, firmware, configuration, etc., can impede the functionality of integrated devices or systems. If merged incorrectly these integrated systems could be combined with disastrous results. This could potentially lead to systems becoming insecure and vulnerable to attack, systems could become unstable resulting in system wide crashing, system performance could diminish, and systems might fail to meet required objectives. Data created, stored, and transmitted within SoS can contain control commands, operational states, and one component's output can be another's input in order to fulfil system objectives, any failures or delays which impede data or its transfer will have negative impact(s) upon the SoS services and assets [2] [3].

The danger of 'bolting on' systems or phasing out components and functions is that it can increase emergent behaviour within the collaborative infrastructure. Furthermore, due to managerial independence, operational independence, and evolutionary development, components and functions can be phased in and out while the SoS is operational, without notification or regards to the impact it might have on other distinct systems. These changes can impact both the device and SoS operations and objectives, could result in systems or the entire SoS being uncontrollable or unpredictable, and can negatively impede the security of the distinct systems and entire SoS leaving them vulnerable and open to security attacks.

Another security challenge associated with these dynamic developments, is the fact that SoS are often deployed without being fully formed and will be forced to continually evolve as objectives are met and new objectives are identified.

1.1.2 Security Issues Within Systems-of-Systems

The collaborative environment of an SoS means the architecture of these networks is reliant upon changing, distributed, and diverse technologies, with varying components and software, and which each have conflicting configurations, security levels, and data access levels. This means assuring network and data security along with the quantification of vulnerabilities, risks, and security properties is immensely problematic. Unidentified vulnerabilities and risks between integrated components have the potential to leave all collaborative systems and components insecure, exposed, and vulnerable to attack vectors [2].

As organisations continue to integrate ICT within their infrastructures, SoS security will be further impeded by the added complexity and size of the networked infrastructure, and the SoS will be increasingly exposed to additional vulnerabilities and risks by their new collaborative relations. The

increased number of access points will provide new vantage points that can be exploited, and can assist in establishing new locations to launch both simple and sophisticated attacks from [2].

Maintaining operational relations on a daily basis between the distinct networked systems within the SoS is essential and an enormous challenge. It is vital that security does not negatively impede genuine and time critical communications during operations. Safeguarding data and maintaining an effective communication network is vital, and great consideration must be taken in how to safeguard secure routing within the topology between dissimilar devices. Securing an SoS under continuous evolution which is fully operational is an immense challenge, as emergent behaviours will often manifest long after integration, and it will be unclear who is responsible for identifying and safeguarding issues, further exacerbated by the fact it will not be possible to simply turn systems off for additional testing.

In future years we will also have to give greater consideration to security risks posed by collaborative systems being geographically distributed, and the impact that will be caused due to systems being governed by different laws and regulations. Geographically distributed SoS are critically reliant upon the transfer of data and networking capabilities, any interruption to data, including that caused due to local governance, between collaborative infrastructures, will result in the SoS failing to meet its objectives. As stated, data is a weak point and an SPoF, and as criminals and those with malicious intent realise the true value of data, it could become a potential target for malicious attacks; therefore increasing data security within SoS is of utmost importance.

The anonymity and benefits cyberspace offers has also allowed attackers with malicious intent to move away from traditional crimes such as bank robbery, and instead launch sophisticated and directed attacks against various infrastructures, for both profit and amusement, allowing battles to be waged on untraditional battlefields, such as the attacks against infrastructures deemed critical. It is vital that we identify relationships and vulnerabilities that form due to integration, and the features, benefits, and issues associated with different SoS topologies (summary in Table 1.1), in order to identify and mitigate risk, and assure security within SoS environments.

1.1.3 Risk Analysis Within Systems-of-Systems

Risk analysis endeavours to support an organisation in identifying vulnerable assets within their infrastructure, the threats, risks, and vulnerabilities which expose the infrastructure, how and when they occur, estimation of their impact upon systems (i.e. damage caused, financial losses, system interruption or failure, etc.), and identify the processes that can be undertaken to manage or mitigate risks to improve security and system robustness, while minimising exposure or damage.

Identifying risks within SoS environments is critical, organisations who fail to perceive or identify associated risks leave their systems exposed and insecure, and any resulting failing due to unidentified vulnerabilities can cause huge financial loss or critical failure. The US Army recognised the limitations of solutions attempting to secure and identify risk within collaborative infrastructures, and developed a Laboratory Risk Reduction Method, to support network integration, design, and risk analysis [4]. While this solution has strengthened development and deployment stages, it is highly time consuming, expensive, and emergent behaviours and risks unimaginable at time of deployment, can manifest long after the SoS has been deployed. Therefore this solution cannot be considered an iterative risk analysis methodology.

Risks that expose cyber SoS can include insecure or exposed ports, insufficient security policies, inadequate system hardening, use of vulnerable protocols, unencrypted communication, inadequate anti-virus, etc. Should vulnerabilities be ignored or remain unidentified, then these vulnerabilities can expose systems to application-level attacks, misconfiguration attacks, operating system attacks, password cracking, viruses and worms, and distributed denial of service attacks, etc. Cyber-attacks which exploit vulnerabilities are not the only risks that leave SoS exposed; human decisions and errors can result in increasing risks within SoS and contribute to failure, along with geographic changes and natural disasters, through to component and system misconfiguration, negative emergent behaviour evolution, cascade failure, and reliance upon networking capabilities. These attacks, vulnerabilities, and risks, can directly result in critical failings and the loss of human life.

Recent misfortunes such as the cascade failings that impeded the UK banking infrastructure [5], disruptions to several US airlines [6] [7], and the Japanese Fukushima Daiichi nuclear disaster [8], demonstrate the deficiencies and consequences that occur when risks remain unidentified and inadequate risk assessment and analysis has been conducted within SoS environments. Equally, attacks such as the Distributed Denial of Service (DDoS) attack against Domain Name System provider Dyn [9], the theft of 500 million account credentials from Yahoo [10], and the theft of money from approximately 20,000 Tesco Bank customers [11], prove that cyber-attacks against SoS are increasing and corroborate there are significant weaknesses in network security, and that existing theoretical and applied risk methodologies are inadequate and leaving SoS exposed and vulnerable.

Current risk solutions are inadequate and these real-world attacks and disasters demonstrate that current methods are failing to be successfully applied to these large dynamic and complex SoS. This is in part both due to the topology of the SoS and due to the risk methodologies being too complicated, too rigid, and broad in nature, so difficult to implement within heterogeneous SoS. The dynamic nature of SoS means it is problematic to identify and monitor vulnerabilities and risks that expose the SoS to potential attack vectors, to predict issues, potential failures, and the consequences of such failures, and identify who is responsible for monitoring systems, identifying issues that propagate, and who is accountable for initiating the appropriate resolutions.

These identified weaknesses and shortcomings that leave SoS security vulnerable and exposed, corroborate the need to develop an appropriate SoS security and risk analysis framework, that can overcome the limitations of other solutions and challenges posed by the topology and dynamic nature of SoS, and which provides the functionality to identify and visualise potential risk factors and model interdependent links between collaborative components. In this thesis the proposed SeCurity Risk Analysis and Mitigation Framework is presented, which aspires to address these issues. If risks can be identified and mitigated, we can increase SoS security and robustness, and limit the SoS exposure to failures and attack vectors. Risk is unavoidable and organisations will always have to contend with risk, but by identifying and understanding the risks which are both acceptable and unacceptable, measures can be undertaken in advance to mitigate or manage the risks effectively prior to failures or attacks. Meaning risk taking becomes a calculated intentional act.

1.2 Motivation and Research Gaps

The motivation for our research stems from the real-world SoS failures acknowledged in Section 1.1.3 above, which confirms the need for continued research into identifying and mitigating risks within SoS, as their initial or exacerbated failures can all be directly attributed to unidentified risks. Had these vulnerabilities been identified prior, then the disasters could have been lessened or fully prevented.

Similarly, having undertaken an in depth literature review, critically assessing current theoretical and applied solutions, and researching the challenges and risks that expose SoS to critical failings, we ascertained that currently there is no single solution that can adequately identify, map, and understand every critical link and vulnerability within SoS topologies for the life of the infrastructure [1] [4]. To some extent, this is attributed to the sheer number of components and the multiple networks which form the SoS infrastructures, the complexity of the topology, the decentralised nature of the environment, the ever evolving infrastructures and adaptations to system objectives, and due to a lack of ability to accurately perceive risk effectively [12], despite increased research and development. These difficulties prove problematic for the majority of solutions that attempt to secure large networks, with many methodologies struggling with the identification of issues, and failing to broadly apply their research [13] or focusing on single vulnerabilities and attacks [14]. While other risk methods when applied, increase complexity within the topology [15].

The dynamic nature of SoS means the collaborative relations formed with other infrastructures will continually expose the entire SoS to additional risks, vulnerabilities, and potential cascade failures caused as a direct result of the tightly coupled links. Often network data is not encrypted and both the network and data is insecure, exposed via various vulnerabilities and risks. With no assurance for security, SoS are powerless against risks and with no central management, and systems continually

evolving, it can be difficult to detect and respond to failures when they occur. Considering the limitations of current solutions, it is vital that proposed techniques secure every component or identify their vulnerabilities, in order to mitigate risks, assure communication security, and increase robustness for the entire collaborative infrastructure.

The work in this thesis is motivated by wanting to address the following main research challenges, which are:

- **Measuring Security between Interconnected Components and Systems:** Due to the sheer complexity, size, and dynamic nature of SoS, the majority of developed solutions struggle to be applied and quantify communication security for these large multi-networked infrastructures. In general, this is directly attributed to the number of distinct networks within the SoS which are managerially independent, and the sheer number of components forming each network. Collaborative infrastructures are often widely dispersed, and are formed between a complex series of ICT communication links and connecting devices, forming vast and complex topologies that are difficult to understand and monitor. Retaining their independent management further complicates network security, as both assets and services can be added and removed without informing or seeking permission from their collaborative partners [16]. This means new security risks can be introduced without warning and remain undetected, due to the complexity of the infrastructure and limitations of security solutions that identify risks and report them. The motivation for overcoming this challenge is to improve upon existing methods and generate a solution that is both backward and forward compatible [4], not domain specific [17], and can be applied to the entire collaborative infrastructure, ensuring that techniques are compatible and dynamic to guarantee security can be accurately identified and quantified to protect systems against unperceived risks, along with potential future integration of components and networks, thus assuring the future security of the network for the life of the SoS.
- **Identification of Risks and Interdependencies:** The dynamic and heterogeneous nature of SoS has heavily impacted organisations, specifically their ability to efficiently identify and measure risks that pose threats and leave systems exposed, and the interdependencies that form due to the tightly coupled bonds that form between collaborative systems [1]. The inability to accurately identify risks is often due to the organisation failing to perceive risk due to a lack of experience or training, and can be impacted due to personal bias [1] [18]. Often risk methodologies are highly complex, too broad, and domain specific, with organisations struggling to apply the methodologies to their systems or not being able to apply them fully to the entire SoS [13] [19]. With risks remaining unidentified SoS are further exposed due to the failure to identify interdependencies that form, attributed to the sheer size and complexity of these SoS. Current solutions do not have the capability to understand and

identify all interdependencies that form within these collaborative infrastructures [1]. The dependencies heavily impact SoS security as, should one system be insecure, then the level of risk for the other collaborative systems increases, while reducing the security in other dependent systems. The motivation for overcoming this challenge is to improve upon existing methods as security is a challenge for SoS infrastructures, and as organisations continue to integrate more systems and service, this will increase interdependencies and risks associated with security will become more complex. In addition, it is essential that solutions are not just backward compatible but forward as well, along with being dynamic to ensure that the automated solution can keep up with the speed of ICT advancement, and the dynamic nature and size of the SoS as they are continually developed during their life cycle.

- **Data Security in Unsecure and Unencrypted Networks:** One of the benefits of SoS is their ability to form a collaborative environment for distinct devices to integrate and combine their resources or share their data in order to meet a shared objective. These collaborative relations are reliant upon the transfer of data in order for them to form relations and meet their objectives. These distinct devices can all be individually configured with varying security grades and solutions, and with differing data access requirements. In order to protect data various solutions have been utilised and proposed, but often these techniques are unsuitable or not appropriate for the SoS environment in which they are to be deployed [20] [21]. A simple solution is to encrypt data as it traverses across insecure networks, however, on SoS such as those formed using sensor nodes and other devices with limited computational power and memory, alternative techniques are more desirable. The motivation for overcoming this challenge is to develop a trustworthy technique that can be applied to diverse systems and their components in order to assure data as it traverses across unencrypted and insecure networks, in addition to the technique not impacting the limited processing resources of components within the topology of the SoS.

1.3 Aims and Objectives

The project aims considered are, to identify the limitations of existing solutions and techniques, which will be reflected upon to assist with the development of a solution capable of identifying and mitigating risks within SoS and multi-level SoS environments. Subsequently, we intend for the solution to measure the security of individual devices and the entire SoS topology, identifying risks and interdependencies that impact and form between collaborative components. In addition, the solution should quantify the robustness of the entire collaborative infrastructure, in order to evaluate the appropriateness of these environments. Ultimately, the proposed solution should resolve the failings of existing techniques that fail to assure data and quantify risk(s), and have the capacity to

secure the collaborative environment utilising only the existing networked resources. By increasing the cyber resilience of SoS and multi-level SoS, when networks or components are attacked or fail the impacts should be minimised due to the applied solution.

The main objectives of this thesis which are necessary to resolve current inadequacies in SoS risk analysis and network security are:

- Conduct detailed background literature research into the challenges and risks that expose SoS, the issues impacting the ability of current solutions to secure these dynamic networked infrastructures, and the methodologies that fail to identify and quantify risks within SoS.
- Develop an SoS security risk analysis solution to calculate the security level of the entire SoS using vulnerability analysis, node property aspects, topology data, and other factors, to improve and mitigate risks without introducing additional resources into the SoS infrastructure.
- Develop a solution that can analyse and quantify the robustness of the SoS environment based on the relevant data captured from the application of the security risks analysis solution.
- Conduct a detailed investigation into optimisation techniques and algorithms in order to identify which solutions suit SoS to mitigate the risks.
- Conduct a case study on a specific network type such as WSN, and expand the solution to encompass a different risk vector utilising the same developed risk analysis framework and robustness techniques.
- Validate that the algorithms and principles are effective for identifying and mitigating risks within multi-level SoS, in order to increase multi-level SoS security and robustness.

1.4 Novel Contributions

The thesis presents a novel approach for SoS security and makes the following novel contributions:

- An evolutionary SeCurity Risk Analysis and Mitigation Framework, which overcomes the inadequacies and limitations of existing solutions. The framework quantifies the security level of each device and the entire collaborative infrastructure. This solution is dynamic and supports the integration of multiple risk parameters, combining vulnerability scores collated via various sources such as parameters identified using vulnerability analysis, and calculated risks scores based on node property aspects, topology data, and other factors. The technique also assists with identifying and visualising vulnerabilities, data access violations, and interdependencies within the SoS infrastructure.

- A statistical robustness measurement technique that can quantify the robustness of the SoS environment. The technique combines five distinct parameters to quantify the suitability of the network configuration and security into a single comparable parameter, and when combined with the evolutionary risk mitigation algorithm, assists to identify topologies that increase and decrease security and risk.
- An evolutionary risk mitigation technique that evolves the configuration of the network communication links between components within networks and collaborative networks, in order to identify the optimal SoS communication configuration. Unlike existing approaches, the proposed technique is automated, and does not require additional resources to be added to the SoS infrastructure in order to secure and mitigate risk factors. This technique also assisted with the study and implementation of other optimisation techniques within the SoS SeCurity Risk Analysis and Mitigation Framework, evaluating their usefulness, and identified that not all optimisation process can be applied to such large dynamic and complex SoS infrastructures.
- An SoS SeCurity Risk Analysis and Mitigation Framework that adopts a hybrid and scalable approach to secure and mitigate risks in multi-level SoS, which are the amalgamation of large distinct SoS. This technique overcomes the limitations associated with complex SoS, providing an accurate means to measure, identify, and visualise security and vulnerabilities, to identify and quantify vulnerabilities and mitigate risks, and to measure the robustness of the entire multi-level SoS. This limits the multi-level SoS exposure to failures and attack vectors, with analysis undertaken on multi-level SoS that consist of up to twelve unique heterogeneous SoS.

Aspects of the research undertaken and presented in this thesis have been published in eight academic research journals and conferences, with a comprehensive list of publications being provided at the beginning of the thesis.

1.5 Research Findings

Research outcomes – An in-depth review of existing research and development within the field of SoS has identified limitations with existing solutions. Inadequacies include their inability to identify and mitigate risk within dynamic and complex SoS, and difficulties quantifying security and the robustness of the entire infrastructure. The weaknesses identified are primarily attributed to the inadequacies of existing risk and vulnerability analysis techniques, which allow for vulnerabilities to remain unidentified, and their inability to quantify accurate security and robustness levels due to inaccurate vulnerability identification and risk assessment, and due to the inability of these techniques

to overcome the complexity, dynamic nature, and size of the SoS infrastructures. This allows vulnerabilities and risks within SoS to remain undetected; resulting in the security of SoS being exposed to attack vectors and allows for issues to propagate.

Research observations – Addressing the aims set out in this thesis, our research identified the limitations of existing methodologies in order to ensure that the proposed techniques and framework do not experience the same issues. In general, the proposed research and solutions are highly theoretical or not applied to large distributed networks and SoS, meaning there is no assurance in regards to scalability, and their appropriateness when applied to multi-level SoS. Research identified that integration increases complexity, interdependencies, and the risk of SPoF evolving, and as societal dependence on the assets of SoS continues and development of these infrastructures increases, system complexity will correspondingly increase. Methods and visualisation techniques in regards to risk analysis, attack analysis, failure analysis, and approaches that focus on identifying complexity, cascading failure, and interdependencies, typically focus upon a specific area and type of infrastructure.

Novel contribution outcomes – An in depth review of existing research and developments within the field of SoS has influenced our work and assisted us in developing a novel methodology and framework capable of identifying and mitigating risks within SoS without introducing additional resources into the infrastructure, and the capacity to accurately quantify the security and robustness levels of the entire collaborative multi-level SoS. During the development and implementation of our proposed methodology it became apparent that some of the applied algorithms and techniques were imprecise and unpredictable, therefore it became necessary to develop and implement improvements.

1.6 Thesis Structure

The research detailed in this thesis is arranged into seven subsequent chapters, with the following overview outlining the order and content of these chapters:

Chapter 2: Background

This chapter defines the types of infrastructures which can be categorised as Systems-of-Systems, the rewards and challenges associated with SoS, and identifies the main issues that expose networked systems leaving them insecure and vulnerable to attack vectors which are directly attributed to the SoS characteristics. Presenting this background information ensures that the reader can comprehend the challenges and inadequacies that currently exist within this area of research. In addition, this chapter also summarises methods that attempt to identify and quantify the risks associated with larger heterogeneous infrastructures, and those which model interdependencies and cascading failures such as attack graph generation methodologies.

Chapter 3: Related Work

In this chapter we provide a critical review of research which relates to network and SoS risk analysis and assessment, vulnerability identification, including risk modelling, and network optimisation techniques. We review how these methodologies are applied in regards to network security, their effectiveness in identifying vulnerabilities and interdependencies in an attempt to mitigate risk(s), and how they can be improved to provide effective solutions to the challenges outlined.

Chapter 4: SeCurity Risk Analysis and Mitigation (SCRAM) Framework

This chapter presents the proposed SoS SeCurity Risk Analysis and Mitigation (SCRAM) framework design. Firstly, the section offers an analysis of the problems identified via the conducted research, followed by a comprehensive overview of the structure and design of the proposed SCRAM solution. This is followed with a detailed description of the framework's processing stages, which are the Initial Operation Stage, Network Discovery Stage, Attack Graph Generation and Analysis, and Risk Mitigation Analysis. In addition, the theoretical principal algorithms and methods which are incorporated into SCRAM are discussed. These novel techniques assist us to effectively meet our identified aims and objectives, and we summarise how they support the framework's ability to measure security between interconnected components and systems, quantify the robustness of the network, and identify and mitigate risks without utilising additional resources.

Chapter 5: Implementation and Evaluation

This chapter defines how the SCRAM solution was developed, and how the theoretical principles were implemented, discussing the configuration of the essential methods and simulated environment in order to evaluate them. This section describes initial evaluation of the SCRAM framework and applied techniques, evaluating the methodology against the fundamental design requirements. Corroborating the framework's effectiveness to identify and mitigate risk, and ensure the aims and objectives established are accomplished. The chapter concludes with a case study that validates the appropriateness of the SCRAM solution and applied techniques, and corroborates the framework's dynamic capabilities to adapt to additional risk factors within SoS environments.

Chapter 6: Multi-Level SoS Security Analysis and Evaluation

This chapter presents the experiments generated within SCRAM that have allowed us to analyse and evaluate both the proposed SCRAM solution and the integrated theoretical techniques when applied to multi-level SoS. Each generated multi-level SoS environment provides a rich topology, formed from multiple distinct SoS each constructed from differing devices, communication links, and vulnerabilities. By generating these multi-level SoS environments, a comprehensive data set is produced for analysis and evaluation. Assisting us to evaluate and determine the SCRAM solution principles' and techniques' functionality and its ability to adequately identify and mitigate risks within

multi-level SoS topologies, and corroborate the solution's ability to quantify the security and robustness for the entire collaborative environment.

Chapter 7: Conclusion and Future Work

This chapter presents a summary of the work presented in this thesis and discusses the successful novel contributions achieved in order to overcome the limitations of existing solutions, and the challenges associated with SoS security and risk analysis. In this section we also summarise the limitations of our SCRAM framework, discuss future work that could be undertaken in order to extend our framework and develop it further to address other challenges, and the identified solution limitations. Finally, we conclude by summarising the presented work in this thesis and its novel contributions that have overcome the associated issues of multi-level SoS security, and risk identification and mitigation.

Chapter 2

Background

Over recent years ICT has augmented interoperability, and assisted with the integration of both Physical and Cyber Assets. Numerous industries, organisations, and governments have been quick to take advantage of the many benefits these technologies offer. As ICT has not only allowed complex objectives to be fulfilled and automated, it has also reduced financial cost of operation and maintenance, and increased the overall robustness, performance and reliability of these large networked systems. Subsequently, vast complex heterogeneous networks have emerged in areas where the infrastructure's operation is often deemed vital for society (i.e. critical infrastructure, disaster management, banking, etc.). These infrastructures are often independently operated and void of a central management structure, and can be further defined as Systems-of-Systems. Figure 2.1 provides a high level representation of the interconnected relationships that can now form between cyber based infrastructures, which allow organisations to collaborate and form extended networked infrastructures (i.e. Systems-of-Systems).

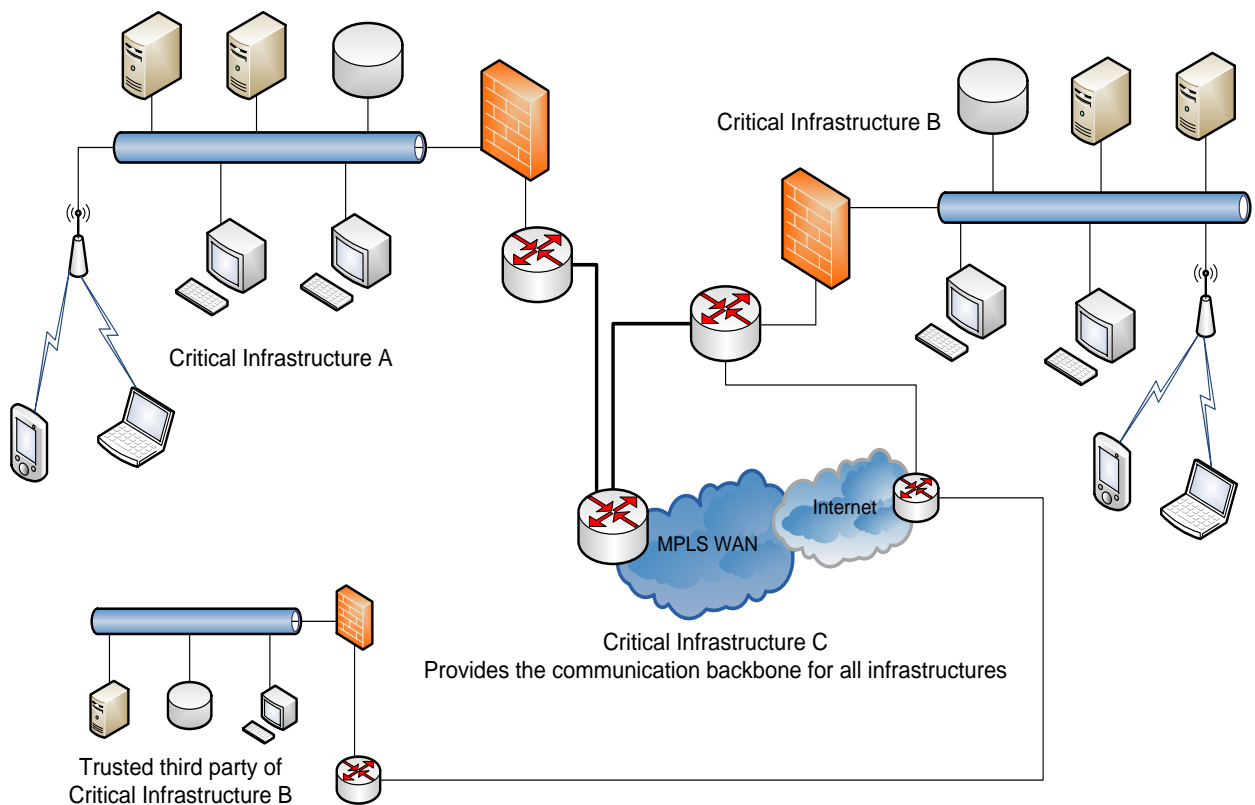


Figure 2.1. Cyber Systems-of-Systems Architecture

Nonetheless, these large networked infrastructures struggle and are affected by the demand for interoperability, performance, security, and usability. In addition, when integrating distributed technology formed from varying components, with differing security levels, performance, and dynamic in nature, it becomes challenging to predict how they will interact and be affected should failure occur or the exchange of critical communications stop. Unpredictable system and component performance means initiating failures can quickly develop and cascade to other systems with no warning. Environmental factors surrounding SoS can also profoundly impact systems and their functionality, exposing these structures to potential attack, failure, and risk vectors due to unidentified vulnerabilities.

In this chapter we review the paradigm of SoS and the challenges which plague these infrastructures. Also examined are the elements of risk within SoS that jeopardise the security of the network and have the potential to be the source of SPoF. Network risk assessment is conveyed within this section, focusing on risk analysis, intrusion detection, and the methods that are utilised such as attack graphs that attempt to assist in risk identification, and security assessment and visualisation.

2.1 Systems-of-Systems

Academics and industry practitioners have remained divided upon a single definition to best describe all Systems-of-Systems. Jamshidi defined SoS as “*large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal*” [22], and Kotov defines them as “*large-scale concurrent and distributed systems the components of which are complex systems themselves*” [23]. Maier [24] distinguished three separate types of SoS (directed, virtual and collaborative), basing each upon the actual relationships perceived between the distinct systems which formed each of the collaborative networks. While Dahmann and Baldwin [25] expanded upon this work and gave recognition to a fourth type of SoS (acknowledged), due to development and growth within the US Department of Defence systems.

Boardman and Sauser [26] in an attempt to differentiate SoS from distinct systems, concentrated upon five distinguishing characteristics thus opting to ignore definitions, the characteristics they identified are Autonomy, Belonging, Connectivity, Diversity and Emergence. While Shenhar [27] back in 1994 was one of the first to propose a systems classification matrix, and defined SoS as “*A large widespread collection or network of systems functioning together to achieve a common purpose*”.

We define Systems-of-Systems as ‘a collection of distinct systems, each capable of being operated and managed independently, that when integrated can collaborate together to form a much larger extended infrastructure that then functions on objectives that the distinct systems could not fulfil on their own’. In addition, we consider multi-level SoS to be ‘an accumulation of SoS, which are reliant

upon the assets or services provided by each of the distinct SoS in order to overcome the limitations of separate SoS, with the assembly of SoS performing as a single entity to meet the desired and often complex objectives of the multi-level SoS’.

These platforms allow for physical, cyber, and human elements to be combined, and while SoS are formed by the integration of components and multi-level SoS via the integration of SoS, they are only truly capable of collaborating upon objectives via the exchange of data. This unavoidable reliance upon the transfer of data to ensure integration (i.e. one component’s or SoS output being another’s input), means the dependence upon data which previously did not exist introduces new cyber interdependencies and increases complexity within SoS and multi-level SoS, along with introducing the risk of cascading failures and impacting security.

2.1.1 Systems-of-Systems Types

The four main classifications for SoS are directed, virtual, collaborative, and acknowledged, and are based upon perceived relationships between the components and systems which form the SoS.

- **Directed Systems-of-Systems** – These SoS are centrally managed to fulfil specific objectives, however, their distinct systems operate independently, thus remain subordinate to complete their managed objectives or new objectives defined by managers [24] [25]. An example is the Royal Naval collaboration between the ARTISAN 3D radar and Seawolf Mid-Life Update program, which delivers an SoS in an attempt to increase ship survivability. Both the radar and missile system can function independently, and potentially can simultaneously collaborate with other distinct systems to achieve objectives outside the scope of this SoS [28].
- **Virtual Systems-of-Systems** – These SoS are neither centrally managed nor have a centrally agreed objective. Within these types of SoS large scale behaviours potentially can emerge, forcing the SoS to heavily rely upon invisible components to meet objectives [24] [25]. An example would be the World Wide Web, as it is physically and administratively distributed, and since its establishment no single organisation has directly controlled it [24].
- **Collaborative Systems-of-Systems** – These SoS do not have the ability to command systems directly to fulfil objectives from the central management, instead the distinct systems must volunteer to work in partnership to fulfil any agreed central requirement. An example would be the Internet; this began life as a directed SoS however is now defined as a collaborative SoS due to its continuous evolution. This is because organisations like the Internet Engineering Task Force developed standards for the Internet, however, currently have no authority to enforce them [24] [25].

- **Acknowledged Systems-of-Systems** – These SoS have defined management, resources, and established functions, though the collaborating systems each maintain their distinct functions, ownership, funding, and development. Should distinct systems evolve or require adaptation, both the SoS and the distinct systems must collaborate. An example is the Department of Defence when SoS managers have no direct control over the collaborating distinct systems, instead they only have the ability to advise and influence other managers as the SoS evolves and new objectives are identified [25].

Due to the broad definition of SoS, when we study real world SoS and their failings, these classifications act as low level subcategories, and as stated allow us to group together SoS based upon perceived relationships. Consequently, this has also allowed us to identify and investigate correlated failings and risks which directly impede and expose specific types of SoS, based upon infrastructure relationships. It was essential to understand the configured relationships forming SoS and the challenges that must be overcome, especially as we wish to apply our proposed framework and techniques to not only diverse SoS but also multi-level SoS which are an accumulation of divergent SoS, and could contain any number of differently classified SoS. Critical research of SoS classified in each of these areas allows us to develop a solution capable of considering these types of relationships, this will also ensure that classification types will not affect the results, validating the method's appropriateness to be applied to diverse multi-level SoS

2.1.2 Systems-of-Systems Examples

SoS are not new developments, they have been heavily developed and researched for several decades after the realisation that single systems can no longer meet all required objectives. These infrastructures have developed and emerged in areas that include manufacturing, healthcare, transportation, telecommunication, banking, critical infrastructure, military applications, space research, and disaster management. It is irrelevant if the infrastructure has emerged in place, been long established, pre-planned, or created quickly in response to a critical incident, these infrastructures have formed intricate tightly coupled structures like 'spiders' webs' which have manifested themselves and become so integrated within our daily lives, we don't notice they are there until one fails. The scale and size of these varied SoS over the last few decades has surprised many industry practitioners, particularly how rapidly they have been developed and deployed within many infrastructures which are deemed critical. Three examples of SoS are critical infrastructures, Smart Cities, and the human body which we summarise.

2.1.2.1 Critical Infrastructures

Critical Infrastructures (CI) can be defined as an SoS, due to the fact over the last few decades they have been quick to embrace new technology and merge it with aging legacy systems, due to the many advances within ICT. The industry has integrated physical and cyber assets with their existing infrastructures, meaning they no longer remain isolated and standalone systems. Instead they now rely on telecommunication assets as their backbone to provide the necessary links to allow for interconnectivity between their collaborative systems, which form large extended networks. Telecommunications are responsible for not only ensuring that the distinct systems collaborate by transferring data in a timely fashion and for data to be coordinated, but this infrastructure often is also responsible for connecting and controlling the operation of other CI [29] [30].

Systems controlling infrastructures deemed critical and the CI themselves have been forced to implement upgrades without taking systems off-line, due to societal dependence upon their services and assets. This means that as new requirements were identified, systems were merged on top of existing infrastructures like building blocks [31], and vigorous testing could not be applied to the network to ensure long term connectivity or functionality.

While CI expansion has the potential to increase economic development, any direct failure to these infrastructures could result in huge economic loss or cause catastrophic consequences. CI cannot be simply turned off or replaced overnight, and updating or transferring the functionality of legacy systems to new modern cyber ICT systems certainly will take years, if it is both viable and possible. Assuring the security for these infrastructures is problematic leaving CI exposed, and it will cause short and long term challenges to system integration [12]. Furthermore, as organisations and governments become more integrated, attacks against these infrastructures will also inherently increase.

The USA Patriot Act defines CI as “*systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters,*” and categorise their critical infrastructures into thirteen sectors as shown in Table 2.1 [32]. While the UK defines its CI as “*certain ‘critical’ elements of infrastructure, the loss or compromise of which would have a major, detrimental impact on the availability or integrity of essential services, leading to severe economic or social consequences or to loss of life,*” and categorise their critical infrastructures into nine sectors as shown in Table 2.1 [33].

Table 2.1. USA and UK Critical Infrastructure Sector Comparison

USA	UK
<ul style="list-style-type: none"> • Agriculture • Food • Water • Public Health • Emergency Services • Government • Defence Industrial Base • Information and Telecommunications • Energy • Transportation • Banking and Finance • Chemical Industry and Hazardous Materials • Postal and Shipping 	<ul style="list-style-type: none"> • Communications • Emergency Services • Energy • Financial Services • Food • Government • Health • Transport • Water

Source: U.S. Government Publishing Office, Public Law 107-56-Oct.26, 2001 [32], The Intelligence and Security Committee of Parliament, Foreign involvement in the Critical National Infrastructure: The implications for national security [33].

One of the main factors behind the integration of key systems within CI is to increase system robustness and to automate processes which are time critical and reliant upon being initiated in sequence, hence increase the lifespan and performance of these key systems. These infrastructures then have the built-in capacity to adapt to various unexpected situations, as it allows the distinct components and systems to modify their operation and adapt to fulfil objectives should malfunctions appear in other key components or systems.

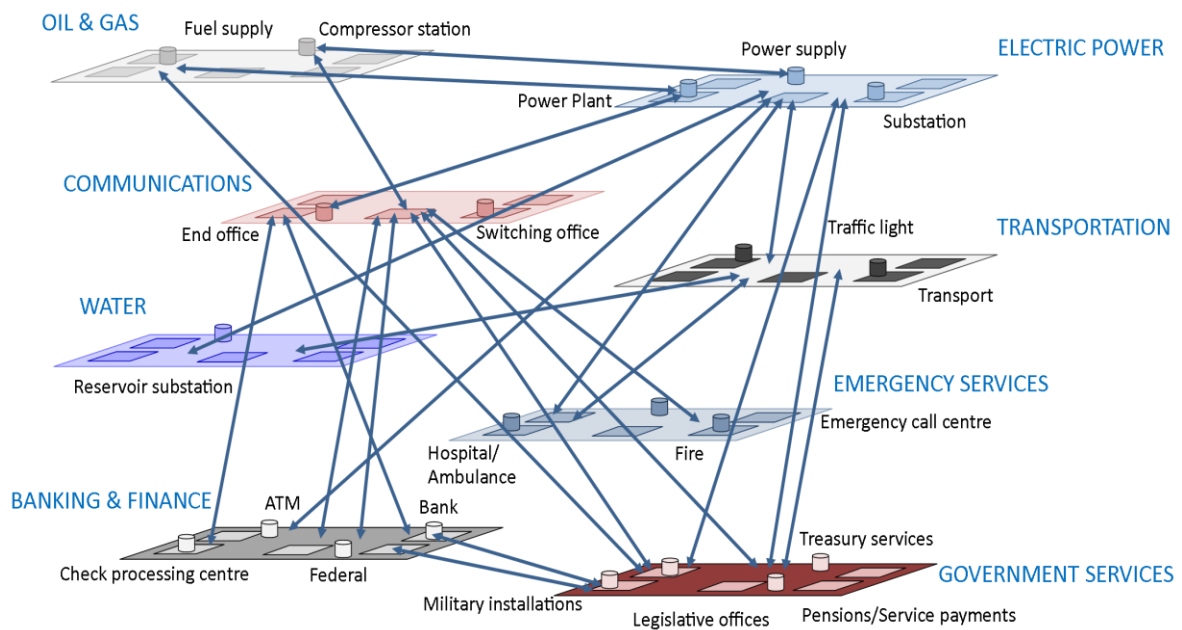


Figure 2.2. Critical Infrastructure Example Architecture

Source: Schematic representation of interdependent relations within the architecture of a critical infrastructure, Forester et al. [34].

Figure 2.2 demonstrates how CI have become heavily integrated and reliant upon each other's services and assets, and assists to visualise the complexity and introduction of interdependencies that previously did not exist, which have the potential to expose the distinct infrastructures to additional risks or can impact functionality in the event of partial or full cascade failure occurring. Alarmingly the majority of CI are owned and run by the private sector and can make changes without informing other infrastructures who are reliant upon their services and assets.

In addition, there are few protection methods in place such as specific laws, practices, and regulations that operators must conform to, or minimum security standards that must be applied or upheld. Furthermore, as CI continue to integrate and form extended infrastructures, in the future they may be extensively geographically dispersed in different jurisdictions or countries, meaning in addition the local laws, governance and policies will be forced to be considered and applied. An excellent example of this is the USA which has 50 states, 4 territories, and has more than 87,000 jurisdictions of local governance, with many of the country's CIs bordering a number of different national borders that inevitably require international cooperation [29].

2.1.2.2 Smart Cities

Smart Cities are emerging globally in an attempt to manage urban population growth and the assets of cities. It is estimated that over half of the world's population resides within an urban region. Increasing urban population growth is causing a significant number of difficulties, and cities globally find themselves struggling with inadequate and aging infrastructures, inadequate resource management (including power, water and waste), air pollution, traffic congestion, and issues with resource volume and allocation. This relatively new paradigm is generally formed via the integration of ICT, taking advantage of the benefits provided by platforms such as the Internet, IoT, and WSN, which are incorporated with new and existing infrastructures. These evolvments can provide new and improved services for its citizens, while reducing administration costs and improving the city's management of its assets [35].

Again, there is no single accepted definition to best describe all Smart Cities, nor a unified frame for comparison. Harrison et al. [36] state Smart Cities are *“connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city”*, and Hall [37] define Smart Cities as *“A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rail/subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens.”*

Smart Cities allow distinct components and systems to ‘pool resources’ and data. These dynamic networks working in partnership will endeavour to provide vital services to accomplish both simple and complex tasks, which they individually could not achieve, relying upon data being transferred across a complex series of devices and a number of critical data communication links. Failures or delays which impede data transfer will have negative impacts upon the Smart City’s services and assets.

The eight main critical factors that impede Smart Cities were highlighted in the work of Chourabi et al. [35], these internal and external challenges that must be addressed are categorised as management and organisation, technology, governance, policy context, people and communities, economy, built infrastructure, and the natural environment.

When the paradigm of IoT is incorporated into Smart Cities, we have to consider that this infrastructure is comparable to other ICT based networks and is fraught with issues such as scalability, trust and security. IoT encompasses a variety of devices or ‘things’, such as physical components including sensors, actuators, surveillance cameras, home appliances, displays, vehicles, and mobile phones. These devices when utilised within a Smart City, are dispersed and interconnected via a variety of communication links, and with varying security and access policies [38].

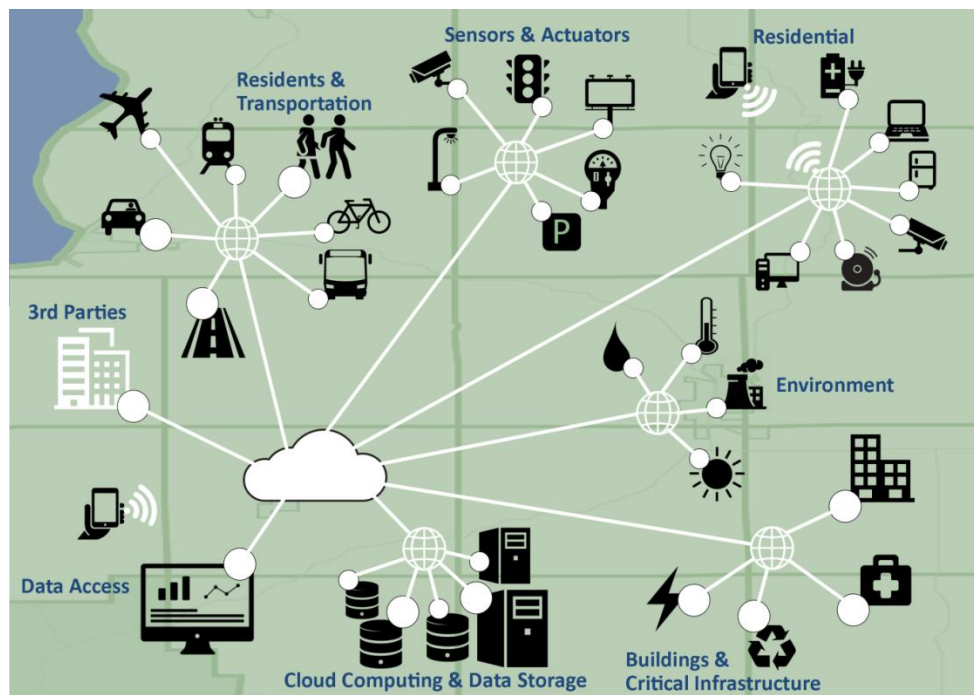


Figure 2.3. Smart City Topology Schematic

Likewise, the incorporation of WSN within Smart Cities is fraught with challenges, as WSN platforms connect objects to the Internet via a gateway. Typically the gateway forwards sensed data collected via the WSN across the Internet using a communication protocol. These protocols traditionally only communicated one-way, however, it is possible for two-way communication via the

WSN. This means that sensors are capable of both receiving and sending data such as temperature, pressure, motion, voltage/current, etc., and act upon commands received, which enhances both automation and user experience within the Smart City environment [39]. Figure 2.3 is a demonstrative example of such devices and connections.

As Smart Cities continue to integrate IoT and WSN with their existing active infrastructures, exposure to attacks will inevitably increase. In part this is due to the increase in new access points, which provide new vantage points that can be exploited, and assist in establishing new locations to launch attacks from. In addition, these cities will be further impeded by the sheer size and complexity of the integrated components forming the topology, and networks will be increasingly exposed to additional vulnerabilities and risks by their new collaborative relations.

When cities integrate changing distributed and diverse technologies, with varying components and software, and which each have dissimilar security levels, quantifying vulnerabilities, risks, and security properties is highly difficult. We consider this as one of the most important challenges which must be overcome to prevent serious flaws exposing entire cities.

We surmise that all collaborative components which form Smart Cities, WSN, and IoT (e.g. actuators, sensors and smart devices), are systems in their own right. Subsequently, we consider all of these platforms both interconnected and individually as Systems-of-Systems.

2.1.2.3 The Human Body

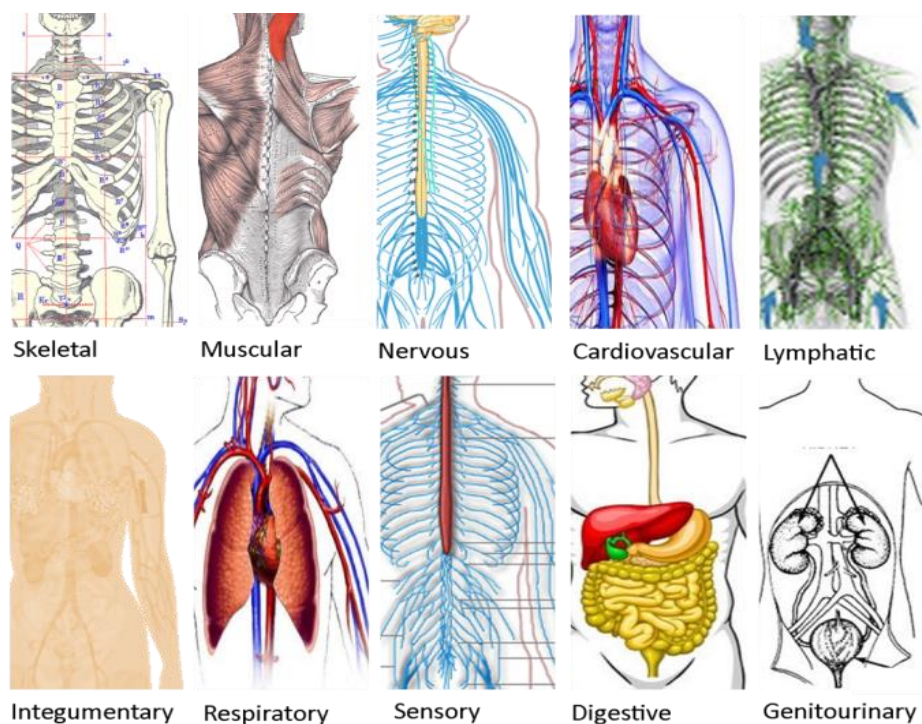


Figure 2.4. Human Systems Topology

A more extreme example of SoS includes the human body, as it is also made up of collaborating systems each functioning together to fulfil the objective of life (Figure 2.4). These distinct systems include the skeletal system, muscular system, nervous system, cardiovascular and lymphatic system, integumentary system, respiratory system, sensory system, digestive system, genitourinary system, and endocrine system, summarised in Table 2.2 [40].

Table 2.2. Collaborative Systems Within the Human Body

Type	Description
Skeletal	<ul style="list-style-type: none"> The Skeletal structure consists of 206 bones. These bones form the human body's internal framework, allowing for an upright posture and protection of all vital organs. The Skeletal structure functions include assistance with movement, storage, and maintenance of chemical level. Bones are also categorised as a living organ.
Muscular	<ul style="list-style-type: none"> The largest system in the human body is the Muscular system. Muscles are essentially positioned throughout the human body, and are solely responsible for all human movement. Muscles are categorised into three different types: Skeletal muscle which is the only muscle tissue that is voluntary. Smooth muscle which is found typically in hollow organs, its main function is to propel objects. And Cardiac muscle which is found only in the heart, its function is involuntary and pumps blood throughout the heart.
Nervous	<ul style="list-style-type: none"> The Nervous system is often considered as a master system, and is responsible for controlling and communicating thoughts, actions and emotions within the body. This system functions the fastest, in comparison to all other systems and is the most complex. This single system functions through intra-cellular communication via electrical signals.
Cardiovascular and Lymphatic	<ul style="list-style-type: none"> The Cardiovascular systems is responsible for blood circulation within the human body. The Lymphatic systems is comprised of moving fluid, vessels, lymph nodes, and organs. This system comprises of small ducts, minor glands, specialised cells, and organs throughout the body, and is capable of removing bacteria invasion and products of cellular breakdown. The lymphatic system can remove excess fluid.
Integumentary	<ul style="list-style-type: none"> The Integumentary system, designated skin, is an ever changing organ that covers the entire body. This system acts as a protective barrier against the external environment and assists in maintaining the body's temperature. The system can also gather sensory information from the environment and attempts to protect against disease.
Respiratory	<ul style="list-style-type: none"> The Respiratory system is responsible for supplying the body with oxygen, which is achieved by circulating air via the body's systems.
Sensory	<ul style="list-style-type: none"> The Sensory system ensures that the body survives, grows, develops, and is responsible for the body experiencing pleasure. This system is reliant upon sensory receptors that respond to a variety of stimuli. The general senses such as pain, touch, pressure, and proprioception are located in various locations around the body. Other sensors such as taste, smell, hearing, and sight are categorised as special senses and are located in specific areas of the body.
Digestive	<ul style="list-style-type: none"> One of the most complex systems in the body is the Digestive system. This system prepares food for use by the body, achieved by modifying the food physically and chemically (i.e. transforms food to energy), then disposes the unusable waste.
Genitourinary	<ul style="list-style-type: none"> The Genitourinary system encompasses the organs which are responsible for production, formation and release of urine. Systems include the kidneys, ureters, bladder, urethra, and all organs responsible for reproduction.
Endocrine	<ul style="list-style-type: none"> The Endocrine system encompasses a small system of organs that control the release of hormones. This system assists with the regulation of metabolism, growth, development including puberty, tissue function, and mood.

Source: Siu and Brodwin [40].

While these systems collaborate to sustain human life each system functions independently accomplishing their own unique objectives, and while an individual system can fail or be disrupted it

does not mean the entire SoS will fail resulting in the loss of life. While the human SoS is considered out of the scope of this research it is an excellent broad example of what can be defined as an SoS, in the sense that we aim to define a solution and algorithms that can be applied in the real world to varying types of SoS. We hope by quantifying the risks associated with distinct systems we can increase the SoS robustness for its entire lifespan.

2.1.3 Systems-of-Systems Associated Rewards

Various industries have been quick to integrate systems and develop SoS not only because of the advances within ICT, but also as SoS provide many benefits to organisations within varying industries. They not only allow for complex objectives to be fulfilled which the distinct systems could not fulfil on their own, they also allow for new and aging technology to be integrated thus can increase the lifespan of infrastructures. Another factor for their development was that SoS allow for processes to become automated, these processes are highly complex, and could not be fulfilled by a single distinct system nor could they be processed via the human factor, and often are time critical.

These types of infrastructures also allow distinct systems to be integrated forming larger extended system networks which not only have the capability to increase system robustness, performance, and reliability; they can also reduce the financial cost of operation and maintenance, as they provide a platform that allows physical, cyber, and human elements to be combined, alongside aging and new assets. Advantageously, all this can be achieved while each of the distinct systems maintains their independent operation and management, allowing for the distinct systems to not only function as part of the SoS; they can retain their independent operation, or could even collaborate with a differing unrelated SoS.

Furthermore, SoS are highly beneficial as they provide the means for continuous execution over extremely long durations and via many evolutionary cycles, these infrastructures have the capacity to adapt to various unexpected situations making these types of infrastructures more robust than distinct systems. Unfortunately when integrating distinct systems, emergent behaviour can manifest and while its negative effects can be highly problematic, its positive influence can be highly beneficial to the robustness of the infrastructure, as it allows distinct systems to modify their operation and adapt to fulfil objectives should malfunctions appear in other systems, hence it prevents the entire SoS failing and allows for objectives to be met. This behaviour can also allow for systems to be dynamic and have the capacity to adapt to unexpected and unanticipated situations.

The ad-hoc nature of SoS is to exploit interconnected services and infrastructures which are dispersed and interconnected via a variety of communication links, allowing for the distinct components to 'pool resources' and data, in order to fulfil identified objectives.

Smart Cities for example, have formed and their services advanced via the adoption of the Internet of Things (IoT), Wireless Sensor Networks (WSN), and other ICT solutions, which were amalgamated with their existing city infrastructures including those deemed critical. These heterogeneous networks encompass a variety of devices or ‘things’, all of which have varying computational power, energy supplies, and component and software configurations. Integration means collaborative devices are reliant upon the generation and distribution of data across the infrastructure for use by its assets and services, and data access control and data security has become one of the most fundamental challenges for Smart Cities, i.e. SoS [38].

2.2 Systems-of-Systems Challenges

When combining distinct systems major challenges must be faced, if merged incorrectly then integrated systems could be combined with disastrous results, systems could be insecure, vulnerabilities could remain unidentified, and systems could be left vulnerable to security attacks. Similarly, systems could become unstable, system wide crashing could occur, systems performance could diminish, and systems might fail to meet required objectives [3].

2.2.1 Systems-of-Systems Associated Characteristic Challenges

Using the characteristics defined by Maier (operational independence, managerial independence, evolutionary development, emergent behaviour, and geographical dispersion) [16] [41], we categorise the challenges and potential vulnerabilities which impede and expose SoS, and provide real world examples of SoS which have failed to overcome the associated challenges.

2.2.1.1 Operational Independence

Challenges attributed to operational independence can include:

- Distinct systems maintaining their independent operation.
- Systems established with unique policies.
- Systems are compiled using varying components, software, and security.
- Systems can be obliged to simultaneously function on objectives outside of the SoS with which they have a collaborative relationship, and can form collaborative relations with other unrelated SoS.

These challenges if not addressed can cause incompatibilities to develop causing conflicts within protocols and technology, systems security, operations, and can impact the ability for systems to meet objectives. Considerations must also be given to the network's security as the system with the weakest security can expose the entire SoS to potential attacks, due to varying security and components. In addition, should issues arise there can be difficulties with coordinating detection and response to issues [16].

Real world example – The disaster of the Mars Climate Orbiter demonstrates failure due to operational independence. The structure was destroyed in the atmosphere of Mars, and its loss can be directly attributed to the sheer complexity of the integrated systems, its independent system development, and failings with collaborative testing, which did not identify a fatal flaw in regards to navigational measurement [42] [43]. Fortunately distinct systems are not affected by operational independence unless they have been integrated within an SoS [16].

2.2.1.2 Managerial Independence

Challenges attributed to managerial independence can include:

- Distinct systems can maintain and/or prioritise their own objectives.
- Each collaborative system can be independently managed.
- Distinct systems can be altered via the management of other systems.
- Consultation is not required, meaning system management can add, remove, or update systems without consultation.
- Difficult to detect and respond to issues and security.

These challenges if not addressed can result in altered systems being unable to fulfil desired functions and objectives either independently or as part of the SoS. Should any alteration to security occur then systems can become vulnerable to attacks, and any alterations to systems or their security can cause conflicts to arise, thus impact or alter system operations. These issues can result in none of the collaborative organisations having the ability to control the SoS. Responding to these vulnerabilities and detecting them would be highly challenging, mainly due to the large number of collaborating systems which form each SoS [42] [44].

Real world example – The problematic open day of Heathrow Terminal T5 demonstrates failure due to managerial independence. The baggage handling systems failed and 68 flights had to be cancelled, which was directly attributed to one collaborative organisation delaying construction and providing equipment late, resulting in untrained staff and untested systems. Collaborative organisations also

failed to establish crisis management, consequently as the SoS failed, response was hampered as it could not be identified which systems were involved, what solutions should be implemented, nor which staff were responsible [44].

2.2.1.3 Evolutionary Development

Challenges attributed to evolutionary development can include:

- SoS can be deployed without being fully formed.
- Continual evolution of the SoS as new requirements identified and objectives are met.
- Functions and components can be phased in and out while the SoS remains operational.

These challenges if not addressed can cause an increase of emergent behaviour within the SoS environment. Systems are also more vulnerable to unknown and unpredicted security attacks, and during the operation of the SoS, changes could be made that impact collaborative systems' objectives and their system components' ability to function [16] [45].

Real world example – The problems of the US Coast Guards acquisition program Deepwater is a good example of an SoS being impacted due to evolutionary development. After the tragic events of 9/11 Deepwater was forced to re-evaluate its objectives and execute new requirements, forced partly due to new government legislation. Requirements included increasing security, incorporating chemical, radiological and biological defences, and increasing communication with external agencies. Because evolutionary development was thrust upon the project, Deepwater struggled with rising costs and delays, with the US Coast Guard eventually taking control of the program to salvage projects that still showed promise [45] [46].

2.2.1.4 Emergent Behaviour

Challenges attributed to emergent behaviour can include:

- Emergent behaviours surface after the SoS has been deployed.
- Identifying the system(s) responsible can be challenging.
- Identifying who should respond and who should find and implement a solution can be highly problematic.
- Ensuring mechanisms are robust to monitor the entire SoS in near real-time to ensure security breaches, system misbehaviour, and emergent behaviour is identified and reported effectively

Should emergent behaviour develop within the SoS then systems can quickly become unpredictable, fail and repeatedly crash, severe disruptions can impact systems' performance and their ability to fulfil objectives both inside and outside of the SoS, and severe security vulnerabilities could also arise [16]. While emergent behaviour is a major challenge to be faced, positive emergent behaviour can also occur within SoS. It can allow SoS to become more robust with systems altering their operations to ensure objectives are met should other systems become incapacitated. Research continues in this area to see if this behaviour can be identified and guided to harness its capabilities but is outside of the scope of our work [31] [47].

Real world example – The chaos caused by NatWest Bank, Ulster Bank and Royal Bank of Scotland is an example of emergent behaviour causing an SoS to fail. In the summer of 2012 these three banks rolled out a software upgrade resulting in unpredictable abnormal behaviour developing, when the SoS failed, customers were denied access to the entire bank's resources, with many accounts inaccessible for several days [48].

2.2.1.5 Geographic Behaviour

Challenges attributed to geographic behaviour can include:

- Systems can be governed by different laws and regulations.
- Systems are reliant upon networking capabilities to allow data to transfer between systems.
- Data security must be heavily considered between collaborative systems.
- Language barriers and time zones can be a hurdle.

When SoS are geographically dispersed with collaborating systems being located in different jurisdictions and countries, local laws and policies must be considered and applied. What is permitted in one country could be considered a crime in another, hence system location can directly impact how components function, security is applied and upheld, and can affect a system's ability to meet objectives [16]. Collaborating globally means different languages, time zones, and jurisdictions can delay and hamper collaborating efforts between managers, delays can prevent objectives from being fulfilled, reduce system performance, and cause weaknesses in security. Language can also impact the ability of managers to learn collaborators' systems, as often the complex manuals describing the systems are developed in their native language and use local dialect and slang terms [42]. A crucial challenge with geographical distribution is distinct systems' heavy reliance upon networking capabilities, more accurately distinct systems only share data to collaborate and fulfil objectives. If data cannot be transmitted between components then the SoS could fail to meet its objectives. Hence, ensuring data flow is vital, or data flow is at risk of becoming a single point of failure.

Real world example – The disastrous rescue attempt of AirFlorida Flight 90 in 1982 is an example of an SoS failing due to geographical dispersion. Federal, State and local agencies struggled to communicate, as systems had been developed independently, also contacting and integrating systems between agencies within other jurisdictions proved difficult. The delays severely hampered the rescue attempt resulting in only 5 people surviving the crash [49].

2.2.2 Single Points of Failure Within Systems-of-Systems

A Single Points of Failure (SPoF) is a component within a networked infrastructure, which in the event of its failure prevents large sections or the entire infrastructure from communicating. These SPoF can be responsible for additional failings rippling across the infrastructure causing both partial and full cascade failure, meaning SoS will fail to meet objectives.

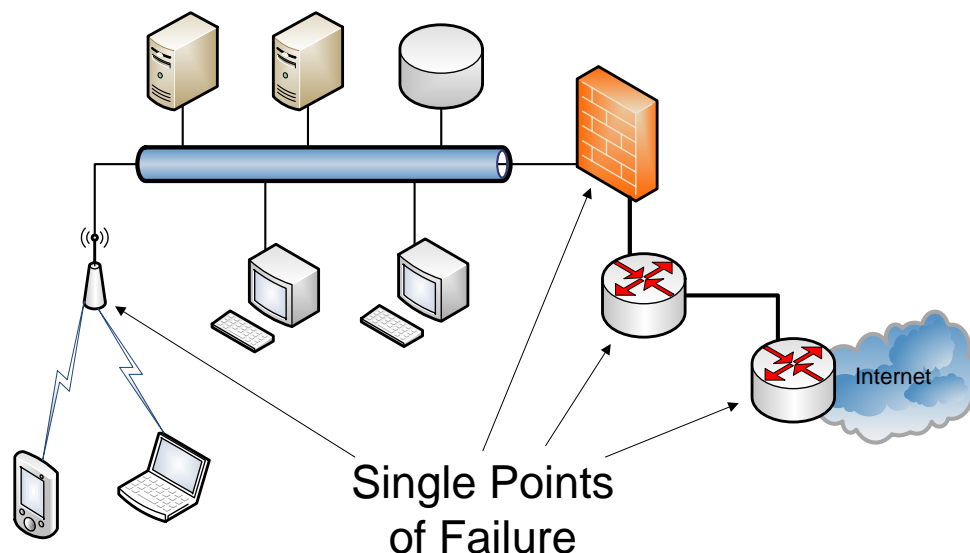


Figure 2.5. Schematic Representation of Single Points of Failure

SPoF can also be attributed to improper systems configuration, poor design, and incorrect component and system implementation. Under all categories defined by Maier [24], a common element that exhibited problematic challenges and vulnerabilities was that of data. The uncertainty associated with this single element can be a significant cause of risk within SoS. For example, our research clearly identifies that data as a whole has the potential to be a SPoF. If data cannot be created, stored, or transmitted within the SoS environment, then the collaborative infrastructure has the potential to fail in its entirety. It could be described that data has a life cycle, it is created, stored, used, shared, archived, and then destroyed [50]. Yet data does not instinctively flow between components forming the SoS, many challenges have to be overcome to allow systems to be integrated and collaborate upon objectives, many of which were described in Section 2.2.1.

With the advancement of ICT and the Internet, SoS organisations are no longer isolated, instead they have increased their connectivity to become dynamic, and become reliant upon the transmission of critical data over these links (i.e. communication assets have become the backbone to allow systems to collaborate and form SoS). This dependence creates cyber interdependencies within the infrastructure which increases elements of risk. Electronic information links also provide a new means of attack as they provide multiple access points (e.g. intranets, phone lines or Internet) that prior to system integration did not exist, which attackers with little knowledge or skill and using freely available tools can exploit. Malicious attackers take advantage of these new connections by launching attacks and eavesdropping upon data as it is transmitted between components. Information such as authentication credentials, credit card transactions, email content, and control commands, can easily be exposed by listening to transmissions. Once an attacker has access to data transmissions they can launch simple attacks directly against data such as altering, corrupting, destroying or injecting false data, or they could simply impede the flow of data. Any alteration to the transmitted data packets could impede an SoS from fulfilling its objectives, thus creating a single point of failure [20] [21].

Attacks can include access control, injection and execution of malicious software or data, object reusability, masquerading attacks, sniffing, snooping, and DDOS attacks. While managers can implement several solutions in conjunction in an attempt to secure systems and ensure data flow, integrity, security and availability, there is currently no single solution that guarantees total security. Incorporating security features within an SoS does not ensure that data communications will be secure. Features currently employed include intrusion detection and prevention systems, firewalls, virtual private networks, content filtering, antivirus solutions, access control, and cryptography and key management. Implementing such features can also be time consuming as not all security solutions are automated, instead hosts must be correctly configured, policies must be periodically updated, and every integrated system must be secure or has the potential to be a point of attack [20] [51] [52] [53].

While it could be perceived that data is at its most vulnerable state during its transmission across the SoS, we look at data in its entirety and recognise there are many weaknesses that can truly affect data within SoS. Data is not just at risk from malicious attacks by outsiders, data can be at risk from legitimate user error, components within the SoS, the physical structure of the networked systems and Internet, as well as natural disasters [54].

For example, data can become corrupt via system components during its creation and processing; also malicious attackers from within the SoS can alter, corrupt or delete data just as easily as a legitimate user's unintentional action(s) [55]. Data stored in components that are deployed in remote areas, unguarded and accessible by attackers, potentially can suffer power loss, be damaged by natural occurrences, have difficulty connecting to SoS due to loss of connectivity, and attackers can remove components or destroy them. This can affect the flow and availability of data. Malicious attackers can gain physical access to stored data within components, along with jeopardising its integrity, this

access also gives attackers the opportunity to inject false data into the SoS [20] [21]. Attackers also exploit highly publicised vulnerabilities associated with open standards, off the shelf hardware, and software, and use freely available tools to launch directed attacks, eavesdrop upon network traffic [20], and in the future could exploit gateways that were previously isolated and do not support security features such as authentication, confidentiality, integrity, and data privacy [56].

Furthermore, as infrastructures are reliant upon data being transmitted across different types of communication links, this increases complexity within the infrastructure, and introduces the risk of cascading failures between collaborating systems.

These are just a few of the challenges we have identified during our research that impact data directly, and have the potential to heavily impact the resilience and security of SoS. We also clearly identified the challenges (discussed in Section 2.2.1) associated with operational independence, managerial independence, evolutionary development, emergent behaviour, and geographical dispersion that can directly prevent data from being created, stored, or transmitted, thus can become SPoF and prevent SoS functionality and objective completion.

To overcome some of the challenges which relate to SPoF many organisations are turning to third parties (e.g. migrating to the Cloud), in an attempt to circumvent some of the vulnerabilities and to assure data integrity, availability, and security. While third parties offer services as a low cost solution they unfortunately just provide a new platform for malicious attackers to exploit. More concerning is the fact that organisations once they have outsourced their services rely heavily upon data flow to function, and these operations become a weak link thus creating a new SPoF that can impact the organisation and its ability to function [20] [21] [57].

2.3 Systems-of-Systems Security Challenges

Securing cyber networks is of vital concern, even in its infancy malicious attackers targeted ICT platforms. In 1988 one of the first distributed computer worms was identified and reported called the Morris Worm. Decades later new directed attacks are still being developed, impacting ICT based networks with differing levels of complexity and impacts. Table 2.3 presents a small example of known attacks that have occurred in each decade since 1988, demonstrating that attacks have increasingly become more targeted and sophisticated.

Table 2.3. Example Security Attacks

Attack	Impact	Date
Morris Worm	Multi-platform worm that impacted approximately 6,000 devices, which relates to roughly 10% of the computer devices connected to the Internet [58].	1998
Melissa Virus	Email worm impacted over 300 organisations and 100,000 computer devices connected to the Internet [59].	May 1999
MyDoom Worm	Email worm infected 60,000 devices in less than 8 hours; at its peak the infected hosts attempted 30,000 queries per second [60].	January 2004
Stuxnet Worm	Worm targeting industrial control systems, impacting at least 14 industrial locations in Iran [61].	June 2010
WannaCry Ransomware	Ransomware virus software attacked over 200,000 victims encrypting their data and demanding a ransom payment, impacting hospitals, organisations (including government), and over 150 universities [62].	May 2017

Typically attacks exploit vulnerabilities within the networked infrastructures, taking advantage of both unknown (zero-day attacks) and known vulnerabilities. With the wide adoption of the Internet and ICT, organisations have been quick to take advantage of the interconnectivity and services they offer, extending their unconnected infrastructures into widely distributed collaborative systems. Increasing connectivity to systems means they provide new entry points to previously secure and inaccessible systems, and they have increased their attack surface and formed greater sized networked infrastructures via their adoption of ICT and having formed new collaborative relations with third parties.

In the context of SoS, these infrastructures are exposed to the same dangers and risks as traditional cyber based networks, however, the complexity, dynamic nature, and size makes securing them more taxing. Traditional security methods are difficult to adapt and integrate within SoS, struggling to identify vulnerabilities that expose the entire SoS and leave systems insecure. Often employing countermeasures to systems after an attack has occurred and after there have been considerable delays.

Traditionally it was easy to physically secure systems, but increased connectivity and deploying networked infrastructure across towns, cities, and globally (e.g. Smart Cities, IoT and WSN) means malicious attackers have opportunities to gain physical access to distributed devices, and remote access across the Internet. Therefore it is vital to secure SoS in order to restrict unauthorised access, without impacting the functionality of the systems or causing delays.

Additionally, as integration continues and organisations continue to both form new collaborative relations and outsource their ICT to third parties, security challenges in regards to privacy will need to be considered, along with laws and legislation that must be upheld to geographically dispersed SoS. As adapting security requirements, and differences between the physical, software, and configurations will all unduly impact the application and effectiveness of applied security techniques, other factors

that can greatly impact the security of SoS include interdependency, complexity, and cascading failure.

The challenges that impede SoS security are characteristically dissimilar to large enterprise ICT. For example, enterprise systems and large extended networked infrastructures typically have a distinct and identifiable management structure, responsible for managing the networked systems' security with no competing interests, and typically maintain sole control of an organisation's infrastructure, unlike SoS, that characteristically have no top layer management to oversee security, identify and evaluate risks and vulnerabilities associated with the collaborative environment, nor their consequences. As a result testing and evaluating enterprise systems would be more attainable under the control of a single management, unlike SoS which struggle to identify those responsible for the implementation, testing, and evaluation of the SoS security, along with the security of the distinct systems.

Enterprise systems and processes are generally well established and are not under continuous evolvment and development, however, as required they can be easily adapted or deviate from requirements as an organisation identifies new needs, thus changing their functions and priorities. Whereas SoS would struggle to deviate due to their composure and reliance upon specific assets and service provided by distinct systems, and due to the collaborative systems remaining under control of their distinct management. Unlike large systems and enterprise ICT, SoS are further impeded as distinct collaborative system requirements can conflict with the objectives of the SoS. Similarly, SoS could struggle to meet objectives and remain operational while balancing the needs of these individual systems, and may encounter operational limitations and failings should distinct systems not be available due to their integration with unrelated SoS. In Section 2.3.1 below, we discuss factors which can impact SoS security, and in Section 2.4 we overview the limitations of SoS risk analysis.

2.3.1 Factors Which Can Impact Security

2.3.1.1 Interdependency

Cyber assets allowed organisations to become automated, allowed new and aging technology to be integrated, provided powerful tools, a large adaptable platform, and improved efficiency. Unfortunately, this also led to the introduction of cyber interdependencies, and introduced new vulnerabilities into the majority of collaborative infrastructures impacting cyber security [63]. It must be stated though, that while distinct systems are highly complex and suffer with varying challenges and vulnerabilities, the risks associated with interdependency only arise when distinct infrastructures collaborate. *“An interdependency is a bidirectional relationship between infrastructures through which the state of each infrastructure is influenced by or correlated to the state of the other”*, as defined by Rinaldi [64].

Rinaldi and Peerenboom surmise that “*interdependencies can cause risk in one infrastructure to be a function of risk in another. If infrastructure i depends on infrastructure j, and j has a high risk of failure, then the likelihood of i being disrupted or failing is correspondingly higher than if i were independent of j*” [63]. These interdependencies can be categorised under four distinct types based upon their linkages, which are physical interdependency, cyber interdependency, geographic interdependency, and organisational interdependency as shown in Table 2.4, while Lee et al. [65] acknowledge five types of interrelationship which are input dependence, mutual dependence, shared dependence, exclusive or dependence, and co-located dependence, summarised in Table 2.5.

Table 2.4. Types of Interdependency Based Upon Their Linkages

Type	Description
Physical Interdependency	<ul style="list-style-type: none"> • Infrastructures which rely upon the physical transfer of resources between each other, specifically one element relies upon the output of another as its input to fulfil its own objectives. • Infrastructures are classed as physically interdependent if they cannot function without these physical resources. • These infrastructures directly influence each other, hence these organisations are physically interdependent and should an issue arise in either infrastructure then the failure could rapidly ripple across and impact heavily upon the other.
Cyber Interdependency	<ul style="list-style-type: none"> • Infrastructures transmit often critical data over electronic information links, thus one infrastructure is reliant upon the electronic output of another infrastructure as its own input in order to fulfil its objectives. • Infrastructures are classed as cyber interdependent if they cannot function without these digital links. • Organisations which have embraced and incorporated cyber assets into their infrastructures have become heavily reliant upon those cyber systems, more specifically they have become reliant upon the data which is created, stored or transmitted by these assets.
Geographic Interdependency	<ul style="list-style-type: none"> • Organisations that are situated in the same close proximity should an incident occur such as an explosion, can be heavily impacted due to the events and influences caused by the disturbance resulting in abnormal operations. • However they are only geographically interdependent should that incident impact or cause abnormal operations in each of them simultaneously because of that initial specific incident. • This type of interdependency is not physical or cyber interdependent, it is solely down to the infrastructures being geographically located within the same proximity.
Logical Interdependency	<ul style="list-style-type: none"> • Infrastructures which are interdependent yet these dependencies are not physical, cyber, or geographically interdependent are classed as logically interdependent.

Source: Rinaldi et al. [63], Rinaldi [64], Engineering the Future, The Royal Academy of Engineering, Infrastructure Interdependencies Timelines Report [66].

Research determines that interdependency between systems adds to external risk, and due to the sheer size and complexity of these SoS at this current time it is not possible to gain a complete overview of these infrastructures. Consequently, those in industry do not have the capability to understand or identify every interdependency and dependency within these large complex infrastructures [1]. This only reinforces the need for continued research and development.

BT is an excellent example of this. BT developed site recovery plans (over 5,500), invested in mobile exchange recovery units (over 100), and developed emergency operations management centres and mirrored sites within the UK. These assets and responses were developed in an attempt to manage

risks within their network, as it is not currently possible to map or understand every critical link that is vulnerable due to the sheer size and complexity of their infrastructure [1]. Despite this substantial investment they still struggle to manage risk resulting in critical failures still occurring. Unfortunately, many external organisations depend on infrastructures such as BT's communication assets to provide the backbone for their infrastructure to allow for collaboration and control of their systems.

Table 2.5. Types of Interrelationships Between Collaborative Infrastructure Systems

Type	Description
Input Dependence	<ul style="list-style-type: none"> • Infrastructures are reliant upon at least one output from another distinct system or its services in order to meet either their own objectives or to provide additional services.
Mutual Dependence	<ul style="list-style-type: none"> • Systems within a collaborative relationship are dependent upon each system with which it has been integrated in order to meet at least one of its objectives. • For example, if there are two systems within the collaborative relationship. Mutual dependence is when system A is reliant upon the output from system B for its own input, while system B is reliant upon the output from A for its input.
Shared Dependence	<ul style="list-style-type: none"> • Physical components or processing is shared by several systems within the collaborative environment in order to provide services.
Exclusive Or Dependence	<ul style="list-style-type: none"> • Only a single service out of two or more services can be provided by an infrastructure. • Exclusive Or can potentially occur within a distinct system or via two or more systems.
Co-located Dependence	<ul style="list-style-type: none"> • Modules within two or more systems are geographically located within the same region.

Source: Lee et al. [65].

Interdependency also heavily impacts security within interdependent cyber infrastructures. Should security be weak in one infrastructure then the level of risk in the other collaborative infrastructure increases, while it reduces the level of security in other dependent systems [63]. This means cyber security becomes a particular challenge for these types of infrastructure, and as society integrates more systems making them even more interdependent, the risks associated with security will also increase and become more complex.

What must also be considered is that in the future these challenges may have to be considered and overcome on a global level, as organisations are becoming more geographically dispersed with assets being located in different jurisdictions and countries which have differing laws and regulations. Generating more interdependencies might allow for increased interconnection and improve reliability; however, it develops and increases both complexity and risks associated with interconnection.

Figure 2.6 depicts the interdependent links that can form between essential infrastructures within a Smart City environment, their reliance upon services and assets by others, and the Smart City's dependence upon communication assets in order for it to establish and form a collaborative environment.

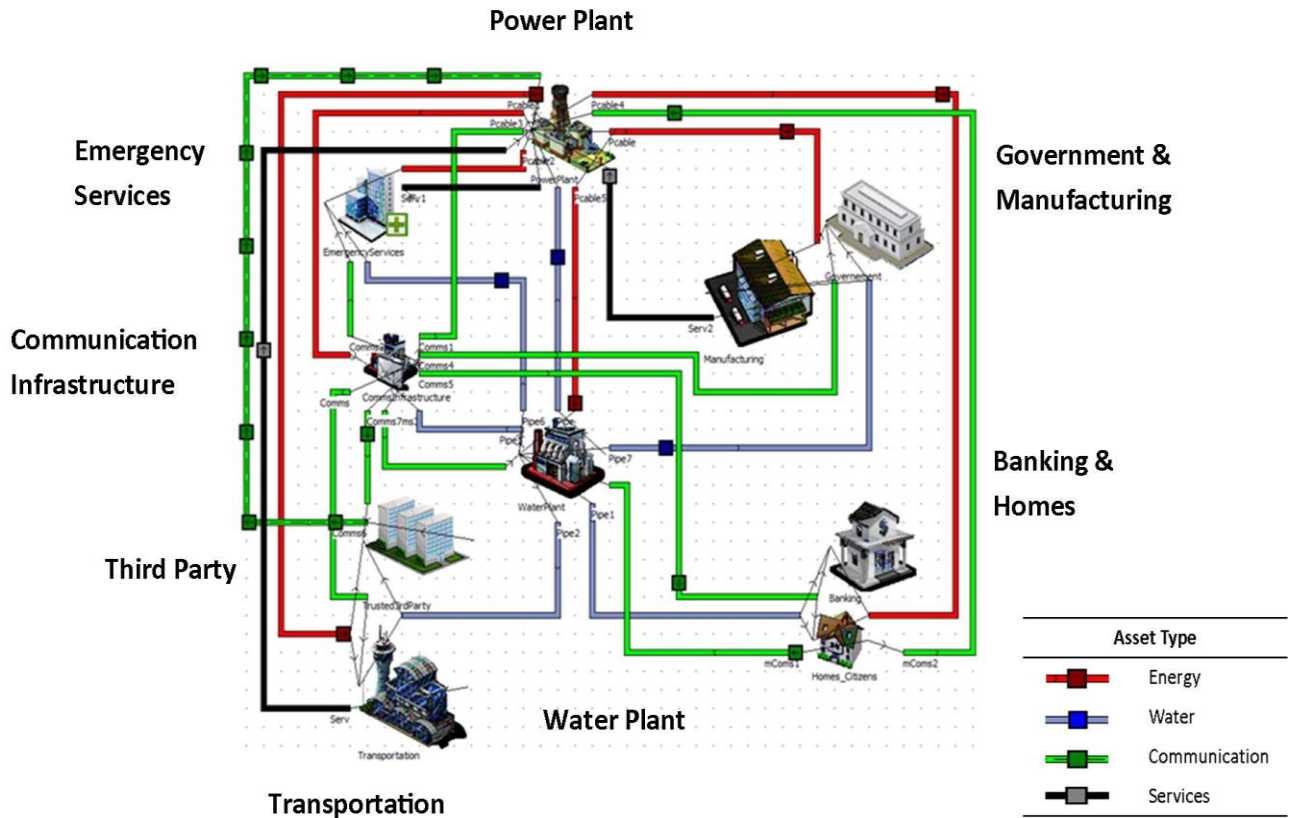


Figure 2.6. Schematic Representation of Interdependent Relations in a Smart City

2.3.1.2 Complexity

There is no consensus regarding a definition for complexity. In 1983 the IEEE Standard 729-1983 [67] defined complexity as “*The degree of complication of a system or system component, determined by such factors as the number and intricacy of interfaces, the number and intricacy of conditional branches, the degree of nesting, the types of data structures, and other system characteristics.*” By 2010 IEEE amended their definition to keep up with new technology and research, and because of a deeper understanding in regards to complexity. Thus, IEEE Standard 610.12 [68] defines complexity as “*1. The degree to which a system’s design or code is difficult to understand because of numerous components or relationships among components 2. Pertaining to any of a set of structure-based metrics that measure the attribute in (1) 3. The degree to which a system or component has a design or implementation that is difficult to understand and verify.*”

Kopetz [69] classifies complexity under two distinct categories which are Complexity as a Property of a scenario and Complexity as a Relation. Complexity as a Property of a scenario consists of Structural Complexity which focuses upon the actual topology of components and also the links between components, and Dynamic Complexity which focuses upon the behaviour of the components and their dynamic interactions. Complexity as a Relation consists of Cognitive Complexity which is the

relation between the scenario and an observer, and Socio Political Complexity which is the relation between a scenario and society [69]. While Asprou et al. [70] confer in regards to modelling, there are two distinguishable main interdependent complexity components which are Structural Complexity, and Dynamic/Operational Complexity.

As defined by Dr Kaplan SoS are “*Uncertainly unbounded*”, there is a distinct lack of control, as it cannot be realistically defined as to when the SoS will no longer be developed further and grow in size. Even at the time of development and deployment SoS objectives have not been fully defined, meaning often they become operational without knowing their true purpose or lifespan. Complexity is further increased when we have to consider that these infrastructures will be not only be continually evolving and perhaps in constant operation, but also functions will be phased in and out, new policies and standards will be forced upon the systems forming the infrastructures, software and hardware upgrades will periodically occur, and systems will be required to not only be backward compatible but they must also be forward compatible to allow for integration between aging legacy systems and emerging new technology [71].

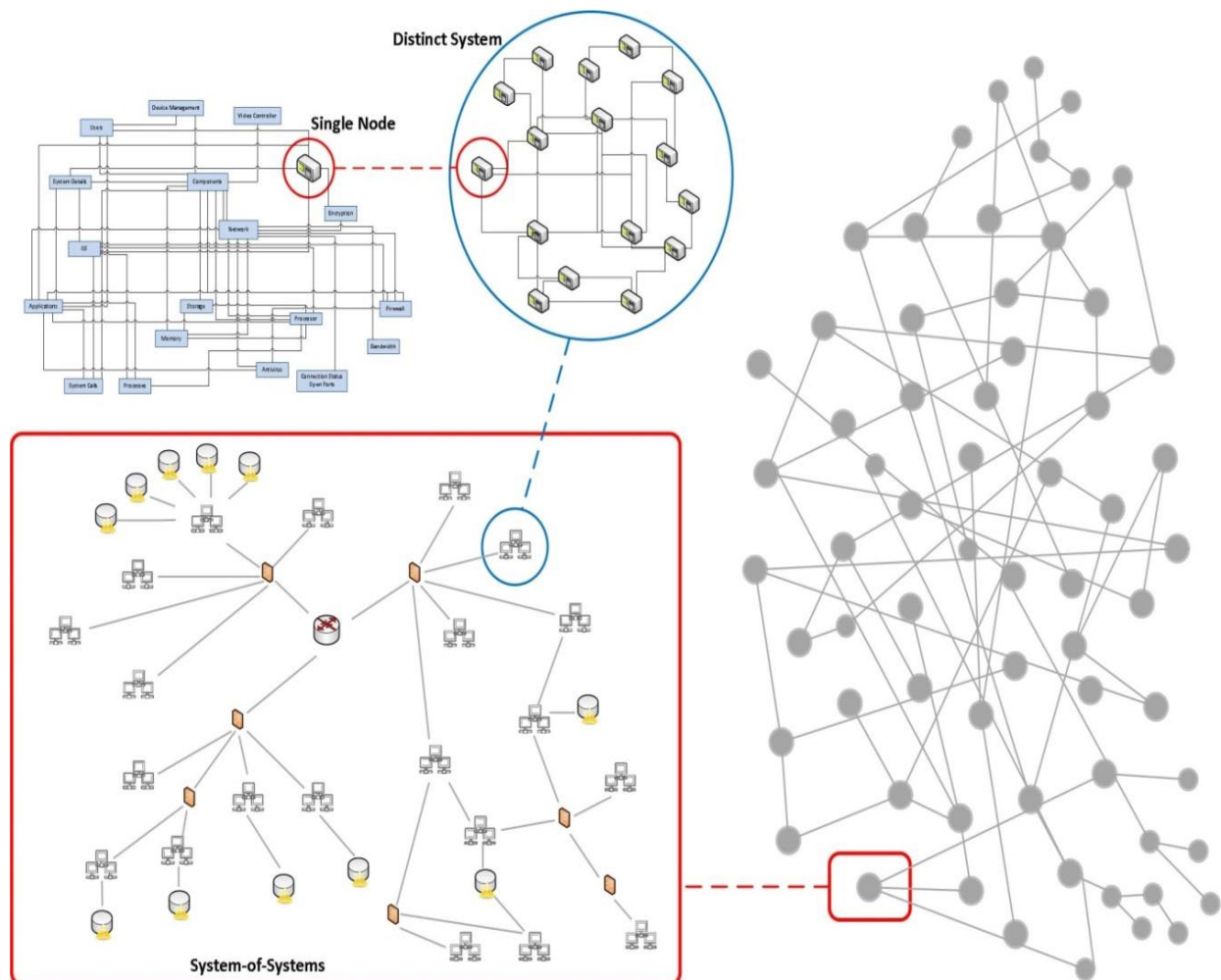


Figure 2.7. Complex Multi-Level Systems-of-Systems

Integrating systems is highly complex especially as there are often technical issues to overcome. Figure 2.7 depicts the complexity of a multi-level SoS, providing a schematic representation of the complexity associated with a distinct component, and the complex relationships that form due to their integration into larger extended networks and SoS.

Legacy systems emerged in varying industries, often unplanned and many developed as single standalone systems. As industry pushed the boundaries of legacy systems, new complex requirements were demanded from them. At the time of their conception, it was never conceived that legacy systems would be integrated or used for roles outside of their remit. Furthermore, the security of these systems was not fully investigated to ensure issues would not develop after amalgamation [72].

While some systems have the ability to share data and functionality, this was often overlooked or never considered important. Legacy systems which were not initially designed to fulfil many of the requirements now demanded from them, struggle and are affected by the demand for interoperability, performance, security and usability. These infrastructures are often heavily reliant upon the transfer of data between systems, however, data is critical to these infrastructures, plus often application dependent and not suitable nor designed to be shared collaboratively [72].

Furthermore, SoS are integrated using varying distinct systems that retain their unique management. Managers are not required to inform other collaborating parties when upgrades, modifications or repairs take place. Meaning modifications within interdependent systems often are implemented without being synchronised and without suitable analysis, consequently complexity significantly increases. Vendors producing products and applications are not currently required to inform other vendors of specific operations, configurations, non-compatible standards or conflicts. Meaning when organisations use assets produced by two or more vendors, there is a significantly high probability that incompatibilities will form between the collaborating systems. Thus, interoperability issues arise causing a higher degree of system integration complexity [71]. As complexity derives from the number and type of relationships between the collaborating systems and components, and also originates between the actual systems and their environments, complexity makes it difficult for research and industry to design and model the architecture and security of these infrastructures [73].

2.3.1.3 Cascading Failure

While interconnection has the ability to improve reliability and regrettably increase complexity, it also allows for emergent behaviours to arise which can result in both positive and negative influences upon collaborative systems. As society is reliant upon SoS, should negative emergent behaviour arise to the point that it is directly responsible for an incident occurring, then its effects have the potential to result in catastrophic consequences. Additionally, due to the tightly coupled bonds between interdependent

systems which restrict links and reduce flexibility between these systems, it means in the event of failure there are no alternatives.

Therefore, these incidents have the capacity to ripple from one infrastructure to another resulting in both direct and indirect effects, with the capacity to disrupt the SoS ability to fulfil objectives (see Figure 2.8). This cascading effect can impact both locally and on a larger geographical scale, even having the capacity to impact on a global level. Cascading failures have the capability to directly influence or hinder organisations, governments, and society, and when systems are restricted and inflexible then these cascading failures have the potential to exacerbate incidents further and can even loop back to the originating disruptive system [63].

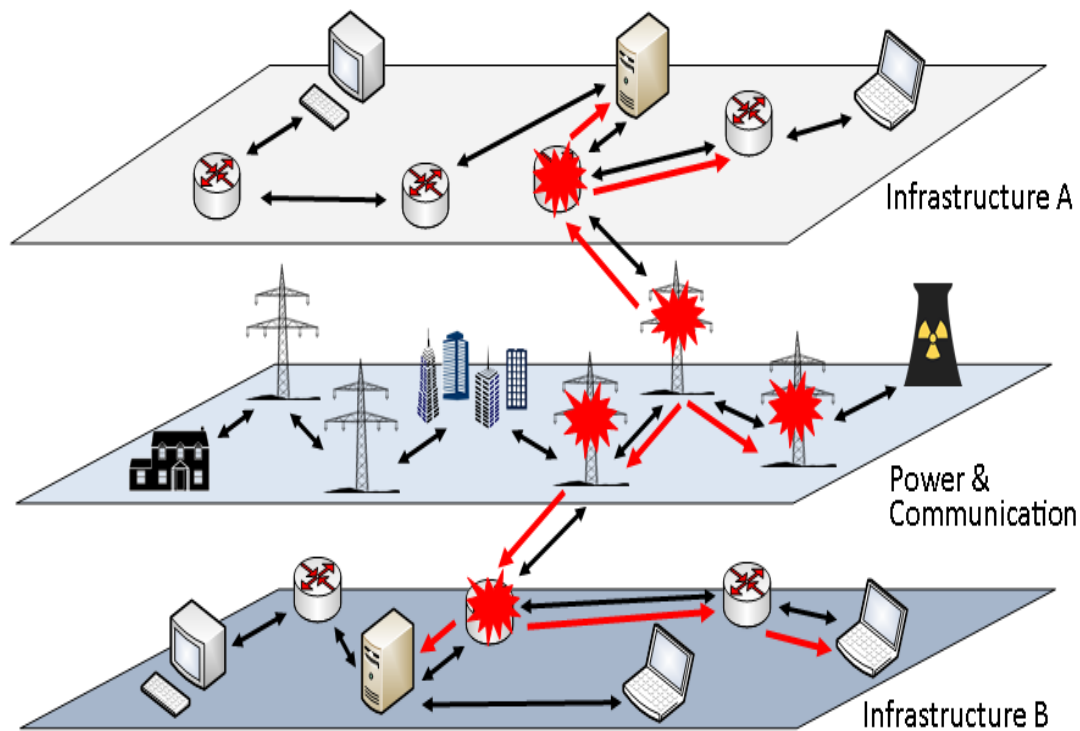


Figure 2.8. Schematic Representation of Cascading Failures

Interdependencies increase the risk of vulnerabilities, disruptions, and failures, meaning feedback loops and the complex architectures which form SoS, initiate and propagate issues that are unusual and difficult to predict. If SoS can be designed to be more robust and secure in the event of failure the risks will be mitigated and consequences lessened. Rinaldi et al. [63] summarised interdependence-related disruptions as cascading, escalating, and common cause failures, outlined in Table 2.6.

Table 2.6. Types of Interdependence Related Disruptions

Type	Description
Cascading Failure	<ul style="list-style-type: none"> This is when a significant failure occurs within an infrastructure, this failure is then directly responsible for causing a failure or disruption to a component within a second infrastructure, and consequently this effects the normal operation of the second infrastructure.
Escalating Failure	<ul style="list-style-type: none"> This is when a significant failure occurs within an infrastructure which causes disruption, this then exacerbates an independent disruption of a second infrastructure. Exacerbated disruption typically increases the severity of the failure in the second infrastructure or severely hampers and delays recovery from the disruption.
Common Cause Failure	<ul style="list-style-type: none"> This is when two or more infrastructures are disrupted or experience failures at the exact same time, these types of failure occur due to the infrastructures simultaneously being impacted by a common cause. Common causes can include infrastructures being directly impacted by the same natural disaster as they are located in the same locality or the root cause is widespread.

Source: Rinaldi et al. [63].

A case study that accurately demonstrates the severity and long term effects of cascade failure is that of the Fukushima Daiichi nuclear disaster, that remains an active risk to society. The disaster occurred on the 11th March 2011, when a massive earthquake struck Japan, which was followed by a tsunami which caused water to penetrate the nuclear plant. As water penetrated the site it caused power loss to many of the systems within the site, as a direct result this caused the cooling systems to fail resulting in explosions. These explosions caused further damage to systems, then the nuclear meltdown of three reactors. This allowed for radioactive materials to leak into the immediate area, and the cascade failings and radiological contamination hampered emergency workers [8] [74] [75].

While this is a natural disaster the effects caused by the cascading failures resulted in the not only the plant's destruction, but will continue to impact society for several more decades. In 2013, radioactive material continued to escape into the locality and Pacific Ocean at a rate of 300 tons each day. The surrounding area has been declared as too radioactive for human habitation, causing entire towns, agricultural lands, businesses, and homes to be abandoned, resulting in an estimated economic loss of approximately \$250-\$500 billion US. Contamination to food has heavily impacted the agricultural and food industry with fishing in the area being banned, and public health and safety has been heavily impacted with many children being diagnosed with thyroid cancer [8] [74] [75].

Japan has also been forced to implement stricter seismic safety because of the crisis, directly resulting in several nuclear power plants remaining closed as they cannot meet the standards that were established. The continued loss of power production is also heavily impacting manufacturing, security, government, and the national economy to name but a few. More alarming is the fact there are currently 23 nuclear reactors located in the US with the exact same design as the Fukushima site, and unfortunately the pools in the US contain more spent fuel rods than then pools in Fukushima. As shown by this disaster in Japan these types of nuclear infrastructures are vulnerable to cascading

failures caused by a loss-of-coolant and have the potential to cause a catastrophic disaster, and thus far the associated risks have not been resolved [8] [74] [75].

2.3.1.4 Identified Systems-of-Systems Risks and Attacks

Based on the identified security challenges, we can see that attack vectors are more prominent in SoS. If these risks remain unresolved and unprotected then SoS could fail with dire consequences. Having analysed network risk and the awareness that the more interconnected an SoS becomes the more susceptible it becomes to network vulnerabilities and threats, from our extensive review network vulnerabilities can be summarised as [12] [76] [77] [78] [79]:

- Insecure or exposed ports.
- Unnecessary and indiscriminate enabling of services and applications.
- Improper system configuration.
- No formal configuration management.
- Inadequate anti-virus.
- Inadequate application whitelisting.
- Inadequate intrusion detection systems.
- Insufficient password and security policies.
- Insufficient account management.
- Failure to encrypt passwords.
- Indefinite passwords or shared passwords on network devices.
- Unrestricted and improper access control.
- Unrestricted or unmonitored downloads from untrusted sites.
- Ill-configured applications and programs.
- Application backdoors.
- Insufficient security policies.
- Failure to monitor logs and warning messages.
- Disgruntled employees.
- Corporate espionage.

- Inadequate training.
- Unrestricted or inappropriate security levels for legitimate users.
- Inadequate physical protection and security for components and systems.
- Inadequate authentication for internal and remote access.
- Wireless Local Area Network (LAN) technology is often used to connect devices.
- Unpatched software.
- Inadequate patch management.
- Inconsistent documentation.
- Use of vulnerable protocols.
- Redundant and inadequate firewall rules.
- Inadequate system hardening.
- Inadequate testing prior to application integration.

If the above summarised network vulnerabilities are left unprotected or are exposed by malicious attackers, our review of the associated literature indicates that the following summarised attacks can be launched [12] [76] [77] [78] [79]:

- **Application-Level Attacks** occur when attackers exploit insecure computer operating systems or applications, which have failed to be secured prior to their release. In general the complex features and functionality of the application along with the failure to incorporate security measures into the application at time of development, allow the attackers to take advantage of the insecure application, and thus evade access controls gaining control over the application or system.
- **Misconfiguration Attacks** occur when network administrators fail to adequately secure and configure their networked systems; this can be as a direct result of their inexperience and training, or due to the complexity of the networked systems, etc. Attackers can easily take advantage of misconfigurations via default accounts, unpatched applications, web pages, unprotected files, insecure directories, etc.
- **Operating System Attacks** occur when attackers exploit operating system vulnerabilities such as running services, open ports, default, settings and user accounts, etc. As networked systems continue to be developed and expanded, operating systems will only become more complex, and their services and communication access will only grow, increasing the demand and functionality of the operating system and its associated vulnerabilities.

- **DDoS** is a coordinated attack against the services and resources of a targeted system or systems. Typically these attacks are as a direct result of compromised secondary systems (e.g. botnets) that are controlled by the attacker, which prevents legitimate users from accessing the available resources and services of an organisation, by repeatedly sending identical requests which exceed the organisations available resources.
- **SQL Injection** can occur by attackers exploiting the security of a web application that allows for non-validated SQL input commands to be executed. The attacker can input SQL code via the web form input box, allowing them to gain access to the backend databases or provide them the ability to corrupt, alter or delete data.
- **Social Engineering** exposes networked systems to vulnerabilities and attacks as a direct result of the human element. Attackers will target the employees of an organisation in an attempt to retrieve sensitive information or access details. Social engineering is considered one of the most difficult types of attack to defend against, as it is impossible to place physical or software based security measures to defend against such attacks, and it requires strict policies to be in place and education for employees. Using tactics such as fear, trust, or assistance, attackers will attempt to extract information via email, phone calls, or talking in informal environments, etc., attempting to extract confidential information from naïve individuals so that they can gain unauthorised access and exploit systems.
- **Sniffers** are a program or device that allows attackers to capture data from the network's traffic. Network traffic can allow attackers to retrieve unencrypted passwords and user names, emails, files, etc., and protocols which are susceptible to sniffers include HTTP, FTP, SMTP, and POP.
- **Buffer Overflow** occurs when an attacker exploits an application that is waiting for a user's input, the attacker generates data to be inserted into the application, and this data is larger than the temporary data storage assigned and overflows into other buffers. This type of attack can escalate an attacker's privileges, or could corrupt or overwrite data that was stored within the buffer. These attacks can be categorised into two different types of attack which are Heap-based which are difficult to execute, and Stack-based which are more commonly conducted.
- **Password Cracking** can be a simple attack where an attacker finds a username and password written underneath a keyboard, to simple attempts to find information by searching through a user's rubbish to find suggestions of usernames and passwords. More sophisticated attacks can use techniques such as brute force, dictionary, word list substitution, pattern checking, and hybrid attacks, which allow attackers to retrieve the relevant data from computer devices' memory for example, or to decrypt passwords allowing attackers to gain unauthorised access to systems with authorised credentials.

- **Phishing Attacks** occur when attackers seek to gain information from victims by impersonating trustworthy individuals or organisations. Typically these types of attack are attempting to obtain user credentials, bank details or credit card details, and will be conducted via emails or instant messages that direct individuals to fake websites that have the appearance of genuine sites.
- **Viruses and Worms** are malicious software programs developed by attackers. A computer virus can reside in the memory of a computer device and replicate its own code and attach copies of itself into other executable code, without being identified by the computer and unknowing user. Viruses can also alter their code to remain unidentifiable, and can encrypt themselves and alter disk directories in an attempt to conceal themselves. A computer worm does not require human assistance to propagate and infect other computers. Unlike viruses worms can replicate themselves and spread in order to infect entire networked systems, but do not have the ability to attach to other programs.
- **Trojan** is a malicious program that has the appearance of a legitimate application. When executed, Trojans perform malicious activities on the computer device, and can allow attackers to disable software and security, provide remote access to the device, send data, etc. Trojans can be introduced into a computer device via physical access, applications, emails, untrusted web sites, fake programs, email attachments, etc.
- **Spamming** is when attackers gain access to a large numbers of email addresses and send the same message to all of them. While spamming can have legitimate use in cases such as organisations advertising, spam emails can also contain malicious viruses and Trojans, allowing them to gain access to the devices as the malicious code infects and alters systems.

The effects of such attacks could result in the following consequences [12] [76] [77] [78] [79]:

- Loss of human life.
- Result in widespread panic.
- Economic impact, with local, national, or global effect.
- Impact to the reputation of organisations attacked.
- Catastrophic system failures.
- Impact on the quantity and quality of services and production of goods.
- Loss of intellectual property.

These vulnerable network issues are just one categorised example of vulnerabilities, which can drastically impede SoS causing exploits and failings within these types of infrastructure. We have considerably condensed our findings and provided these examples to highlight the scale of our research area and the difficulties, which must be overcome to increase SoS security and robustness. Vulnerabilities cannot just be simply categorised as either technical issues with software or physical components, procedural and engineering errors can also contribute to vulnerabilities forming within the SoS [12].

2.4 Systems-of-Systems Risk Analysis

The International Standards Organisation (ISO) standard 31000:2009 [80] defines risk as the “*effect of uncertainty on objectives*” regardless of circumstances or domain. However, a hazard or an event should not truly be described as a risk; instead risk can be better defined as the combination or the likelihood of an internal or external factor or influence, which directly has the potential to impact or cause consequences to occur, while the organisation attempts to fulfil their objectives against an uncertain environment [19] [81].

Risk is unavoidable. Organisations will always be forced to contend with risk due to either creating or altering risk during all decision making processes or when phasing functions and assets in and out of SoS. It is vital for organisations to perceive risk during all stages of the SoS life cycle, and regrettably there will always be uncertainty involved, consequently risk can never be eliminated. Nevertheless, via a better understanding of risk and potential consequences associated with collaborating systems, risk taking could become a calculated intentional act rather than a ‘blind stab in the dark’ with a lack of perceived ideas or poor understanding [19].

Risk methodologies and assessment methods aid organisations in quantifying any identified risk and assigning it a numerical metric. This figure can then be associated with a monetary value to support decision making processes, and predicting the likelihood of a specific attack. Security risk assessment methods are categorised as either qualitative or quantitative.

- **Qualitative Risk Assessments** deliver a descriptive estimate of the risks which are identified by the method, such as assigning low or high as a value. These risks are often assigned based upon a traditional collective method such as interviews or questionnaires. These methods tend to be used by organisations who fail to perceive the risks which are associated with their specific systems, and do not have the aptitude to estimate the likelihood or impact of potential threats. Qualitative assessment methods would be highly impractical for large heterogeneous SoS.

- **Quantitative Risk Assessments** deliver a numerical estimate of the risks which are identified. Estimates are produced using a predefined formula or mathematical expressions, and are ideal for organisations that have the capability to provide estimates for their system's identified vulnerabilities. This could be achieved via risk assessment and analysis through population based attack graph modelling for example.

2.4.1 Risk Assessment and Management

According to ISO 31004 [19] risk management is “*coordinated activities to direct and control an organisation with regard to risk*”. Risk assessment has become a fundamental requirement for network security over the past several decades, as the size of networks and their connected components vastly increased along with the number of vulnerabilities. As a consequence of the increased value of digital data, criminals are utilising ICT platforms to perform sophisticated and targeted attacks to retrieve sensitive data, cause service interruptions, or cause complete unavailability to services and data. As a direct result of these attacks and potential threats, organisations have been forced to evaluate their infrastructures, specifically security requirements for the protection of their systems, services, and data. When attempting to secure their systems, organisations have a propensity to over compensate with regards to their security defence, or undercompensate, overlooking potential risks or trust issues [82].

This irregularity in judgement is due to the decision makers' (i.e. those responsible for the network's security and management) failure to understand associated risks to the organisation's systems and surrounding area, or the resulting consequences that can potentially occur on a daily basis. Similarly, they fail to recognise the vast financial losses with regards to associated risks and potential networked vulnerabilities. This inability to accurately perceive risk could also be due to a lack of experience or deficient training. Risk methodologies and assessment methods aim to assist decision makers in identifying an organisation's vulnerabilities within the networked systems. Methods to quantify identified risks and assign them with numerical metrics that can be associated with a monetary value are used. This will support decision making processes, along with predicting the likelihood of specific attacks; nonetheless these methods have limitations and are often too complex.

Previously, network administrators or 'red teams' were given the task of ensuring the topology of the network and its security. These individuals would be responsible for mapping and understanding the links between components forming the network, including the identification and documentation of potential failures, conflicts, and vulnerabilities. Modelling and understanding the links between components, their environments, and penetration testing were further responsibilities. 'Red teams' would manually draw out these relations in an attempt to identify and visualise possible vulnerabilities which expose the security of the network. This task was time consuming and often

inaccurate even on small to medium size networks. Manual assessment of dynamic SoS today would be impractical and very likely impossible, due to the sheer number and diversity of integrated components which form these extended multi-networks and due to the complexity of systems [82] [83].

Risk management frameworks often include policies, objectives, mandates, and a continuous commitment to manage risk, they can also include overall strategic and operational policies and practices. Risk management is part of the decision making process, which allows organisations to make informed choices, allows for actions to be prioritised, and can assist in identifying alternative actions. It also takes account of uncertainty, the nature of uncertainty, and how to address it. Therefore, it is vital for risk management to be dynamic, iterative, and responsive to change, constantly adapting and updating, and thus have the capacity to sense and respond to changes within the organisation [19] [80].

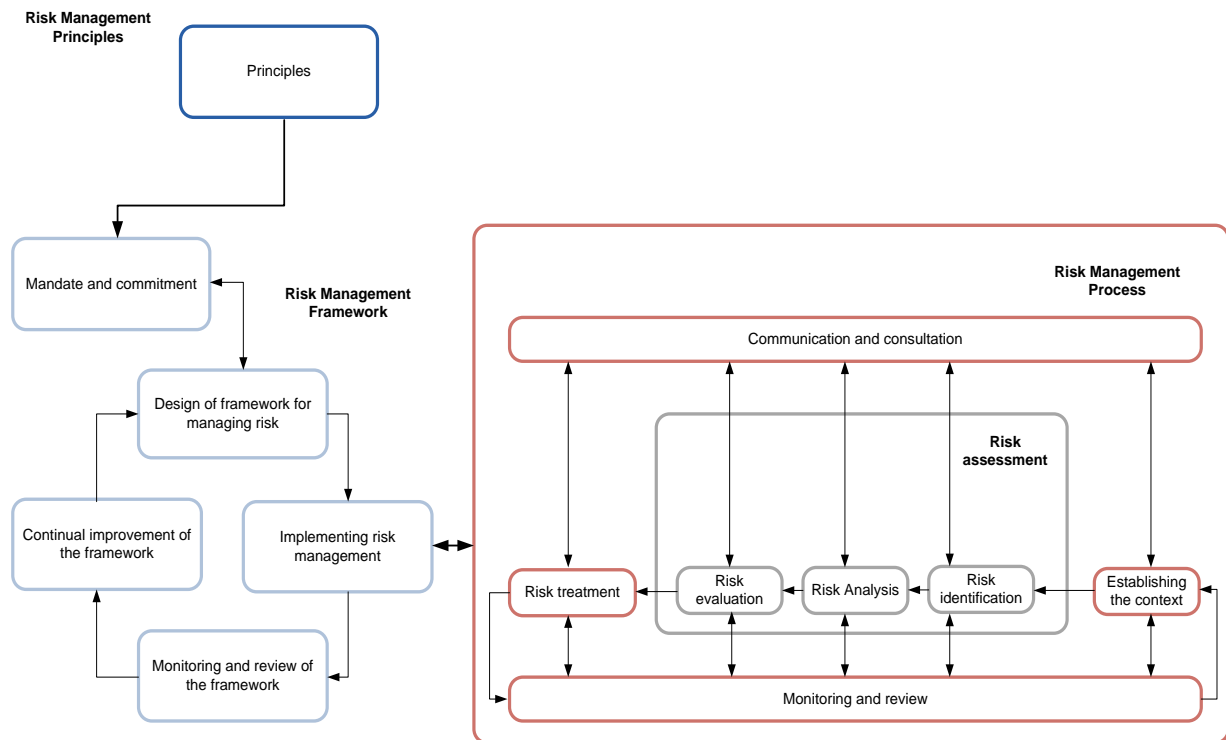


Figure 2.9. ISO 31000:2009 Risk Management Principles, Framework, and Processes

Source: Schematic representation of ISO 31000:2009 risk management policy, International Standards Organisation [80].

ISO 31000 [80] is an example that demonstrates the limitations of the existing standards and methods which aim to assist industry to identify risks within their structure’s systems. Figure 2.9 simplifies the process yet shows the relationships between the risk management principles, framework, and processes of ISO 31000. The standard identifies eleven principles to ensure risks are managed, and within the risk management review process two processes are identified that continually act [80] [81].

ISO 31000 [80] was developed to present principles and generic guidelines on risk management which were non-specific to any industry or organisation, thus could be broadly applied to varying situations and industries. The risk management framework provides a set of components which allow for continuous design, implementation, monitoring, reviewing and thus allows for continuous improvement of risk management during an organisation's lifecycle.

However, these principles would need to be greatly adapted to meet the requirements of a specific SoS in regards to their distinct systems, objectives, and varying needs (i.e. context, structure, operations, processes, functions, projects, products, services, assets and specific practices). Due to the complexity of ISO 31000, organisations failed to implement the principles and guidelines. Risk management ISO 31004 [19] was developed to review and form a new guide. In an effort to assist organisations implementing ISO 31000, the standard adopted the same broad process as AS/NZS 4360:2004 for managing risk. Inefficiently, the principles defined by this standard required heavy adaptation to meet the requirements for specific collaborative infrastructures. As a result, the risk management methods generated are non-transferable and are organisation specific, based on the infrastructure's unique composition.

Table 2.7. Risk Management and Assessment Methods

Method	Description	Risk Management Methods	Risk Assessment Methods
SP800-30 [84]	Developed in the USA, this method assists with the identification of risks and provides guidance for those responsible for the security of networked systems. It helps the decision maker recognise what to consider when implementing their Risk Management and Risk Assessment processes.	<ul style="list-style-type: none"> • Risk Assessment • Risk Treatment • Risk Acceptance 	<ul style="list-style-type: none"> • Risk Identification • Risk Analysis • Risk Evaluation
RiskSafe Assessment [85]	Developed in the UK, this method assists with the identification of risks via Cloud based software. Its risk assessment methods support the identification of threats and vulnerabilities, including assessing the level of risk, and can conduct business impact assessment.	<ul style="list-style-type: none"> • Risk Assessment • Risk Treatment • Risk Acceptance • Risk Communication 	<ul style="list-style-type: none"> • Risk Identification • Risk Analysis • Risk Evaluation
A&K analysis [86]	Developed in the Netherlands by Dutch public company RCC, and completed by the Dutch ministry of internal affairs. A handbook was produced describing a risk analysis method which allowed for threat identification and characterisation to be conducted, along with risk characterisation and exposure assessment.		<ul style="list-style-type: none"> • Risk Identification • Risk Analysis • Risk Evaluation
Mehari [87]	Developed in France, this method presents a complete risk management schema which includes asset classification, security service audits, risk identification, and situation analysis. Along with its ability to be used by decisions makers to assist with the implementation of ISO 27005.	<ul style="list-style-type: none"> • Risk Assessment • Risk Treatment • Risk Acceptance • Risk Communication 	<ul style="list-style-type: none"> • Context Establishment • Stakes Analysis • Risk Identification • Risk Analysis • Risk Evaluation

As stated, principles are bound to an organisation's specific objective, varying need and distinct topology (i.e., context, structure, operations, processes, functions, projects, products, services, assets

and practices). Table 2.7 conveys other risk management and assessment methods which are important to note. These methods suffer with similar limitations as ISO 31000 and ISO 31004, resulting in organisations struggling to implement the principles and guidelines which they define [19] [80].

To manage risks effectively we must consider shifting from past preoccupations with emphasis upon risk as the possibility of an event occurring, to the possibility of consequences of an effect upon objectives. It allows for consideration that risks are not simply events or just consequences, but instead can be descriptions of what has the potential to happen and how objectives could fail or be disrupted [81].

Current assessment methods are often too broad and require heavy adaptation, to allow for implementation. However, having such a metric specifically dealing with the complexity of SoS, and the ability to tailor standards to SoS would be beneficial. SoS can vary in size, complexity, and nature, but still have the same underlying processes, this is why these risk management and risk assessment methods are broadly developed and require heavy adaptation. Currently we feel that these broad generic methods are hindering the overall advancement of protecting SoS environments, as even though there is continuous development and remediation within the area, our research corroborates that SoS are still failing. As collaborative infrastructures become more interconnected, and as ICT trends seem to be moving towards the integration of Cloud computing, we need to resolve these complex problems before we open up SoS to an even bigger vulnerability, threat or critical risk.

We must also consider that it is no longer viable to just ensure that methods are backward compatible to allow for the integration of legacy systems with today's technology. It is essential methods are forward compatible and will continue to have the capability to identify risk far in to the future ensuring SoS are monitored for risk(s) for their entire lifecycle. Moreover, if methods are too rigid they will fail to keep up with the speed of technological advancement, which in our opinion is what appears to be happening with current risk and security solutions.

2.5 Systems-of-Systems Vulnerability Analysis

SoS security is in part determined by the vulnerabilities that expose the infrastructure to risk(s). It is vital that vulnerabilities can be identified and mitigated in order to improve the robustness of the SoS and security in order to defend against attacks or limit the impacts of failures. It is also essential to understand the vulnerabilities identified, the consequences of their failure or exploitation, and how multiple vulnerabilities can be combined by malicious attackers to increase their attacks or strengthen their footholds.

2.5.1 Network Vulnerabilities

In relation to cyber networks, a vulnerability is a weakness or fault that exposes and reduces the networked system's security. SoS are composed of a combination of physical, cyber, and human elements, and each element can be susceptible to differing vulnerabilities. Table 2.8 provides a broad summary of vulnerabilities that can reduce security within SoS and expose them to attack vectors, including categorised elements that are susceptible within an organisation's networked infrastructure, the types of attack that can be instigated, the effects of such attacks, and the requirements an attacker would utilise in order to exploit a vulnerability.

Table 2.8. Summary of Network Vulnerabilities and Attack Factors

Element	Vulnerabilities	Attack Types	Attack Effects	Attacker Requirements
<ul style="list-style-type: none"> • Hardware. • Software. • Network. • Communications. • Human Resources. • Organisation Facility. • Organisation Resources. • Data Mechanisms. • Disaster Recovery. 	<ul style="list-style-type: none"> • Insecure ports. • Misconfigured systems and software. • Programming errors. • Unpatched vulnerabilities and software. • Inadequate password and account policy. • Inadequate use of cryptography. • Inadequate access control. • Application backdoors. • Human error and intentional acts. • Inadequate physical security. • Inadequate firewall rules. • Unnecessary enabled services and applications. • Unauthorised software. • Inadequate staff training. • Network congestion. • Cache holding user ids. • Directories not secured. • No assets to protect against power loss. • Inadequate data back-up and storage. • Inadequate disaster recovery plan. • Inadequate security policies. 	<ul style="list-style-type: none"> • Brute-force. • Sniffer. • Application-level. • Misconfiguration. • Operating system. • Eavesdropping. • Reverse engineering. • Malware, viruses, and worms. • SQL injection. • Control hijacking. • Injecting false data. • Compromised-key. • Data modification. • Buffer overflow. • Cross-site Scripting. • Identity spoofing. • DDoS. • Man-in-the-Middle-Attack. • Application-Layer. 	<ul style="list-style-type: none"> • Loss of life. • Denial-of-service. • Catastrophic system failure. • Quality of service. • Loss of intellectual property. • Data leakage. • Reputation loss. • Financial loss. • Economic impact locally, nationally, and globally. • System damage. 	<ul style="list-style-type: none"> • Internet Remote Attack, attacker with Internet access can discover and send messages to device via the network with no access privileges. • Local or Remote Access, the attacker connects via local or Internet access to the device, and must have some type of privileged access. • Physical Proximity attack, the attacker is in close proximity but does not need physical access; generally these attacks are conducted against wireless nodes. • Direct Physical Attacks, the attackers have access to the physical device, and privileges might not be required to access data or system.

Source: Knapp and Langill [12], Awodele et al. [76], Stamp et al. [77], Wu et al. [79], Kumar et al. [88], Papp et al. [89], Myerson [90].

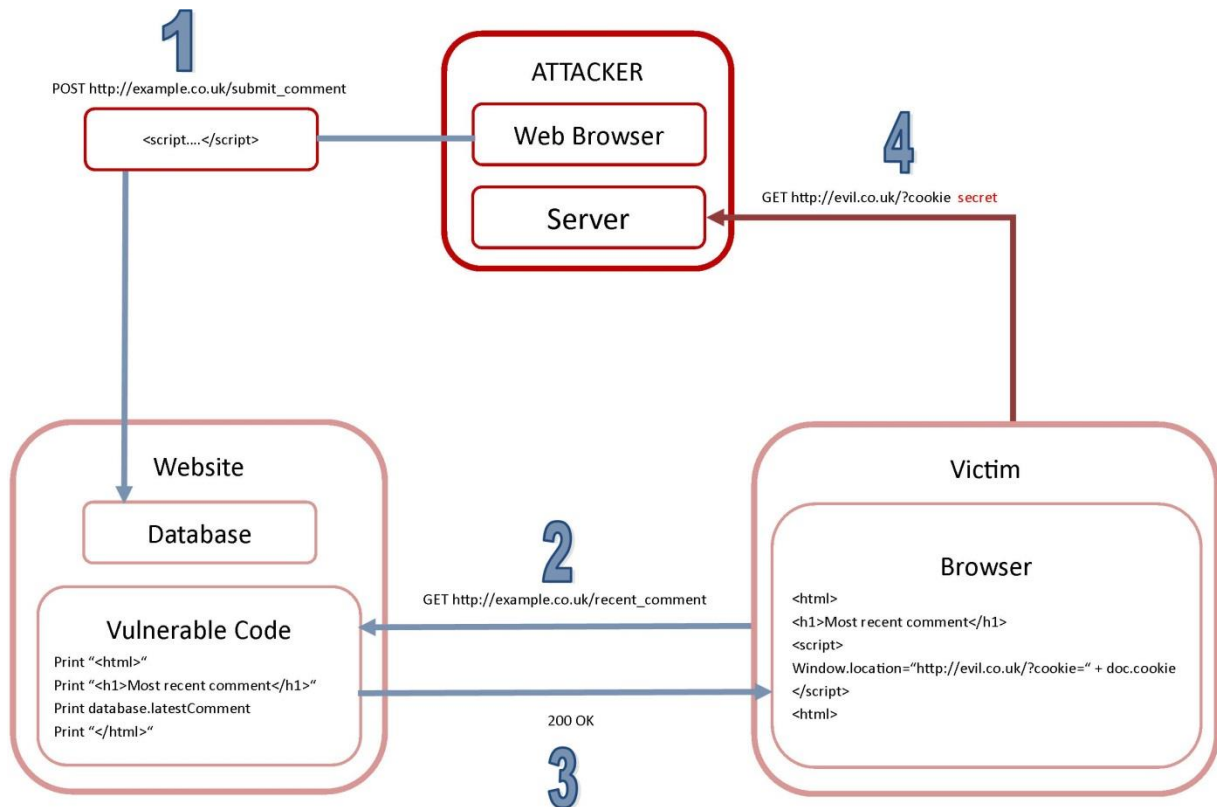


Figure 2.10. Cross-Site Scripting Attack

Source: Schematic representation of a simple XSS attack, acunetix.com [91].

One example is Cross-site Scripting (XSS); this type of attack is the submission of a malicious script into a web application or website by a malicious attacker. This critical attack uses known vulnerabilities in web applications, web servers, or web plug-in systems, and is considered one of the most utilised attacks against website security. This attack impacts both the security of the website/application and the victim's device [91].

Attacks are not directly made against the victim, instead a malicious script is injected into a web page or application that will be accessed by the individual, i.e. the website/application is a 'Trojan Horse' which delivers the malicious script within the victim's browser. This attack is only achieved if the webpage allows for direct user input to its pages, allowing for an attacker to insert code that will be incorporated into the web page and executed by the victim's web browser. Once the victim's browser innocently executes the code, in general they are unable to prevent the attack or realise the attack has occurred [91].

Recently, the National Vulnerability Database had recorded over 10,600 identified Cross-Site Scripting vulnerabilities. Table 2.9 presents six of these reported vulnerabilities, providing a detailed description of the exploitable vulnerability, its access vector and impact type, and reports if the attacker would require authentication to take advantage of the vulnerability. XSS is on occasions confused with SQL injection attacks, malicious attackers use SQL injection to enter code within a

browser's search field for example, impacting the executed query and gaining access within the database to results that would normally be inaccessible. The following solutions discussed in the remainder of this section, such as network vulnerability scanners, vulnerability analysis, scoring and exploit databases, attack graphs, and network intrusion detection systems and analysers, can assist to identify and evaluate these vulnerabilities.

Table 2.9. Identified Cross-Site Scripting Vulnerabilities

CVE-ID	Description	Access Vector	Authentication	Impact Type
CVE-2017-3133	A Cross-Site Scripting vulnerability in Fortinet FortiOS versions 5.6.0 and earlier allows attackers to execute unauthorized code or commands via the Replacement Message HTML for SSL-VPN.	Network exploitable. Victim must voluntarily interact with attack mechanism.	Not required.	Allows unauthorized modification.
CVE-2017-11611	Wolf CMS 0.8.3.1 allows Cross-Site Scripting (XSS) attacks. The vulnerability exists due to insufficient sanitization of the file name in a "create-file-popup" action, and the directory name in a "create-directory-popup" action, in the HTTP POST method to the "/plugin/file_manager/" script (aka an /admin/plugin/file_manager/browse// URI).	Network exploitable. Victim must voluntarily interact with attack mechanism.	Required.	Allows unauthorized modification.
CVE-2015-3454	TelescopeJS before 0.15 leaks user bcrypt password hashes in websocket messages, which might allow remote attackers to obtain password hashes via a cross-site scripting attack.	Network exploitable.	Not required.	Allows unauthorized disclosure of information.
CVE-2017-5528	Multiple JasperReports Server components contain vulnerabilities which may allow authorized users to perform cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks. The impact of this vulnerability includes the theoretical disclosure of sensitive information. Affects TIBCO JasperReports Server (versions 6.1.1 and below, 6.2.0, 6.2.1, and 6.3.0), TIBCO JasperReports Server Community Edition (versions 6.3.0 and below), TIBCO JasperReports Server for ActiveMatrix BPM (versions 6.2.0 and below), TIBCO Jaspersoft for AWS with Multi-Tenancy (versions 6.2.0 and below), and TIBCO Jaspersoft Reporting and Analytics for AWS (versions 6.2.0 and below).	Network exploitable. Victim must voluntarily interact with attack mechanism.	Not required.	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service.
CVE-2017-2683	The NVIDIA profiler in Android before 2016-10-05 on Nexus 9 devices allows attackers to obtain sensitive information via a crafted application, aka internal bug 30162222.	Network exploitable. Victim must voluntarily interact with attack mechanism.	Not required.	Allows unauthorized modification.
CVE-2016-8356	An issue was discovered in Kabona AB WebDatorCentral (WDC) application prior to Version 3.4.0. The web server URL inputs are not sanitized correctly, which may allow cross-site scripting vulnerabilities.	Network exploitable. Victim must voluntarily interact with attack mechanism.	Not required.	Allows unauthorized modification.

Source: NVD.nist.gov [92].

2.5.2 Network Vulnerability Scanners

Vulnerability scanning is “*the process of methodically reviewing the configuration of a set of hosts by attempting to discover previously identified vulnerabilities that may be present*” [12]. A network vulnerability scanner allows vulnerabilities within both the network’s topology and its hosts to be scanned. These tools have become highly popular both with organisations who implement them as part of an automated risk strategy and malicious attackers who seek to gain unauthorised access to infrastructures. As a result, vulnerability scanners provide specifics on weaknesses such as open ports, network configurations, system components, operating systems (OS), software applications and services, logons, and active IP addresses, etc. Network vulnerability scanners can also assist in prioritising the implementation of solutions, and commonly have the capacity to detect malicious services such as Trojans.

In general a network based vulnerability scanner would be located on a single device, and be responsible for network discovery and analyses of target hosts, collating results, comparative analysis of results against vulnerability database, and the presentation of results. Figure 2.11 presents a high level overview of a network vulnerability scanner.

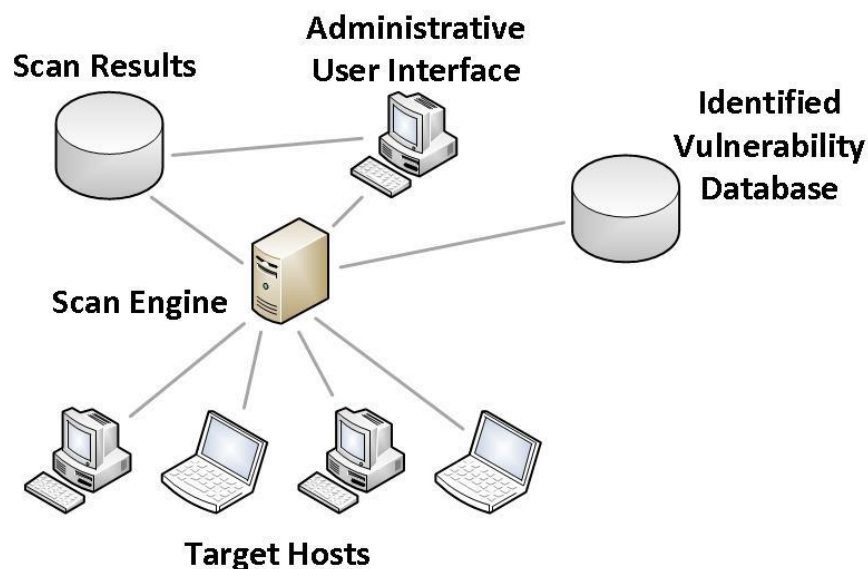


Figure 2.11. Overview of Required Network Based Vulnerability Scanner Components

These tools are efficient when monitoring for known vulnerabilities and signatures, however struggle with the identification of new vulnerabilities, with false-positive warnings requiring administrative intervention. When scanners fail to perceive the associated risks to systems and components and if a system is reported with no known vulnerabilities, it does not mean that the system is secure, unidentified and unknown vulnerabilities leave systems within the collaborative infrastructure unprotected and exposed, and susceptible to zero-day attacks.

Scan results do not determine relationships between any identified vulnerabilities, or ascertain how they can be combined by attackers attempting to penetrate the network. Systems could also have been misconfigured, software might be improperly installed, and networks might have been hardened incorrectly. Network vulnerability scanners fail to provide a complete view of the network and the associated risks, and are reliant upon external data and regular updates to maintain their own local vulnerability database [12].

Unlike firewalls, anti-virus, and IDS, network vulnerability scanners provide a proactive approach to security within ICT rather than focusing upon defending against attacks. These schemas endeavour to deliver automated platforms that identify vulnerabilities and analyse network states. Developed network vulnerability scanners include:

- Nessus [93].
- Retina [94].
- Nmap [95].
- MaxPatrol [96].
- Nexpose [97].
- OpenVAS [98].
- Saint 8 [99].

Website based network vulnerability scanners include:

- Pentest-Tools.com [100].
- Acunetix [91].
- Qualys [101].

The popular vulnerability scanner Nmap for example, has been widely adopted within industry, as it evolved as an open source platform that could be applied to the majority of common OS. When compared to other similar tools we note that there is great variance between their functionality, and no distinct database which is utilised by them all. While these vulnerability scanners prove effective within distinct networked infrastructures, there are no guarantees of their suitability when applied against SoS, as typically these infrastructures only share collaborative relations and do not always divulge or allow access to their entire network of systems. This means that vulnerability scanners will not have access or generate a complete overview of the networked systems with which they share a collaborative relationship. Therefore it is ineffective to apply these tools if we cannot evaluate every component and link which is connected via a collaborative relationship.

Vulnerability scanners only protect against known vulnerabilities that have been identified and logged within a database, which it evaluates the network against. Both open source and commercial vulnerability scanning tools can negatively impact networked infrastructures when applied, therefore extra care must be taken when applying these methods to SoS as negative impacts could be further exacerbated and potentially cause cascading failures. Therefore, to deploy a vulnerability scanner within an SoS environment considerable testing would be required, however, as stated with many critical SoS it is impossible to simply shut them down and run vigorous testing against them. In these instances simulated environments are the most effective means to ensure issues do not arise.

Characteristically vulnerability scanners are automated, and scanners that inject data into a network topology are considered to be highly dangerous and only suitable for networked infrastructures within SoS that are non-operational and offline. The use of passive tools is more dependable and less dangerous for collaborative infrastructures. There have been incidents that have been directly attributed to automated tools which have shutdown networked systems. These disruptions can result in huge financial losses and can impact the reputation of organisations. Similarly, these failings can also affect the credibility of the security firms who develop the systems and who have implemented them.

Additionally, it must be stated that not all vulnerabilities are exploitable. It is also essential that any detected vulnerability is not only identified precisely but managed effectively. Quantitative methods assist in quantifying if the risk could be exploited immediately or in the future. Solutions or corrective measures that can be applied to secure the vulnerability need to be weighed against the potential benefits of modification and the costs that would occur.

Ranking and prioritising vulnerabilities and solutions can assist in identifying the risks that pose the biggest threats that need immediate attention and those that have no immediate impact and pose no threat and risk. The cost of inflation will also play a part when prioritising remedies, as while a risk might pose only a small threat, the cost to secure it in the future might increase and be more costly later. Ranking vulnerabilities will shift the analysis of scans away from how vulnerabilities will impact a component, system, or network to the overall impact to the SoS [12], i.e. will rank the vulnerabilities with a focus of the consequences of the risk posed to the entire SoS which can have local, regional, or even global impact.

2.5.3 Vulnerability Analysis, Scoring, and Exploit Databases

A number of organisations have been established in order to assess the severity of cyber based security vulnerabilities and to develop industry standards, to assist with the assignment of numerical

values to represent the severity of the vulnerability and exploitability. Vulnerability scoring and exploit databases developed over the last couple of decades include:

- Common Vulnerability Scoring System (CVSS) [102].
- National Vulnerability Database (NVD) [92].
- Common Vulnerabilities and Exposures (CVE) [103].
- Bugtraq security database [104].
- SecurityFocus Forum [104].
- Open Source Vulnerability Database (OSVDB) [105].

Each schema identifies and measures vulnerabilities in a variety of ways, with differing focuses. Some of the schemas provide threat warning systems, whereas others provide vulnerability databases. Several vulnerability scoring methods also have the capability to assist with vulnerability identification. Vulnerabilities can be tracked and cross referenced between databases. But due to the size of the repositories it can be time consuming, therefore automated processes tend to be applied to simplify and speed up the process [12].

While there are other methods, research ascertains that CVSS and NDV have been increasingly integrated into several research methodologies, which aim to resolve issues associated with assigning risk values to collaborative network vulnerabilities. We have also utilised these two vulnerability scoring and exploit databases within our implemented solution, outlining the industry established metrics for generating the quantitative risk values in Section 4.6.1, and discuss how the methods assist in the assignment of risk, and show the complexity of automating the risk assessment process.

2.6 Network Security Systems

In order to protect against the risks and challenges discussed in this paper, and identify vulnerabilities that expose systems and increase security, there has been considerable research into attack graph generation methods and intrusion detection systems, allowing for administrators to analyse and evaluate the security of their networked cyber systems, and identify security needs.

2.6.1 Attack Graph Generation

While initially perceived in 1998, it was not until 2001 that attack graphs were developed further encompassing an automated process. Attack graph generation has gained prevalence over recent years and has long been associated with network security. This method is increasingly used to determine

and visualise how multiple vulnerabilities can be combined to penetrate a network by a malicious attacker. Traditionally, attack graphs relied upon the manual entry of data (often drawn by hand) that had been deposited within databases containing known exploits and vulnerabilities. In contrast, the size and complexity of infrastructures today means this is no longer a viable option and is a highly unrealistic approach as part of risk assessment practice [82] [83].

Attack graphs generally represent an attack at an initial starting node attempting to reach a particular goal state and define possible sequences and routes (attack paths) an attacker could exploit to reach the desired state. Generally, nodes and edges represent vulnerabilities that can be exploited and alterations caused by the attacker penetrating the network. These graphs identify and visualise potential threats that if exploited and combined form a possible attack path to goal state(s). It is possible to predict attacks by providing these details of known vulnerabilities throughout the topology. Network hardening techniques can be applied to mitigate impact and increase security within the infrastructure.

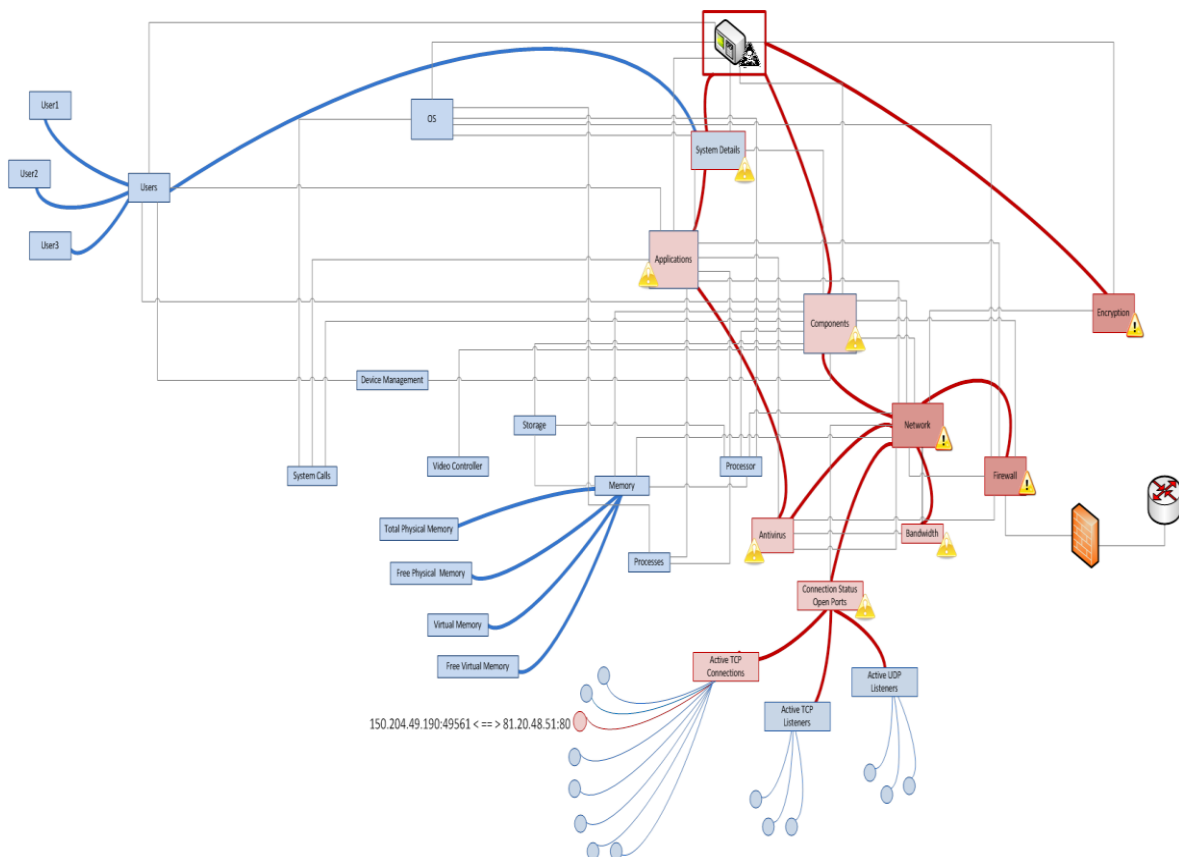


Figure 2.12. An Overview of the Identified Risks Visualised in Graph Form

The role of these graphs is to identify and detail all known vulnerabilities within an infrastructure along with detailing connectivity. Because of this, attack graphs quickly become highly complex and large in size often resulting in state explosion. Even when executed on small networks formed between small numbers of connected devices, graphs which display every possible scenario are

incomprehensible and impractical. This is due to the sheer volume of identified paths. It is unlikely that an administrator or decision maker could make sense of these graphs intuitively, or determine which route is the most prominent route of attack. Figure 2.12 demonstrates these limitations and depicts vulnerabilities which impact the security of a single networked device.

Likewise, dependency attack graphs model dependencies, relationships, and transition states with regards to network configurations. Additionally, they model the vulnerabilities within the network and the potential exploits which are then represented as attack paths. Rather than assuming attackers will comprise all states in an attack graph, dependency graphs concede attackers will not necessarily exploit all vulnerabilities within each component. Graphs visualise attack scenarios and paths between source state and target, calculating a range of attack graphs and paths that expose the network to risk [106].

Dependency graphs focus on identifying multiple paths which are likely to be exploited, rather than focusing on the most or least likely attack path, or the path that poses the highest risk. Potentially, this solution could conceal exploitable paths and leave the network exposed. Often the vertex represents the condition state of system settings, while the edge represents casual relations between each condition. Multiple paths are available for further analysis, with each state assigned a numerical value, representing the expected loss or likelihood of the state being achieved [106].

Advances within ICT allowed automated processes and algorithms to be applied to attack graph methodologies, allowing for precise graphs to be generated on the condition that accurate source data is provided. These schemas are being heavily applied to a variety of security areas including Security Risk Assessment, Intrusion Detection and Prediction, and Digital Forensics.

Attack graphs are indispensable for administrators as they provide a platform that can assist them in identifying potential exploitable vulnerabilities within their network. They also offer insight into what security measures should be deployed, assist with prioritisation of planning and implementation for network hardening. In addition, attack graphs can facilitate the understanding of the network, network topology, and the impact of potential actions.

Several schemas have been developed over the last couple of decades that assist with attack graph generation. Each schema has a variety of tools integrated within them, as well as the functionality to model different interactions, exploits, and varying simulation types. The schemas process a variety of data sources and are based on a variety of platforms. Several popular schemas are summarised below.

2.6.1.1 Multi-Host Multi-Stage Vulnerability Analysis

Multi-host Multi-stage Vulnerability Analysis (MulVAL) is an automated schema that uses Datalog for element analysis. MulVAL models interactions of software bugs along with network data, system

configuration data, and other relevant data as required. The schema leverages existing vulnerability databases and scanning tools such as OVAL (Open Vulnerability Assessment Language) and ICAT (Internet Catalog). Each host within the network is scanned asynchronously; outputs are then encoded as Datalog facts prior to being fed into the MulVAL reasoning engine. The reasoning engine is composed of a series of Datalog rules, capturing operating system behaviour and interactions between components [107].

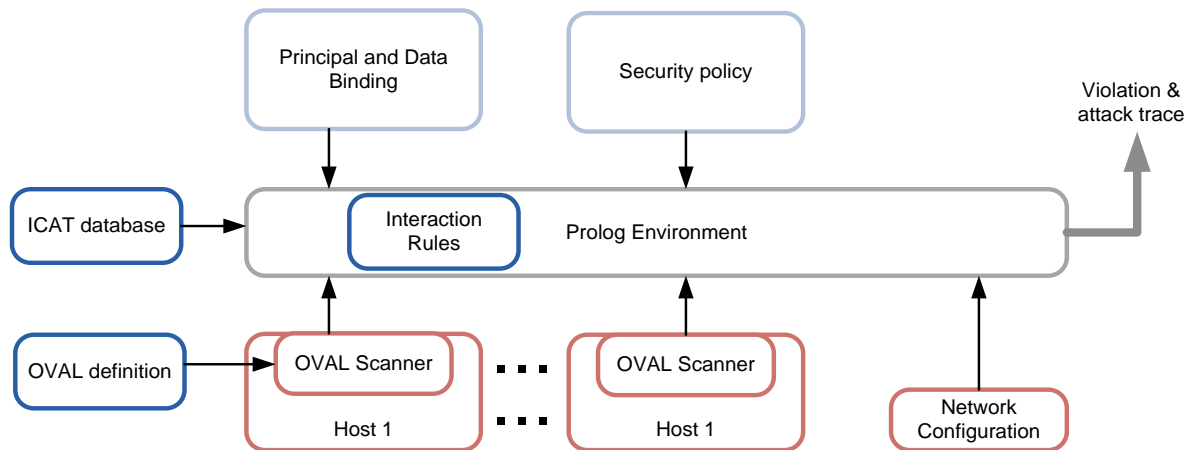


Figure 2.13. MulVAL Framework

Source: Schematic representation of the MulVAL architecture, Ou et al. [107].

MulVAL analysis inputs include advisories, host configuration, principles, interaction, and policy, this framework is outlined in Figure 2.13. Although, the MulVAL framework aims to identify potential vulnerabilities prior to an attack being launched against the network, while complementing IDS. This schema struggles, scaling as $O(N^2)$ – $O(N^3)$, as the number of hosts increases within the monitored network [107].

2.6.1.2 Network Security Planning Architecture

Network Security Planning Architecture (NetSPA) has the functionality to build network based attack graphs, using a graph structure called multiple-perquisite graph. Leveraging OVAL, NetSPA uses firewall rules, network vulnerability scans (Nessus), and vulnerability databases (NVD), to model potential attack paths of known vulnerabilities and computes network reachability. The schema can model server-side, client-side, credential-based, and trust based attacks. This is achieved by NetSPA capturing data from the input sources and forming the data into a network model which is converted into a binary file. NetSPA's computation engine reads the binary file and computes reachability; this then forms the generation of an attack graph which is analysed. Recommendations and the computation of security metrics can then be generated from these results [108] [109].

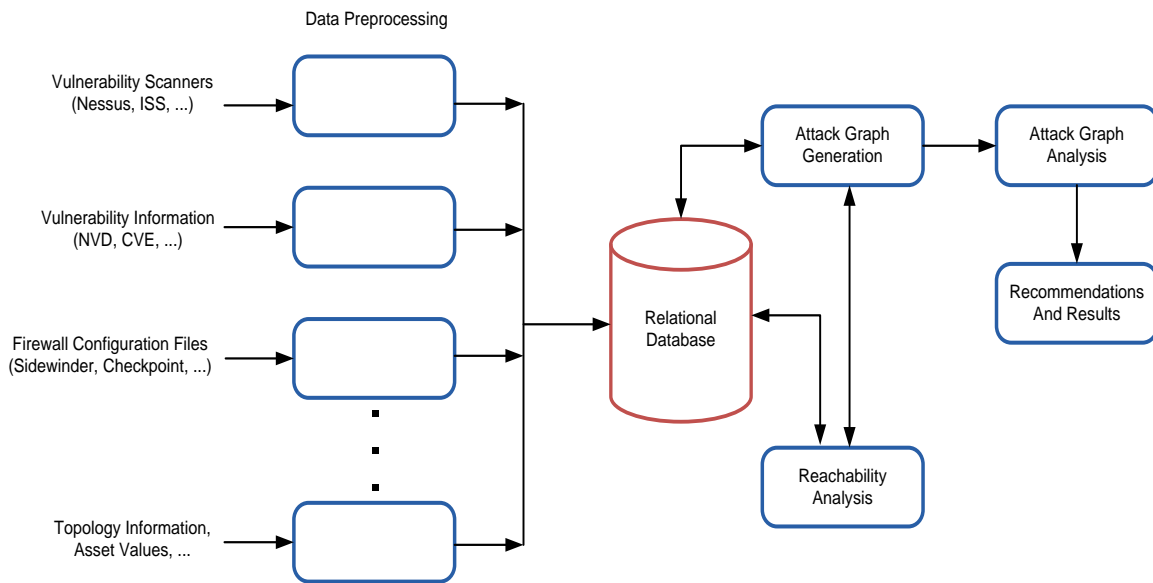


Figure 2.14. NetSPA Framework.

Source: Schematic representation of the NetSPA architecture, Ingles et al. [108], Chu et al. [109].

NetSPA has reachability systems that can emulate network firewalls and compute reachability between hosts. Consequently, NetSPA can pose as an attacker and determine vulnerabilities within firewall rule sets, which previously had been overlooked. Subsequently, GARNET (Graphical Attack graph and Reachability Network Evaluation Tool) and NAVIGATOR (Network Asset Visualization: Graphs, ATtacks, Operational Recommendations) have been developed building upon the functionality of NetSPA, which is used as their backend engine (these schemas are discussed below). The framework for NetSPA is outlined in Figure 2.14, and it has also been integrated with the OVAL-based scanner. The schema struggles with complexity, scaling as $O(n \log n)$ as numbers of hosts increase within the monitored network [108] [109].

2.6.1.3 Topological Analysis of Network Attack Vulnerability

Topological Analysis of Network Attack Vulnerability (TVA) is a method that can be applied for the use of attack graph generation. The approach analyses vulnerability dependencies and identifies each potential attack path within the monitored network. These graphs then assist in computing network hardening, recommendations, intrusion detection deployment, and alarm correlation. The method can assist with identification of optimal attack response. This method has the capability to be integrated with Nessus, Retina, and FoundScan, and can process data from a wide number of differing vulnerability databases, including NVD, Bugtraq, OSVDB, and CVE [110] [111].

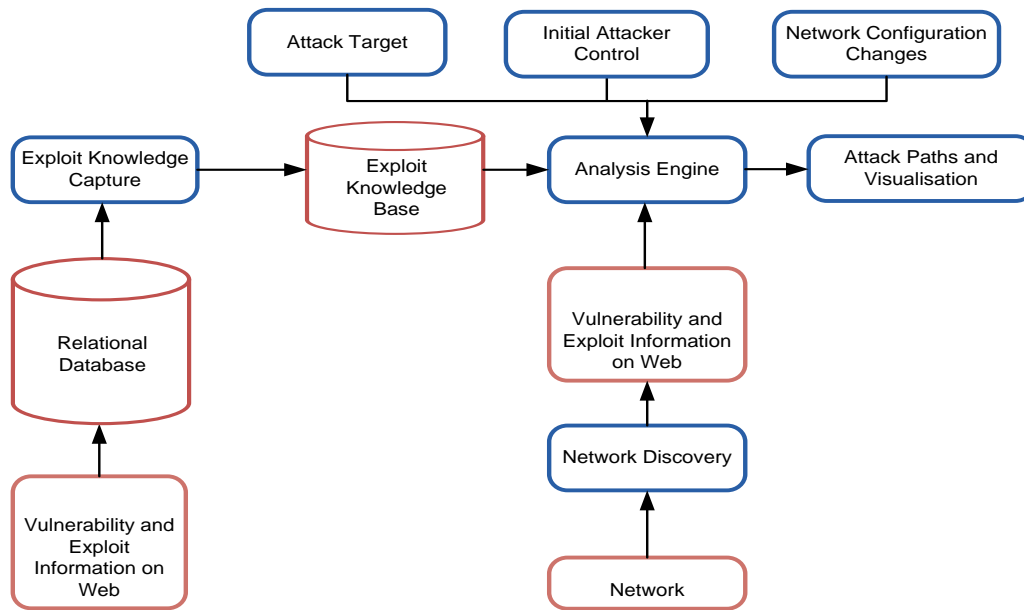


Figure 2.15. TVA Framework

Source: Schematic representation of the TVA architecture, Jajodia et al. [110], Jajodia and Noel [111].

The framework for TVA is outlined in Figure 2.15. The method states it employs efficient algorithms with worst-case quadratic complexity, and the protection domain abstraction is reduced to linear within each domain. This method scales to on $O(n^2)$ for each domain for n hosts worst-case complexity. While grouping hosts into protection domains allows the complexity to be further reduced to $O(n)$, and the complexity is $O(e)$ for e exploits [110] [111].

While considerable research has been conducted into the field of attack graph generation, existing schemas still fail to overcome attack graph generation complexity and scalability issues. The tools summarised in this section have also been heavily incorporated into the frameworks of other proposed attack graph generation methodologies, which are summarised in Section 5.

2.6.2 Network Intrusion Detection Systems and Analysers

When highly sophisticated devices are integrated with complex functionality, a higher degree of complexity and vulnerabilities is introduced. The more interconnected these systems become the more susceptible they become to network vulnerabilities and threats. The most common protection measures are Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), which can also monitor data transmission between integrated systems.

IDS have the functionality to gather and analyse data from multiple sources within a network or from a distinct device, identifying potential security breaches from both within the network and externally. IDS are generally categorised under two different types, which are Network Intrusion Detection

Systems (NIDS) or Host-Based Intrusion Detection Systems (HIDS). NIDS endeavour to identify malicious attacks or behaviour such as eavesdropping, port scanning, and DDoS attacks. This is achieved by monitoring all inbound or outbound traffic that traverses between all devices on the network. HIDS endeavour to monitor the distinct devices within a network, monitoring and analysing inbound and outbound traffic to ensure aspects such as host configuration are not modified or deleted [112].

One characteristic of IDS are their reliance upon signatures of known attacks. This requires an in-depth knowledge of the vulnerabilities associated with the protocols which are to be monitored. IDS struggle to monitor several protocols which are heavily relied upon by collaborative infrastructures, such as Distributed Network Protocol Version 3 (DNP3) and Modbus. These protocols are often relied upon in smart grid and critical infrastructure environments, where Supervisory Control and Data Acquisition (SCADA) controls are necessary for the functioning of these systems [20] [21].

Commercial IDS can be a significant investment for organisations that wish to implement them within their networked systems, and are generally an unattainable option for organisations with small networks due to cost. Open source IDS have become a suitable alternative, with vast developments being conducted in this area. Developed IDS-based platforms include:

- Snort [113].
- Bro [114].
- tcpdump [115].
- Shadow (Secondary Heuristic Analysis for Defensive Online Warfare) [116].
- M-Ice [117].
- Shoki [118].
- Spade [119].
- Firestorm [120].

2.7 Summary

This chapter provides essential background information in order to understand the principal concepts for this research. In this chapter we review the associated challenges with risk and vulnerability identification, specifically focusing on SoS environments. We identified that the main issues that leave systems insecure and vulnerable to attack vectors can be directly attributed to the SoS characteristics associated with operational independence, managerial independence, evolutionary development, emergent behaviour, and geographic behaviour. While the dynamic nature of SoS can

be rewarding, the associated challenges caused by these characteristics can result in issues arising which directly cause additional independencies, increase the complexity of networked systems, and can intensify cascade failures.

The methods that attempt to identify and quantify the risks associated with large heterogeneous systems such as network risks assessment methods, and those which model interdependencies and cascading failures such as attack graph generation methodologies, struggle with the dynamic nature, sheer size, and complexity of these SoS, and evident weaknesses have been identified. The limitations of existing frameworks and solutions that attempt to overcome the challenges discussed in this section are presented in Chapter 3.

Chapter 3

Related Work

In this chapter we focus on critically analysing existing research solutions, highlighting their benefits and weaknesses. This is to convey the inadequacies of existing approaches and will provide validation that the motivation for this research is essential. The research focuses on risk analysis as we need to develop a solution capable of identifying vulnerabilities within multi-level SoS in order to quantify the security for the entire SoS, and to increase security and mitigate risks without the introduction of additional resources. To achieve this it was vital to understand general security in regards to SoS and risk analysis, to understand what risk analysis is and how others have developed and applied it, and how SPoF theoretical and applied solutions work and how we can embed similar techniques into our solution to assist with mitigating risks. In addition, we need to comprehend how cascading failure solutions work, as we need to explore interconnected networks, as there will be an increased chance of SPoF developing and resulting in cascading effects. Complexity is another issue we need to address as we focus on multi-level SoS that introduce additional complexity into the process, and emerging behaviour techniques require examination in order for our solution to have the capacity to be dynamic and mitigate risks within deployed evolving SoS. Similarly, we also need to have a deep understand of existing solutions that model network risk, in order to determine how these techniques can be embedded within our solution to support the visualisation of risks, and assist to measure the security of critical resources and the robustness of the SoS, how data assurance techniques are utilised as we need to explore methods to secure data and mitigate risks to data as it traverses and is created and stored within an unencrypted and insecure network, and how optimisation techniques are applied in order to establish their usefulness in supporting SoS communication configuration in order to prevent additional resources being required to increase SoS security.

3.1 Systems-of-Systems Security

The aim of our research is to mitigate risks and increase security within SoS, without adding additional resources into the collaborative infrastructure. In the previous chapter (Section 2) we defined different problems that impede SoS security, and in order to facilitate our objectives it is essential that we consider and understand the identified security issues, but more importantly the methods and solutions that are currently utilised within the field of network security, moreover, determine their inadequacies and limitations when applied to multi-level SoS.

Security solutions and tools have been focusing on specific security issues in SoS or attack, and security tools are part of SoS. This section evaluates security solutions that endeavour to increase security and improve upon security techniques, and those that attempt to secure systems against specific attacks.

3.1.1 Securing SoS Against Malicious Attacks

Systems-of-Systems are impacted and exposed by comparable limitations and challenges associated with security as those of traditional cyber networks. Considerable research has been conducted into defending cyber networks against malicious attacks such as DDoS, the following research focuses on DDoS security in cyber networks, with a summary of the methods provided in Table 3.1:

Aroura and Zouari [121] propose a theoretical framework for detecting and responding to DDoS attacks, extending the Saher Architecture, and analysing the alert level in Internet Service Providers. Developing a first round of defence, that is adapted from an existing consensus algorithm, in an endeavour to alert the entire cyberspace if under attack. This then allows nodes in the environment to run a reactive mechanism, depending on the type of attack.

Singh et al. [122] provide a comprehensive insight into DDoS classifications and defence methods, which are based on deployment types (centralised and distributed). Identifying that distributed defence systems are marginally more effective than centralised DDoS defence solutions, but failing to determine the researched solution's efficiency.

Pacheco et al. [14] focus their research on ascertaining the viability of an amplified reflection DDoS attack within IoT topologies. Ascertaining how IoT environments can become targets due to their limited processing and energy resources, and exploring IoT security vulnerabilities, by applying a DDoS attack against the topology.

In the work of Naseer et al. [123], they endeavour to secure cyber networks against DDoS flood by proposing a multi-agent DDoS Mechanism, to distinguish traffic characteristics by probing traffic and classifying flash crowd traffic event and DDoS attack threat.

To protect cyber networks against jamming attacks, the following research presents solutions that attempt to defend against such attacks, with a summary of the methods provided in Table 3.1:

Houssaini et al. [124] propose a novel detection method for jamming attacks based on the application of a statistical process control, applied on the packet drop ration.

In the work of Sharah et al. [125], the authors present a reputation-based collation game to detect and mitigate insider jamming attacks within mobile ad-hoc networks. The method uses a modified

security characteristic function to enter nodes into a coalition, which makes strategic security defence decisions to exclude malicious nodes based on reputation value.

Yalu et al. [126] having conducted an in depth review of existing methodologies that endeavour to prevent jamming attacks, proposes an easy to implement protocol to increase wireless sensor network security.

Table 3.1. DDoS and Jamming Attack Detection Methods Summary

Method	Basic Concept	Pros	Cons
Aroua and Zouari [121].	DDoS attack detection and response, to enhance the security level of Saher's architecture.	<ul style="list-style-type: none"> • Different alert levels. • Coordinates Internet Service Providers to detect attacks. 	<ul style="list-style-type: none"> • Theoretical. • Inadequate distributed authentication mechanisms. • Requires national collaboration between all Internet Service providers.
Singh et al. [122].	Comprehensive insight into DDoS classification and defence methods.	<ul style="list-style-type: none"> • In depth review of existing DDoS defence methods and classification. • Comparison of DDoS defence methods against performance metrics. • Determines that distributed defence systems are more effective. 	<ul style="list-style-type: none"> • No proposed solution.
Pacheco et al. [14].	Ascertain the viability of utilising an amplified reflection DDoS attack in IoT topologies.	<ul style="list-style-type: none"> • Evaluates the efficiency of a DDoS attack in IoT. • Explores IoT security vulnerabilities. • Examines network protocol stack IEEE 802.15.4, 6LoWPAN, UDP, and CoAP. 	<ul style="list-style-type: none"> • Failed to determine compromise scale. • Does not explore other risks associated with distributed IoT.
Naseer et al. [123].	DDoS attack detection using an agent based DDoS Mechanism.	<ul style="list-style-type: none"> • Distinguish attack and valid requestors during overload. • Multi-agents activated on demand. • Subsidiary Agents automated and assist with the mechanisms scalability. 	<ul style="list-style-type: none"> • Further investigation required into false rate analysis. • Requires evaluation against different configurable parameters.
Houssaini et al. [124].	Detection method for jamming attacks based on the application of a statistical process control.	<ul style="list-style-type: none"> • Apply the identification scheme at any diffusing nodes. • Identifies identical attacks in real time. • Does not require IEEE 802.11 protocol to be modified. 	<ul style="list-style-type: none"> • Limited as it only measures against a single parameter. • Requires evaluation against other network topologies.
Sharah et al. [125].	A reputation-based collation game to detect and mitigate insider jamming attacks within mobile ad-hoc networks.	<ul style="list-style-type: none"> • Prevent insider attacks in mobile ad-hoc networks. • Monitors transmission rates and reputation of individual nodes. 	<ul style="list-style-type: none"> • Failed to determine coalition scale. • Does not protect against cooperative attacks.
Yalu et al. [126].	Jamming attack prevention within WSN utilising a proposed protocol.	<ul style="list-style-type: none"> • Security in insecure channel. • The security does not rely on PHY-layer parameters. 	<ul style="list-style-type: none"> • Theoretical. • High in computational overhead. • Does not consider other methods to counter selective jamming attacks. • Low throughput.

In addition, significant research has been conducted into eavesdropping, masquerading, and snooping attacks. The following research presents solutions that attempt to defend against such attacks or gain an insight into them, with a summary of the methods provided in Table 3.2:

Zou et al. [127] present an optimal antenna selection scheme for physical-layer security to defend against eavesdropping attacks, and examine the intercept probability performance of their proposed solution against both the single-input, single-output, and space-time-coded transmission.

In the work of Ma et al. [128], the authors focus on defending against eavesdropping attacks, and present a Moving Target Defence method that takes advantage of the Protocol-Oblivious Forwarding customisation capability. They endeavour to randomise message packets and routing paths in order to protect the communication process.

Li et al. [129] examine channel condition including path loss, the shadow fading effect, and Rayleigh fading effect, in an attempt to defend against eavesdropping attacks in Wireless Net of Things. The authors present an analytical model that investigates these attacks considering the various channel conditions, and considers attackers with resources that include omnidirectional or directional antennas.

To identify masquerading attacks in healthcare information systems, Gander et al. [130] proposed a conceptual masquerade detection framework specifically for Integrating the Healthcare Enterprise environments. They develop a two-step detection method that monitors signatures and statistical aberrations via pluggable algorithms.

Kholiday et al. [131] present a Data-Driven Semi-Global Alignment (DDSGA) framework to improve the effectiveness and performance of the semi-global alignment algorithm, which is used to detect masquerading attacks. The DDSGA endeavours to improve the scoring systems by utilising distinct alignment parameters for users, and tolerate small mutations in user command sequences to improve security.

Pratik and Madhu [132] focus on intrusion detection and propose a Cloud Intrusion Detection System for Cloud Intrusion Detection Datasets, to assist with detecting attacks and masquerades that exist within the dataset, and improve the network's security.

To better understand snooping attacks, Marques et al. [133] undertake a comprehensive survey to determine the level of success in conducting snooping attacks by malicious individuals, in an attempt to understand the current levels of security and their ability to defend against such attacks.

Gao et al. [134] propose a privacy-preserving solution to defend cache privacy from snooping attacks in Named Data Networking. Focusing on breaches of privacy that occur when malicious attackers measure the time difference between responses, and address the issue via the use of assigning credit scores to users based on their behaviours.

Table 3.2. Eavesdropping, Masquerading, and Snooping Detection Methods Summary

Method	Basic Concept	Pros	Cons
Zou et al. [127].	Examine the physical-layer security against eavesdropping attack, utilising multiple antennas at source.	<ul style="list-style-type: none"> Multiple antennas at source and destination improve both space-time-coded transmission and optimal antenna selection. 	<ul style="list-style-type: none"> Theoretical.
Ma et al. [128].	Randomise message packet and routing path, to protect against eavesdropping attacks.	<ul style="list-style-type: none"> Increases max shifting space size up to 12000bits, increasing difficulty of brute force attack. 	<ul style="list-style-type: none"> While solution increases difficulty, attackers who capture all session packets and parse the protocols of the packets could recover the message content.
Li et al. [129].	Investigate eavesdropping probability in Wireless Net of Things.	<ul style="list-style-type: none"> High accuracy for detecting eavesdropping attacks. 	<ul style="list-style-type: none"> Ignores the impact of interference. Results targeted to support future development of anti-eavesdropping solutions in Wireless Net of Things.
Gander et al. [130].	Monitor access to patient data, to protect against masquerade attacks in Integrating the Healthcare Enterprise environments.	<ul style="list-style-type: none"> Identifies malicious activity patterns. Identifies hidden activities. 	<ul style="list-style-type: none"> Not evaluated against real distributed clinical information systems data. Models can increase false-positive rates. Solution is learner-based, thus could be trained by the attacker and ignore suspicious behaviour. Solution could require extensive domain knowledge and struggle with unexpected distributions and low sample sizes.
Kholiday et al. [131].	Improve security and computation efficiency of the enhanced semi-global alignment algorithm, to protect against masquerade attacks.	<ul style="list-style-type: none"> DDSGA models user behaviour with an increased accuracy rate. Reduces false positive rates. Detection and update processes can be parallelised without impacting accuracy. 	<ul style="list-style-type: none"> Requires evaluation against network topologies to ascertain its usefulness within SoS.
Pratik and Madhu [132].	Develop an intrusion detection solution which contains a set of complete audit parameters to assist in detecting attacks and masquerades.	<ul style="list-style-type: none"> Integration of individual components. High availability. Scalable. Audit parameters can detect over a hundred instances of attack and masquerades. 	<ul style="list-style-type: none"> Centralised view of data. Not evaluated for real-time analysis of attacks.
Marques et al. [133].	Comprehensive survey to determine the level of success in conducting snooping attacks.	<ul style="list-style-type: none"> Identified predictors of the likelihood of engaging in snooping attacks. 	<ul style="list-style-type: none"> Qualitative assessment not quantitative. Not evaluated the severity of the snooping attacks. No proposed solution.
Gao et al. [134].	Defend cache privacy from snooping attacks in Named Data Networking.	<ul style="list-style-type: none"> Utilises the signatures of abnormal events to detect abnormal user behaviour. Maintains high system performance. 	<ul style="list-style-type: none"> Reliant on the signatures of known abnormal events to detect misbehaviour. Only consider privacy of users connected to same router. Requires evaluation against other network topologies. Evaluate solution against alternative credit score thresholds.

A common feature amongst all of the researched solutions that attempt to defend systems against these varying attacks is that they all focus on a single type of attack against a very specific network or data type. Meaning that their effectiveness to be applied against heterogeneous SoS cannot be ascertained, and the methods could require considerable modifications. These attacks exploit cyber vulnerabilities and risks discussed in Section 2, therefore if our proposed solution can successfully mitigate risk(s) within multi-level SoS, we can defend against these types of attack or lessen and manage the consequences of them.

3.1.2 Intrusion Detection and Prevention Methods

In an attempt to secure networks and prevent malicious attacks and unauthorised access, there has been considerable research conducted which specifically concentrates upon intrusion detection and prevention. Unfortunately many of these methods only identify existing known attacks, and while other methods have been proposed they can be costly, time consuming, and require high processing capabilities. Furthermore, anomaly based detection systems tend to identify a high number of false positives. The following research presents solutions that attempt to defend and prevent malicious intrusions, with a summary of the methods provided in Table 3.3:

One example of the limitations of current methods is the proposed solution of Chung et al. [135]. They outline a Network Intrusion detection and Countermeasure Selection (NICE) for detecting intrusions in virtual environments. This is achieved by incorporating analytical procedures used for the generation of attack graphs with intrusion detection methods. Chung et al. focus upon network intrusion detection caused by a single type of attack, failing to include other intrusion detection solutions and attacks within their virtual environment, which limits its accuracy and its potential usefulness.

Sheikhan and Bostani [136] propose a hybrid intrusion detection framework based on MapReduce to be applied within IoT topologies for distributed detection. This uses supervised and unsupervised optimum-path forest to develop a real-time framework, to detect insider attacks. The authors focus their solution to detect sinkhole and selective-forwarding attacks in IPv6 over Low-Power Wireless Personal Area Networks.

Gai et al. [137] propose a high-level solution for secure mobile Cloud computing, and adopt intrusion detection techniques to apply the mobile Cloud-based techniques into 5G environments (future networks).

In order to detect known and unknown attacks within Cloud computing, Zhao and Zhang [138] propose a distributed hybrid intrusion detection framework, which endeavours to obtain the value of

initial cluster centre and generate a cluster centre table of intrusion detection, and reduce false positive and negative rates.

Xing et al. [139] extend the work of NICE, and propose a new framework to support flexible intrusion prevention in Xen-based Cloud environments, which integrates Snort and OpenFlow.

Table 3.3. Intrusion Detection and Intrusion Prevention Methods Summary

Method	Basic Concept	Pros	Cons
Chung et al. [135].	Network Intrusion detection and Countermeasure Selection (NICE) for detecting intrusions in virtual environments.	<ul style="list-style-type: none"> Reduces the risk of external and internal attacks within Cloud systems. 	<ul style="list-style-type: none"> Ignores other attack types and intrusion detection solutions. Focuses on zombie explorative attacks in virtual environments. Fails to evaluate scalability or a decentralised approach.
Sheikhan and Bostani [136].	Intrusion detection framework to detect sinkhole and selective-forwarding attacks in IPv6 over Low-Power Wireless Personal Area Networks.	<ul style="list-style-type: none"> Detects insider and external attacks in IoT. Offers real-time anomaly detection. 	<ul style="list-style-type: none"> Requires adaptation to detect other malicious behaviours.
Gai et al. [137].	High-level solution for implementing intrusion detection techniques in 5G networks.	<ul style="list-style-type: none"> Support future 5G developers to secure mobile Cloud computing when applying intrusion detection systems. 	<ul style="list-style-type: none"> Theoretical.
Zhao and Zhang [138].	Improve bisecting K-means intrusion detection method, and develop a distributed intrusion detection method for Cloud computing.	<ul style="list-style-type: none"> Improves bisecting K-means method. Improves detection efficiency of attack data. 	<ul style="list-style-type: none"> Failed to distinguish features of high-dimensional data, impacting detection rate and false alarm rate.
Xing et al. [139].	Intrusion prevention in Xen-based Cloud environments, which expands the work of NICE and integrates Snort and OpenFlow	<ul style="list-style-type: none"> Detect intrusions and deploy countermeasures. 	<ul style="list-style-type: none"> Does not control multiple OpenFlow switches or Open vSwitches. Reliant upon a single Snort detection agent. Further analysis required into optimisation solutions to support reconfiguration of the network and to correlate alerts. Requires evaluation on a complex testing environment.
Rodas et al. [140].	Intrusion prevention, and risk detection and mitigation within Wireless Mesh Networks.	<ul style="list-style-type: none"> Robust and scalable architecture, achieved through the integration of layered redundancies. Analyses large data sets in real time. Supports self-healing. Does not unduly impact system resources. Distributed approach. 	<ul style="list-style-type: none"> Requires evaluation against other types of attack. Requires evaluation against other protocols. Requires evaluation against other network types.

In the work of Rodas et al. [140], an Intrusion Blocking for Wireless Framework is proposed, the framework aims to prevent, detect, and mitigate risks in Wireless Mesh Networks (WMN). This is achieved by means of non-relational databases for both data correlation and dissemination of intrusion information throughout the Wireless Mesh Network, to mitigate identified attacks quickly.

3.1.3 Securing SoS During the Development Life Cycle

Securing SoS is a complex undertaking, exceptional consideration, expertise, and knowledge is required to assure these types of collaborative networks. SoS can span multiple different domains, and in some instances even jurisdictions. Threats and security challenges can remain application, system, or domain specific, and have the potential to expose any collaborative network with which the system has a relationship. Hence, considerable research and development has been conducted into solutions for securing SoS, including securing during development stages and those which can be applied to existing developed or deployed infrastructures. The following research presents some example solutions that attempt to secure cyber networks during these stages, with a summary of the methods provided in Table 3.4:

Sanjab and Walid [141] critically reflect on smart grid security, and propose a cyber-physical system security framework, demonstrating how attacks propagate from cyber to physical systems utilising a game-theoretical model, presenting a bounded rationality solution, that was inspired by cognitive hierarchy theory, to model attacker thinking levels. The framework endeavours to increase network security by having a greater insight into the bounded rationality of the attacker and defender.

In order to support the development of resilient and secure SoS, Ruiz et al. [142] propose a security engineering process, utilising security artefacts that contain domain-specific security information and offer security solutions in the form of security patterns, allowing engineers to integrate security intuitively, ensuring that domain specific security knowledge can be used to meet requirements.

Mori et al. [143] implement a System Modelling Language profile to support a viewpoint-driven approach for designing SoS, having identified limitations with Architectural Description Language that was deemed essential for understanding SoS.

Trivellato et al. [144] propose a security framework that endeavours to address security challenges of future cyber based systems. This method would be employed by each collaborative party within the SoS in order to protect the local systems. Utilising a policy enforcement point that acts as an interface between parties, intercepting requests from external and local resources and contacts the appropriate policy decision point to evaluate those requests, then enforces the decision of the policy enforcement point.

To address the limitation of existing solutions Brunner et al. [145] propose a domain agnostic approach in order to model security and safety requirements across the domain, supporting certification processes in accordance with safety and security standards during the design and runtime of operations.

Having conducted an in depth review of security solutions that attempt to secure cyber networks and SoS during the security development lifecycle, we have ascertained that many of the solutions proposed are domain specific, thus cannot be applied to other SoS and infrastructures. It is essential that security methodologies are compatible and can be applied to the entire SoS, i.e. the proposed security solution can be applied to all networked systems with which they have been integrated or share collaborative relations with. Security solutions are also required to be both backward compatible to help secure aging legacy systems or existing developed SoS, and forward compatible to protect against future integration and unperceived risks and threats to assure the future security of the SoS.

Table 3.4. Security Methods Summary

Method	Basic Concept	Pros	Cons
Sanjab and Walid [141].	Cyber-physical system security framework for identifying the propagation of attacks from the cyber layer to the physical system.	<ul style="list-style-type: none"> • Models levels of attacker thinking. • Generated a security model to demonstrate how attacks can propagate from cyber to physical systems. • Computes optimal strategies to defend against attacker type and the response of the defender. 	<ul style="list-style-type: none"> • Requires evaluation against a broader range of SoS, as smart grids are being incorporated into larger extended collaborative networks. • Only considers cyber-attacks, does not consider insider attacks.
Ruiz et al. [142].	Security engineering process for creating secure SoS.	<ul style="list-style-type: none"> • Security artifacts detail security properties of specific domains, utilised during modelling stages. 	<ul style="list-style-type: none"> • Does not secure SoS or mitigate the risks of emergent behaviour. • Only strengthens and increases security for the initial development stage of the SoS. • Cannot be considered an iterative security solution.
Mori et al. [143].	Support designers to develop SoS.	Provides a conceptual model of SoS, capturing SoS concepts and interrelationships.	<ul style="list-style-type: none"> • Does not secure SoS or mitigate the risks of emergent behaviour. • Only strengthens and increases security for the initial development stage of the SoS. • Cannot be considered an iterative security solution.
Trivellato et al. [144].	Security framework to support integration of additional resources and components within SoS.	<ul style="list-style-type: none"> • Provides confidentiality of data, interoperability, and autonomy. 	<ul style="list-style-type: none"> • Requires evaluation against other network topologies. • Focuses on access control, does not consider other security risks.
Brunner et al. [145].	Inspect safety and security requirements within cyber-physical systems during their development and operation.	<ul style="list-style-type: none"> • Considers both cross-domain documentation for safety, and security requirements. • Unifies design and run-time phases. 	<ul style="list-style-type: none"> • Requires evaluation against larger network topologies. • Limited knowledge base in regards to risks.

3.1.4 Securing SoS Using Network Security Solutions

The most prominent method for securing cyber based networks and SoS includes the use of solutions such as firewalls, anti-virus solutions, antispysware techniques, patch management solutions, and virtual private networks. The following researched solutions attempt to apply and improve these existing solutions, with a summary of the methods provided in Table 3.5:

Focusing upon Cloud-based firewalls, Salah et al. [146] present an analytical model to design efficient and elastic scaling firewalls, based upon the principles of Markov chains and queueing theory.

Chomsiri et al. [147] extend their previous work and present an improved Tree-Rule firewall model, utilising IN and OUT interfaces to separate rules.

Cheminod et al. [148] focus their research on industrial firewalls for networked control systems. The authors propose a simplistic approach utilising off-the-shelf hardware and open source software, and evaluate their solution to determine safe operating margins and thresholds, and establish the impact that an industrial firewall can have when integrated within a control system in order to increase security requirements.

Cohen et al. [149] propose a novel solution for detecting infected machines in big data Security Information and Event Management systems, utilising anti-virus labels for supervised classification. The solution generates a labelled training set to assist with the identification of malware, which can be used to detect complex and dynamic patterns of machine behaviour and generate security alerts.

In the work of Dev et al. [150], the authors propose the use of a two-way caching solution, which utilises a local-cache on client system, which is used to detect the virus or other malware while in an offline state, and a Cloud-cache on the Cloud, which utilises the Artificial Intelligence Techniques to provide an optimal search rate for virus and malware definitions.

Sheta et al. [151] propose an anti-spyware solution that integrates data mining and design patterns, in order to create security patterns capable of detecting and classifying spyware.

Nunez et al. [152] present a distributed patch management solution, quantifying the security benefits of a distributed methodology compared to a centralised approach. The proposed solution assists to secure computing environments by increasing the effectiveness of patch management, and increases the organisation's security posture by defending the systems against the spread of malicious code due to the removal of underlying root causes.

Kim et al. [153] propose a patch management solution that engages with vulnerability patch sites to verify vulnerability patch integrity. The automated solution endeavours to improve security of systems and provide an efficient approach to patch collection.

Table 3.5. Network Security Methods Summary

Method	Basic Concept	Pros	Cons
Salah et al. [146].	Determine the minimum number of virtual firewalls required to meet requirements.	<ul style="list-style-type: none"> • Determines the number of required virtual firewalls to meet requirements. 	<ul style="list-style-type: none"> • Assumes measurement fluctuations are attributed to the overhead of virtualization and sharing the physical infrastructure elements and workload of other applications.
Chomsiri et al. [147].	Reduce the size of firewall rules, by separating rules into sets.	<ul style="list-style-type: none"> • Divide a big rule tree into smaller independent rule trees. 	<ul style="list-style-type: none"> • Often results in slower analysis. • Requires evaluation against other network topologies.
Cheminod et al. [148].	Determine safe operating margins when applying off-the-shelf industrial firewalls within control systems.	<ul style="list-style-type: none"> • Establishes operating margins and communication performance when an industrial firewall is integrated within a networked control system. 	<ul style="list-style-type: none"> • Fails to evaluate the solution against different traffic loads and differing characteristics. • Does not evaluate internal latency, throughput, or resilience to message flooding.
Cohen et al. [149].	Utilise anti-virus labels for detecting infected machines in big data Security Information and Event Management systems.	<ul style="list-style-type: none"> • Identifies security incidents that trigger anti-virus alerts accurately. • Identifies suspicious behaviour ignored by comparative anti-virus solution. 	<ul style="list-style-type: none"> • Requires evaluation against other network topologies. • Requires evaluation against other types of network risk. • Need to establish if the training set could be trained by the attacker to ignore suspicious behaviour. • Solution could require extensive domain knowledge and struggle with unexpected distributions and low sample sizes.
Dev et al. [150].	Improve the performance of Cloud Anti-virus Architecture.	<ul style="list-style-type: none"> • Provides quick access to malware and behaviour definitions. • Improves performance of Cloud Anti-virus Architecture. 	<ul style="list-style-type: none"> • Reliant on secure communication between the client and Cloud-cache.
Sheta et al. [151].	Detect and classify spyware.	<ul style="list-style-type: none"> • Can modify itself to detect both known and unknown spyware. • Reusable solution. 	<ul style="list-style-type: none"> • Solution is in part learner-based, thus could be trained by the attacker to ignore suspicious behaviour.
Nunez et al. [152].	Secure computing environments by increasing the effectiveness of patch management, and mitigate the spread of malicious code.	<ul style="list-style-type: none"> • Mitigates risk and eliminates vulnerabilities. 	<ul style="list-style-type: none"> • Difficult to predict how applying patches will impact integrated systems, and fails to consider the introduction of risk factors, i.e. could patches introduce emergent behaviour or cascading failures.
Kim et al. [153].	Secure computing environments by increasing the effectiveness of patch management, and mitigate risks.	<ul style="list-style-type: none"> • Mitigates risk and eliminates vulnerabilities. 	<ul style="list-style-type: none"> • Difficult to predict how applying patches will impact integrated systems, and fails to consider the introduction of risk factors, i.e. could patches introduce emergent behaviour or cascading failures.
Benzid and Kadoch [154].	Reduce handoff delay and minimise packet loss when using virtual private networks in WMN.	<ul style="list-style-type: none"> • Reduces handoff delay. • Improved network quality of service. 	<ul style="list-style-type: none"> • Requires further analysis into power consumption.
Bhat et al. [155].	Establish the need for virtual private networks to be offered as a service in Cloud computing.	<ul style="list-style-type: none"> • Could reduce organisation overheads and costs, and increase security. 	<ul style="list-style-type: none"> • An in depth review of current service architectures and missing features is required. • Would require discrete Cloud service architecture to successfully deploy virtual private networks.

Wireless Mesh Network security can be enhanced via the use of virtual private networks; to address integration issues between the distinct infrastructures Benzid and Kadoch [154] propose a Seamless Handoff Virtual Private Network solution. In an attempt to reduce handoff delay and packet loss rate, and is based on optimal path, Customer Edge based on Virtual Routing and Forwarding, and virtual private network static address.

Bhat et al. [155] provides an informative and compressive insight into virtual private networks and Cloud computing, and introduce a novel Cloud architecture Virtual Private Network as a Service. Discussing the limitations and security failings of Cloud computing, the authors establish the need for preventing data loss, data breaching and traffic hijacking via the use of virtual private networks within the Cloud.

3.2 Risk Analysis

With SoS still failing and organisations and society continually demanding more from these infrastructures and their assets and services, it is imperative that those responsible for managing SoS security identify and understand both the risks and vulnerabilities that are associated with their infrastructures, and the consequences of those risks if left to advance. When organisations fail to perceive risk or their system's tolerance to risk, then any methodology applied to their infrastructure could potentially be redundant and ineffective [12].

Risks which leave systems vulnerable and exposed are continually changing, this is because SoS are characteristically dynamic, independently managed, with systems and functions continually being phased in and out as requirements are fulfilled and new objectives identified. Previously, unknown and unimaginable risks can develop within hours, days or even years. Hence risk analysis is a continual process which is relied upon within SoS to identify and assess systems for risks, and a process that cannot be ignored, conducted once or not initiated for long periods of time. Risk analysis is merely a snapshot in time, and will only reflect the risks and vulnerabilities identified at the time the analysis is conducted. In addition, as suppliers and third parties develop new software and firmware for integration, it is vital that analysis and assessment is conducted to safeguard insecure coding techniques and software development during their lifecycles, so not to introduce new vulnerabilities exposing the SoS to additional risks [12].

3.2.1 Risk Analysis Based Techniques

In order to mitigate risks within SoS, during our extensive research it has become evident that it is not plausible to focus upon a single type of vulnerability, risk, or infrastructure. The following researched

solutions all perform risk analysis with specific agendas, a summary of the methods is provided in Table 3.6:

Yang et al. [156] investigate the methods that are exposed to JavaScript in hybrid applications, and propose a hybrid solution containing static and dynamic analysis modules. The static module detects potential vulnerable applications and collates data to act as a guide for the dynamic analysis. The dynamic analysis executes the application to verify its vulnerability status.

Peikert et al. [157] propose a procedural method to analyse an infrastructure's susceptibility to intentional electromagnetic interferences, based upon fuzzy logic and set theory. The authors extend statistical-based models fault tree analysis, electromagnetic topology, and Bayesian networks with imprecise data, uncertainty with linguistic terms, along with experts' opinions. This assists to identify elements and locations that increase risk, thus the method can contribute to increasing the protection level of the infrastructure.

The work of Liu et al. [158] focuses on cyber-attacks that relate to microgrid control systems, and consider the role of solar photovoltaic and energy storage systems control systems. The authors propose a risk assessment method for evaluating the security of the microgrid, and explore the use of the Monte-Carlo simulation to calculate monitory risk index, which assesses the topology risks when photovoltaic and energy storage systems are hacked.

In the work of Ketabdar et al. [159], the research focuses on attacker's behaviour and motivation, this is then incorporated into the risk analysis process in order to ascertain security risks more precisely. Three main parameters are incorporated into the risk process including vulnerability damage impact, breach cost, and success probability, allowing administrators to analyse the security risks factors from the perspective of the potential attackers. In addition, the administrators will consider the attackers motivation, estimate risks for identified vulnerabilities and attack paths based on the motivation, and identify paths that pose the highest risks.

Zahra and Abdelhamid [17] propose a risk analysis solution based on EBIOS methodology in order to identify the most significant weaknesses and vulnerabilities within IoT infrastructures. The authors aim to identify the most significant risk in regard to an IoT application, to ensure that developers can concentrate their efforts in order to build secure applications.

When conducting risk analysis a broad view of the entire infrastructure is required, this allows for vulnerabilities to be identified along with the risks which they pose, in an attempt to reduce SPoF and the effects of negative emergent behaviour, along with increasing the security of the networked systems. When conducting risk analysis, it is important to recognise that vulnerabilities that expose systems can be both internal within the confinements of the SoS, and environmental which are external and generally out of the SoS manager's control [12].

Table 3.6. Singular Risk Analysis Methods Summary

Method	Basic Concept	Pros	Cons
Yang et al. [156].	Study a JavaScript vulnerability that exposes hybrid applications.	<ul style="list-style-type: none"> Automatic detection method to assist with the identification of Man-in-the-Middle attacks and remote command execution. 	<ul style="list-style-type: none"> Can fail to identify some vulnerabilities and produces false alarms in some instances. Analysis module only works for HTTP traffic and fails to prevent Man-in-the-Middle attacks and threats for insecure HTTPS.
Peikert et al. [157].	Procedural method to support the analyses of infrastructures susceptibility to intentional electromagnetic interference.	<ul style="list-style-type: none"> Identifies both the elements and locations that contribute to the risk. 	<ul style="list-style-type: none"> Incorporates subjective information and assessments of risks based on the opinion of experts. Theoretical, requires analysis against differing topology types and locations.
Liu et al. [158].	Risk assessment methodology for microgrid topologies based on the physical system behaviour.	<ul style="list-style-type: none"> Considers multiple differing attack types, including syntactic attacks and semantic attacks. 	<ul style="list-style-type: none"> Evaluated against a single microgrid. Requires evaluation against a broader set of impact factors such as attacker's capability, risks in microgrid after attack, existing operation status, cyber security detection ability of operators, and other types of attack.
Ketabdar et al. [159].	Incorporate the behaviour of an attacker, to compute an attacker's motivation and the risks of existing vulnerabilities and attack paths.	<ul style="list-style-type: none"> Classifies attackers into four groups. Computes attacker motivation. 	<ul style="list-style-type: none"> Reliant on the perspective and knowledge of administrators, results could be biased and unreliable. Requires evaluation against differing topology types and on larger scale.
Zahra and Abdelhamid [17].	Determine the most important security risk within IoT applications.	<ul style="list-style-type: none"> Customizable, can place importance on specific applications and security service. 	<ul style="list-style-type: none"> Requires evaluation against differing topology types and on a larger scale. Limitations with vulnerability identification and securing potential risks.

Due to the dynamic nature of SoS, risk analysis as stated must be continuously conducted due to the frequent changes within the collaborative infrastructures. Therefore, vulnerability analysis must be repetitive and frequently scheduled or triggered by events which would emulate all system developments or changes [12]. Vulnerability identification can be conducted by either assessing the networked systems or via penetration testing and ethical hacking. The following researched solutions all conduct risk analysis on networked infrastructures or include solutions that utilise hacking and penetration techniques to identify risks, a summary of the methods is provided in Table 3.7:

Vegndla et al. [160] extend the Hacker Attack Representation Method in order to identify vulnerabilities within software, and develop a more systematic penetration testing approach.

Kadam et al. [161] convert Kali Linux and pentoo into a single automated tool for assessing Wi-Fi security. The developed tool is a distinct mobile toolkit application, which can be utilised by security experts to perform network security assessments from a mobile device.

Berger and Jones [162] focus on the benefits of ethical hacking such as open source penetration testing tools, when utilised by small and medium sized enterprises to protect network services and operations. Via the use of Nmap, Google Hacking, Nessus, and Brutus the authors identify vulnerabilities, and identify measures that can be applied to resolve these risks and prevent their case studies data from future cyber threats.

In the work of Guarda et al. [163], the authors propose a set of guidelines for conducting penetration testing within virtual environments. Guarda et al. believe that penetration testing is a vital service to systematically identify system vulnerabilities and weaknesses, analysing breaches, and mapping solutions, which allows for risks to be mitigated effectively. The framework is structured in six phases, first is to identify system vulnerabilities, and second is testing the effectiveness of security defences and resilience to attack. Third stage is the creation of malicious code exploiting specific vulnerabilities, fourth stage is a systematic system evaluation, stage five is the removal of traces and to restore system settings, followed by the final stage which produces a specialised report.

In the work of Wang et al. [15] the authors propose a new method in the area of risk analysis utilising modified Attack-Defence Trees. This approach provides the means to reconstruct attack profiles and allows for the evaluation of countermeasures in regards to security management in the Cloud. It not only considers interactive scenarios of attacks and defences, the method also takes into consideration the cost, and thus allows for analysis to consider risk regarding specific threats. The method allows for the estimation of threat probability to be considered in the event that there is an absence of adequate information. This is achieved via the use of Bayesian Network analysis.

Using complex networks theory which allows for directed and undirected graphs to be formed, and large systems with complex topologies and hidden interdependencies to be analysed, Sanchez et al. [13] propose a topologic-driven approach to model complex collaborative infrastructure interdependencies, and by proposing two new indices (Betweenness Centrality and Efficiency) the framework has the capability to identify critical edges and nodes in the infrastructure and form a topological point of view. Directing their attention to the interactions between system components, and via the development of a risk analysis technique to identify methods to reduce vulnerabilities associated with interdependency. Sanchez et al. ensure their method is not reliant upon power flow computations unlike older methods, and increase the robustness of platforms along with applying the proposed methods to other differing heterogeneous networks. While the authors endeavour to apply this solution to other properties within complex networks such as centrality indexes, path length, clustering coefficient, or to assist in modelling cyber-attacks and their consequences. The proposed

approach was not scalable and had not been applied to networks which had been formed between more than two infrastructures.

Table 3.7. Risk Analysis Methods Summary

Method	Basic Concept	Pros	Cons
Vegendla et al. [160].	Extend Hacker Attack Representation Method to attain a structured approach to bridge penetration test development and security requirements.	<ul style="list-style-type: none"> Penetration tests can be developed from models of possible attacks. 	<ul style="list-style-type: none"> Requires evaluation against differing real-world topology types and on larger scale. Does not establish if bottom-up or top-down process to attack brainstorming is the most effective, as well as if it should be conducted individually or in groups.
Kadam et al. [161].	Automate a Wi-Fi penetration testing tool to be implemented as an android application.	<ul style="list-style-type: none"> Does not require expert knowledge to conduct network and security analysis. Powerful tool that can be run from a small inconspicuous device. User interface has simple one-click commands, tester is not required to memorise long commands. 	<ul style="list-style-type: none"> Theoretical, requires analysis against differing topology types and locations.
Berger and Jones [162].	Utilise open source hacking tools to identify vulnerabilities within small and medium enterprises to improve security and prevent unauthorised access to data.	<ul style="list-style-type: none"> Identified 232 network vulnerabilities, and allowed measures to be put in place to prevent sensitive data from being compromised. 	<ul style="list-style-type: none"> Requires evaluation against differing topology types and on larger scale. Limited by the open source tools capabilities and weaknesses.
Guarda et al. [163].	Guidelines for conducting penetration testing within virtual environments	<ul style="list-style-type: none"> Penetration testing offers valuable support for improving security and mitigating risk in virtual environments. 	<ul style="list-style-type: none"> Theoretical. Security solutions need to transfer with the virtual machines. If failure occurs to security solution within the virtual environment, it could result in serious consequences.
Wang et al. [15].	Proposed an enhanced Attack-Defence Trees method for risk analysis of Cloud security.	<ul style="list-style-type: none"> Considers interactive scenarios of attacks and defences, estimates required cost, in an endeavour to analyse the risk of specific threats. 	<ul style="list-style-type: none"> Compared to other solutions this method has higher computational complexity.
Sanchez et al. [13].	Model complex collaborative infrastructure interdependencies.	<ul style="list-style-type: none"> Not reliant upon power flow computations. Increases platform robustness. Identifies critical edges and nodes, and forms a topological point of view. 	<ul style="list-style-type: none"> Limited by its inability to be applied to multiple networked infrastructures, not scalable.

What also must be considered while conducting risk analysis is decision makers' cognitive and motivational biases which can impact the effectiveness of decision making. Analysts are relied upon to provide accurate assessments in regards to risk and their skills to develop decision models, yet their biases could potentially be overconfident and misjudge the seriousness of the risk they are analysing.

Similarly, when analysts have a stake in the outcome of the analysis, their bias and self-interest could lead them to overestimate the potential consequences of the analysed risk. Should bias impact the validity of the risk analysis process, then the potential consequences of ignoring or over compensating the risks might be irreversible and difficult to correct later down the line.

While there has not been considerable published work undertaken in this area, Montibeller and Winterfeldt [18] summarise the effects bias can play when analysing and conducting decision making processes, and outline techniques which potentially could reduce the effect of bias. Even though bias can result in negative effects and result in serious consequences, some bias based on the experience and knowledge of individuals will strengthen the risk analysis process and have a positive influence on securing SoS.

3.2.2 Risk Analysis: Lab Based Risk Reduction

The US Army recognised the limitations of theoretical solutions to secure SoS, and struggled to mitigate the risks associated with their infrastructures to be deployed within the field. Instead they engaged in the use of Network Integration Events to test emerging technologies in a controlled lab based military environment. This process allows for new technology to be integrated into a physical network and operated by soldiers, in relevant military environments, ensuring that vigorous testing and analysis of the network was undertaken under operational conditions, prior to the networked systems being deployed, thus rectifying issues to prevent deployed troops from being impacted [4].

Endeavouring to strengthen the development and deployment process, the US Army developed the Laboratory Based Risk Reduction method. This solution supports network integration, design, and provides an automated analysis of systems. In addition, the method also has the capacity for issues to be detected, systems to be debugged, can detect misconfigurations, and allows upfront analysis of performance for the network [4].

The Thread-based laboratory testing means networks are built utilising the physical components that will be deployed and are integrated to form extended SoS based on the Army's deployed architecture, forming a fully-equipped brigade skeleton. Equipment could include for example, radios, routers, cross domain solutions, SATCOM, soldier handheld devices, the systems that are being tested and evaluated, etc. In addition, other applications will be incorporated on top of the framework that include call for fire, position location information using fielded mission command application systems, etc. Unified Offered Load will also ensure that systems are provided with an accurate traffic load, replicating the load for the brigade size. The Communications Effect Server-Plus emulates real-time communication effects into the networked system under analysis, can simulate complex networks, and supports integration of real networks to generate hybrid test-beds. Whereas the

Automated Performance Analysis Framework Innovation, assesses the large scale network operations in real-time, utilising compound end-to-end performance metrics, automated requirements verification, and Adaptive Learning Systems for automated determination of the Operational Effectiveness [4].

While the US army's method provides numerous benefits, and provides integration, design validation, and analysis, it is highly time consuming and expensive to implement. The process combines both live and virtualised capabilities encompassing integration and thread testing, unified offered load, emulation of communication effects, and framework analysis. The method has successfully identified numerous flaws and vulnerabilities within the systems under analysis, had the method not been used and issues occurred in the field, then systems would have been severely impacted and identifying the problems and producing solutions would be highly difficult [4].

While this solution is more effective than using modelling and simulation alone, as often these methods hide issues that later manifest once deployed and in use, and small scales of networks can be inaccurate. Network loading, performance, and dynamics tend to not scale, and issues resulting from interoperability and loading which pose significant risks to SoS, tend to manifest after deployment. The method they propose does not increase the resilience of any SoS they develop, as abnormal behaviour will often manifest long after an SoS has been operational for numerous years, and security vulnerabilities can be unknown and unimaginable at the moment an SoS is deployed. This approach truly only strengthens the development and initial deployment stages, highlighting the importance of continued research into security and risks which have the potential to expose SoS throughout the infrastructure's lifecycle. Table 3.8 provides a summary of the method.

Table 3.8. Laboratory Based Risk Reduction Summary

Method	Basic Concept	Pros	Cons
Badger et al. [4].	Laboratory based risk analysis on networked infrastructures operated by soldiers, in preparation for them to be deployed by the US Army.	<ul style="list-style-type: none"> • Successfully performs integration testing in controlled laboratory of the physical systems, to mitigate risk when systems are deployed in the field. • Can perform analysis on different scales of the system networks, from Platoon size to Brigade size. • Injects realistic traffic load into the framework so the systems can be evaluated under realistic loading levels. • Can evaluate the systems under different operational scenarios, and perform 'what-if' analysis. 	<ul style="list-style-type: none"> • Time consuming and expensive to implement. • Issues resulting from interoperability and loading tend to manifest after deployment. • Abnormal/emergent behaviour will in general manifest long after deployment, solution only mitigates risks for initial deployment. • Security vulnerabilities are often unknown and unimaginable at time of development and deployment; i.e. method cannot defend against zero-day attacks. • Does not perform iterative risk analysis for the life time of the SoS.

3.2.3 Prevention and Detection of Single Points of Failure

The complexity and size of SoS means while generally they assist in increasing the overall robustness of these infrastructures, their tightly coupled links and collaborative relations mean they often become reliant upon the transfer of critical data across essential communication links. There are evident weaknesses for example with current software based applications, security solutions, and physical hardware components and their configurations. Any single component failure/interruption or software failure/issue that prevents the transmission of critical data across the collaborative network can become an SPoF. It is essential that we identify vulnerabilities within SoS prior to their failing or exposure, to ensure that SPoF do not develop into critical failings or prevent SoS from meeting objectives.

The use of redundant systems [164], fail-safes [165], and back up equipment and software [166] is often used within SoS, in an attempt to strengthen networked systems and eliminate or lessen the impacts of SPoF. Considerable research and development has been conducted in the area of software-based fault handling and error detection [167] [168] [169] in order to protect and extend the robustness of systems. Problems associated with software based methods include scalability, and that methods generally focus upon the application of a single technique, vulnerability or network type. The following researched solutions all attempt to eliminate the impacts of SPoF, and protect and improve the robustness of the networked systems, a summary of the methods is provided in Table 3.9:

Ageneau et al. [164] examine the trade-off between application tolerated loss rate and network overhead introduced by network coding redundancy in wireless mesh networks. Ageneau et al. propose a deployable distributed redundancy algorithm for wireless meshes, in an endeavour to assure minimum decoding ration at destination, which maintains a low overhead.

Goswami et al. [165] presents a centralised framework to monitor different hardware and software statistics in multiprocessor system-on-a-chip, for fail-safes in advanced driver assistance systems. The solution identifies multiprocessor system-on-chip application behaviour and hidden overheads during run time, and monitors error counts, buffer statistics, processing latencies, bandwidth, CPU load, and low power time collection.

Savas et al. [166] focus on the issues that can arise due to disasters, and propose a Backup Reprovisioning with Partial Protection framework. This method adapts protection paths by exploiting the degraded-service tolerance of connections, in order to manage systems during large disasters.

FlipSphere developed by Fiala et al. [167], is a software based silent detection corruption solution and correction library. The method utilises hashing, erasure codes, and hardware acceleration, striving to increase application resilience for high performance computing applications. Implementing on-

demand page integrity verification combined with a software-based error correcting code, allowing for automated error recovery.

Focusing on the risk of high energy particles traversing through a digital circuit, this can cause permanent damage to semiconductor structures or cause issues with transient voltage pulse, leading to soft errors within avionics. Aydos and Fey [168] propose an error detection solution for field-programmable gate arrays, evaluating partially-based error detection against software-based retry.

Borchert et al. [169] apply an aspect-oriented programming solution to facilitate application-specific tailoring of dependable measures. The generic software-based fault-tolerance framework, reduces the runtime overhead and code size, and recovers software-based memory errors in object-oriented program data structures, that are used concurrently by multiple threads of control. The solution exploits application knowledge about memory access, analysed at compile time and hardened by compiler-generated runtime checks.

In the work of Ulbrich et al. [170], the authors focus on eliminating SPoF within safety-critical systems at the application level and propose a software based redundancy approach named CoRed. In this solution a combination of Triple Modular Redundancy, data encoding and control flow encoding techniques are used in conjunction, to assist with eliminating input and output vulnerabilities and to ensure data integrity in real-time. The solution was initially applied to an I4Copter as it was an appropriate example of a mixed-criticality multi-application real-time system. Fault detection and fault tolerance errors were introduced to test against, due to soft-errors being rare under the system's normal operating conditions.

Ulbrich et al. [170] assume that the acquired results are representative for other real world applications, nonetheless with the method not being applied to any alternative structures its appropriateness for other real world devices and systems are unproven. The outlined technique increases overhead when compared to similar techniques, which the authors consider tolerable due to its functionality in eliminating silent data corruptions and SPoF.

To prevent shut off and deletion of virtual machines and host issues within OpenStack caused by SPoF, a method utilising ceilometer and Senlin is proposed by Wang and Li [171] to achieve fast restoration and reduce complete virtual machine failure. The method relies upon specific restoration data being passed between Nova and ceilometer, which passes the time critical information across to Senlin, which relies upon a HAPolicy. However, this solution itself could be a source of SPoF as it is reliant upon the cluster being bound with HAPolicy, and time critical data transferring between multiple elements.

Table 3.9. Eliminating SPoF and Improving Network Robustness Methods Summary

Method	Basic Concept	Pros	Cons
Ageneau et al. [164].	Simplistic redundancy algorithm for wireless mesh networks to assure minimum decoding ration at destination, and maintain low overhead.	<ul style="list-style-type: none"> • Process is more efficient than static, average, and CodeMP-like schemes. 	<ul style="list-style-type: none"> • Requires evaluation against differing topology types and under different network conditions.
Goswami et al. [165].	Advanced Driver Assistance systems runtime application to monitor application statistics in multi-processor system-on-a-chip fail safe systems.	<ul style="list-style-type: none"> • Identifies multiprocessor system-on-chip application behaviour and hidden overheads during run time. • Powerful tool that can be run from small inconspicuous devices. 	<ul style="list-style-type: none"> • Requires evaluation against differing topology types and under different network conditions.
Savas et al. [166].	Dynamically adapts protection paths by exploiting degraded-service tolerance of connections, to manage large disasters.	<ul style="list-style-type: none"> • Increase system flexibility by incorporating degraded service in backup provisioning. • Can provision and restore connections with extra capacity by degrading backup paths, and better utilising network resources. 	<ul style="list-style-type: none"> • Requires evaluation against differing topology types, and under different network conditions and disaster levels.
Fiala et al. [167].	Software based silent detection corruption solution and correction library, for high performance computing applications, to increase application resilience.	<ul style="list-style-type: none"> • Provides protection for kernels. • Does not require algorithmic changes. • 90% error detection and correction. 	<ul style="list-style-type: none"> • 40% runtime overhead for the majority of applications analysed. • Fails to protect OS heap, Block Started by Symbol, and data sections. • Does not protect the stack or code in the implementation, despite being applied to process sections.
Aydos and Fey [168].	Error detection solution for field-programmable gate arrays, evaluating partially-based error detection against software-based retry	<ul style="list-style-type: none"> • Solutions error detection uses 29% to 36% less overhead than the comparable local triple modular redundancy. 	<ul style="list-style-type: none"> • Focuses on single event upset that occurs inside the flip-flop, does not consider shared nets that can cause multiple bitflips. • If single event upset occurs during a clock cycle, the error is unobservable till the next clock cycle of subsequent cycles. • Potentially application dependent.
Borchert et al. [169].	Software-based memory-error recovery solution, which exploits application knowledge about memory access, analysis of errors occurs at compile time and hardened by compiler-generated checks.	<ul style="list-style-type: none"> • Not reliant upon power flow computations. • Increases platform robustness. • Identifies critical edges and nodes, and forms a topological point of view. 	<ul style="list-style-type: none"> • Only object oriented software is addressed. • Solution can only be implemented on OS implemented in C++. • In some instances the fault resilience gains were minimal, and the increased attack surface increased the fault susceptibility.
Ulbrich et al. [170].	Eliminate SPoF within safety-critical systems at the application level.	<ul style="list-style-type: none"> • Considers input data acquisition, output data distribution, and can extend the fault detection solution to the communication. • Enables selective and application specific soft error tolerance, combining the encoding of data and redundant execution. 	<ul style="list-style-type: none"> • Assumes that results are representative for other real world applications. Requires evaluation against differing topology types and network conditions. • Increase overhead when compared to similar techniques.
Wang and Li [171].	Prevent shut off and deletion of virtual machines and host issues caused by SPoF.	<ul style="list-style-type: none"> • Recovery includes virtual machine restart, creation, and migration to other available hosts. 	<ul style="list-style-type: none"> • Reliant on HAPolicy and time critical data being transferred between elements, potential SPoF.

Rapid development within ICT and the low cost of wireless technology, means small to large sensor networks can be established using little infrastructure to collect data from a variety of environments. In WSN environments tree routing algorithms for example can cause significant SPoF. The following researched solutions propose new algorithms to overcome the limitations of existing applied methods, a summary of the methods is provided in Table 3.10:

The proposed algorithm named Relieving SPOF Tree Routing (R-SPOFTR) proposed by Lin et al. [172], reduces the average hop count and shortens end-to-end delay, while increasing throughput and the lifetime of the WSN in comparison to other approaches such as Tree Routing protocol and Shortcut Tree Routing protocol.

Other approaches include T-ROME [173], which is a cross-layer routing protocol for wireless sensor nodes utilising wake-up receivers, in an attempt to increase energy consumption and latency. Taking advantage of different transmission ranges of wake-up and main radios, the protocol skips nodes during data transfer in order to save energy. Using a set of specific parameters to optimise the relaying process, the method ensures that the most appropriate stopover nodes are chosen in case sink nodes are more than a single communication hop away.

EFMMRP [174] which is an efficient fuzzy based multi-constraint multicast routing protocol for use within wireless mobile ad-hoc networks, focuses on resolving uncertainty issues, allowing for multicast routes to be selected based on minimum fuzzy cost value increasing the network performance. Fuzzy cost is established via the conversion of quality of service, performance constraints in terms of end-to-end delay, channel bandwidth, and energy.

Pham et al. [175] propose Geographical awareness Zone Routing Protocol (GeoZRP) to mitigate routing overhead and end-to-end delay within Mobile Ad Hoc Networks (MANET). This is achieved by introducing a geographical awareness approach into the principles of the Zone Routing Protocol (ZRP), to limit the discovered route area.

The proposed Topology Sense and Graph-based protocol proposed by Rahem et al. [176], is designed to be applied within wireless ad-hoc networks and is dependent upon Triangular Matrix Table and Spanning Tree algorithm. The protocol is designed to reduce the topology information in the memory, reduce control overhead by only updating routing if topology changes and ensure that every node gets the update message, and to reduce discovery time for backup routes.

Table 3.10. Algorithms that Overcome the Limitations of Existing Methods Summary

Method	Basic Concept	Pros	Cons
Lin et al. [172].	Reduces average hop count, shortening end-to-end delay, and increases throughput, this prolongs the network lifetime and reduces SPoF.	<ul style="list-style-type: none"> • Reduces average hop count by 26%, end-to-end delay by 25%, and increases throughput by 42%, relieving congestion. • Network life is extended by over 20%. 	<ul style="list-style-type: none"> • Requires evaluation against physical topology types and under different network conditions and sizes.
T-ROME [173].	Energy efficient cross-layer routing protocol for wireless sensor nodes, increasing energy consumption and latency.	<ul style="list-style-type: none"> • When sending several packets T-Rome performs better than its comparable protocols. 	<ul style="list-style-type: none"> • Requires further analysis in regards to false discovery rates. • Does not consider opportunistic routing approaches and route adjustments based on link quality estimation.
EFMMRP [174].	Control uncertainties issues in order to conserve network resources.	<ul style="list-style-type: none"> • Reduces packet delivery delay. 	<ul style="list-style-type: none"> • Requires evaluation against physical topology types and using different parameters, different network conditions, and sizes.
Pham et al. [175].	Improve Zone Routing Protocol by introducing geographic routing, to limit the discovery area, and improve routing overhead and end-to-end delay.	<ul style="list-style-type: none"> • Improves overhead and end-to-end delay. 	<ul style="list-style-type: none"> • Only marginally decreases packet delivery ratio. • Does not consider the impact of location errors in regards to the performance of the approach.
Rahem et al. [176].	Propose an efficient routing protocol using Graph theory.	<ul style="list-style-type: none"> • Increased performance compared to conventional routing protocols. Improved throughput delay time, packet loss, and overhead. • Reduced bandwidth. 	<ul style="list-style-type: none"> • Poor scalability. Maximum number of nodes is 255, due to the limitations of using adjacency matrix.

3.2.4 Prevention and Detection of Cascading Failures

The impact that cascade failures can have upon the reliability and functionality of SoS is of great concern. Critical systems are heavily relied upon by both society and other networks that are also deemed as critical infrastructures. Should issues propagate within a single networked system which quickly cascades to other collaborative networked systems, then the SoS could fail in its entirety, resulting in both direct and indirect fallings, along with short devastating and long lasting critical consequences.

Cascading failures can be difficult to predict due to the dynamic nature, diversity of systems, and the size of the large integrated networks which form SoS, in addition generally as size increases so does system complexity. Research has been undertaken to identify cascade failures which are directly attributed to the complexity of networked systems and tightly coupled links for example, in order to mitigate risks associated with cascading failures and strengthen the security and robustness of cyber

infrastructures. The following researched solutions all endeavour to increase system reliability and mitigate potential cascade failures, a summary of the methods is provided in Table 3.11:

Cadini et al. [177] propose a modelling and simulation framework, in order to quantify the reliability and availability of power transmission grid indexes, including depicting cascade failure dynamics initiated by weather events. Combining stochastic models to define uncertain weather conditions, a cascading failure model based on DC approximation of the power flows and a proportional re-dispatch strategy, and an evaluation method utilising a customised sequential time Monte Carlo simulation scheme. The method achieves a flexible restoration model, which allows for uncertainties regarding repair process to be captured.

Probabilistic cascade failure models attempt to determine cascade failures and predict potential damage; Zhang et al. [178] consider mean field theory and apply equal load redistribution law to determine cascade failures.

A link cascade model proposed by Feng et al. [179] incorporates strong nodes via an optimisation process, which utilises an annealing algorithm to improve network robustness. While methods often improve the overall robustness of networks, in general they are inadequate and struggle to prevent cascade failures in their entirety from occurring.

It is difficult to analyse and predict how emergent behaviour within independently managed networks will develop and propagate across the collaborative systems. Also it can be challenging to identify components which are relied upon or are so vital to the collaborative relationship, that should that component fail or data transfer between the component and other systems be interrupted or prevented, then the effects will ripple across the entire SoS, directly causing mass failings to the point that the SoS will fail to meet its objectives, or potentially could result in critical consequences.

Applying an extended version of a classic betweenness method, it is possible to determine the load of an edge considering node and edge weight. Wang et al. [180] propose a cascading model which incorporates four unique metrics which quantify the robustness of the network against potential cascade failures. The model can be broadly applied to differing network types as the method's principles focus on the cascading dynamics brought on by the removal of edges with the highest weight, its application was applied to a simple network which consisted of four sub-networks.

Over recent years considerable research has been undertaken to quantify the load on nodes and edges, as this issue has been identified as a significant problem associated with cascade failure. Brummitt et al. [181] integrate a mathematical framework for multitype networks with models of sandpiles on isolated networks, in an endeavour to generate a multitype branching process approximation, capable of identifying cascades of load between simple networks and between power grids.

Table 3.11. Cascading Failure Methods Summary

Method	Basic Concept	Pros	Cons
Cadini et al. [177].	Quantify the reliability and availability of power transmission grid indexes, and represent cascade failure dynamics.	<ul style="list-style-type: none"> • Determines grid reliability and availability levels compared to regulatory constraints. • Supports decision making in regards to grid improvements and different maintenance and restoration strategies, by comparing rankings of alternative options. 	<ul style="list-style-type: none"> • Only grid line failures are considered. • Large processing times.
Zhang et al. [178].	Examine the impact of initial load and tolerance parameter distribution on cascade failure, using mean field theory.	<ul style="list-style-type: none"> • Determines Weibull distribution is superior to other distribution types, and allows for the network to sustain larger attack size and initial load and tolerance parameter. 	<ul style="list-style-type: none"> • Assumes load distribution is fixed, requires evaluation against physical topology types and using different parameters.
Feng et al. [179].	Develop a link cascade model for complex networks.	<ul style="list-style-type: none"> • Determines that a small fraction of lost links can cause the disappearance of a large number of links, thus link cascades can be stopped by integrating strong nodes within the topology that are less susceptible to link removal. 	<ul style="list-style-type: none"> • Does not evaluate weighted networks.
Wang et al. [180].	Quantify initial edge load, considering node weight and edges, and model and quantify network robustness against cascading failures.	<ul style="list-style-type: none"> • Step by step analysis of cascading propagation. • Investigates the parameter of the node weight on the network's robustness against cascading failure. 	<ul style="list-style-type: none"> • Only evaluates solution against four sub-networks, requires analysis against different topologies and network sizes.
Brummitt et al. [181].	Use multitype branching process approximation and simulations to determine how interdependence affects cascades of load.	<ul style="list-style-type: none"> • Corroborates through analysis that some independence is beneficial to networks, and every interconnection can significantly amplify cascades. • Solution can facilitate better prediction of cascading processes on modular random graphs and for multiple networks. 	<ul style="list-style-type: none"> • Does not integrate economic and physical consideration of electrical grid with costs of building connections, meaning other methods are more detailed as they combine results and provide more realistic estimates of optimal interconnectivity levels.
Cai et al. [182].	Model interactions between power systems and dispatching data networks, to increase security, reliability, and to gain a deeper understanding of complexity.	<ul style="list-style-type: none"> • Considers topological and partial transmission characteristics. • Determines the double star topology is better than mesh for power grids. 	<ul style="list-style-type: none"> • Only replicates intentional attacks on power grids, does not consider other risk factors that expose systems and can cause cascade failures.
Xue et al. [183].	Identify interrelation between network structure and operational states during cascading failure.	<ul style="list-style-type: none"> • Identifies dangerous cascading paths within the topology prior to failure, and cascading failures established via those vulnerabilities can be assessed by monitoring the loading level of the cascading paths. 	<ul style="list-style-type: none"> • To be effective solution needs to consider improved metrics and better algorithms to assist with detecting critical cascade paths more efficiently. • Does not analyse interrelation of structure and operational states.
Zhu et al. [184].	Investigate cascading failure and identify attack strategies that select target nodes.	<ul style="list-style-type: none"> • Despite analysing cascading failures from the attack perspective, results can be used to research defence strategies. 	<ul style="list-style-type: none"> • Fails to accurately define relationships between groups of nodes. • Needs to evaluate on larger networked systems, and consider more than two-node combinations.

Cascading failure is one of the most critical issues that impacts power systems. Research has been conducted into modelling the interactions between power systems and dispatching data networks based on dynamic power flow models. To achieve this Cai et al. [182] propose an approximation to detail the interdependence based on the dynamic power flow model, as they believe that model interactions is the ideal means to improve security, reliability, and understand the complexity of the entire infrastructure.

Xue et al. [183] propose a framework to examine cascading failure in an attempt to differentiate and assess relationships between structure and operational states, and define two metrics to indicate the cascading tendency and triggering force in the infrastructure, which can be used to quantitatively assess cascading risk.

Zhu et al. [184] focus upon identifying cascading failures within power grids. They propose a new metric called risk graph, a new search based node attack strategy called reduced search space node attack strategy, and a practical node attack strategy called risk graph-based node attack strategy. The methodology aims to reduce complexity when conducting extensive search node attack strategies and allows them to analyse large networks and define hidden relationships between nodes in the network that have the potential to cause cascading failure. Focusing upon identifying cascading failures which are a direct result of attacks, the technique fails to accurately define relationships between groups of nodes. The schema at time of publication had not been widely implemented on large extended networks; therefore it was difficult to determine its effectiveness as a solution.

While these models assist on providing vital information on the system's vulnerability in regards to cascading failure, and can provide assistance in identifying cascade-safe areas, further development is essential as these models require further analysis to evaluate cascade-safe operating margins.

3.2.5 Detecting Interdependence

The broad adoption and integration of ICT due to its advances and numerous benefits, introduced a large variety of cyber interdependencies and vulnerabilities as new collaborative relations were fashioned and as organisations merged their networks forming SoS. As a result, considerable research has been undertaken in order to identify and visualise the dependent links and relationships which form within large collaborative networks due to their integration, which we discuss below, with a summary of the methods provided in Table 3.12:

Sanchez et al. [13] having reviewed and compared methods which model interdependent infrastructures including Agent-based modelling, Petri Networks, Co-simulation, and Complex Networks Theory, focused their work on Complex Networks Theory. This theory allows large

systems with complex topologies and hidden interdependences to be analysed. Utilising a topologic-driven approach to model complex collaborative infrastructure interdependencies, and by proposing two new indices (Betweenness Centrality and Efficiency), the method identifies critical nodes and edges in large networked systems from a topological point of view. The proposed method aims at providing a platform for an increased understanding of the interactions between different systems components, and assists in identifying methods to reduce vulnerabilities associated with interdependency. However, the method has failed to be applied to networks which have been formed between more than two infrastructures, and no scalability evaluation has been undertaken.

Society has become heavily reliant upon numerous infrastructures which are deemed critical for the welfare and security of its citizens. Many of these critical infrastructures such as power and water distribution struggle to effectively manage and identify interdependencies that develop within their integrated systems, nor do they have the ability to map and understand critical dependent links that form due to integration. When interdependencies exist between components within a collaborative relationship then disruption or failure can result in significant consequences.

Many interdependency modelling approaches have been proposed, one such approach outlined by Heracleous [185] includes the use of open hybrid automata for generating models when systems are formed from a variety of diverse components. The method examines identified cascade faults caused due to interdependent relationships, and has the functionality to determine under what circumstances the dependent components fault will ripple and cascade across to other systems. Heracleous applies this method to the Micropolis virtual city, and focuses on modelling interdependencies between three critical infrastructures as a case study, which are power, water, and communication systems. As cities become smart and with the number of connections and new relationships forming, other external factors within the wider environment must also be considered along with the new interdependencies between other infrastructures such as transportation, banking and finance, emergency services, oil and gas, and government services. Issues could also arise due to dependent relationships for example, with smart grids being reliant upon information. Smart meters for example connected via the internet have the potential to expose power systems and create new links and platforms for malicious attack, and therefore these types of connections and devices also require serious consideration.

Complex Network Theory has also been extended in an attempt to model interdependencies. Zhu and Milanović [186] propose a three-dimensional weighted Complex Network Theory model, allowing for dependencies and interdependencies to be examined within cyber-physical systems. The framework also incorporates system characteristics, ensuring that diverse structures can be studied without modifying the topological model. By identifying critical and vulnerable components, it is possible to evaluate each physical or cyber node within its own system and other systems with which it shares links, and thus categorise weak areas. The methodology provides initial analysis of smart grids and

provides a starting point, allowing for further security, risk assessment, risk management, and defence methods to be further developed.

Tøndel et al. [187] provide a comprehensive insight into interdependencies, both the methods used for identifying and analysing interdependencies within ICT systems, and categorising them as hazard identification, causal analysis, consequence analysis, topological analysis, and dynamic analysis methods. The review focuses on cascading and escalating interdependencies, interdependency types, and their impact on power system reliability.

Table 3.12. Interdependency Methods Summary

Method	Basic Concept	Pros	Cons
Sanchez et al. [13].	Model coupled infrastructure interdependencies, and analyse complex-weighted graphs to ascertain topological indices.	<ul style="list-style-type: none"> Using Complex network theory quantifies the importance of components in coupled infrastructures. Proposed indices for undirected graphs assist to identify critical nodes and edges from a topological point of view. 	<ul style="list-style-type: none"> Only evaluates interactions between two infrastructures. Requires evaluation against larger differing SoS in order to establish its usefulness and scalability capacity.
Heracleous [185].	Model interdependencies between infrastructures, to examine the cascade effects between the systems.	<ul style="list-style-type: none"> Developed a hybrid automata model that can be utilised to study cascading failure within the Micropolis virtual city. Method can assist to determine the conditions responsible for vulnerabilities in one component cascading to others. 	<ul style="list-style-type: none"> Needs to consider wider environmental factors, along with new interdependencies between other infrastructures. As infrastructures become more integrated, new dependencies and communication links will be established, and the solution needs to consider these new platforms, e.g. IoT and smart meters.
Zhu and Milanović [186].	Utilising a three-dimensional weighted Complex Network Theory framework to model dependencies and interdependencies within cyber-physical systems, in order to identify the most vulnerable components.	<ul style="list-style-type: none"> Can evaluate different topologies without modifying the proposed model. 	<ul style="list-style-type: none"> Is not a complete risk analysis method, and requires additional assessment to manage and deploy risk mitigation measures.
Tøndel et al. [187].	Comprehensive insight into the methods for identifying and analysing interdependencies.	<ul style="list-style-type: none"> In-depth review of existing interdependency methods. 	<ul style="list-style-type: none"> No proposed solution.
Heracleous et al. [188].	Model infrastructure components based on identified dependencies, in an endeavour to generate large complex models that can be used for interdependency analysis.	<ul style="list-style-type: none"> Large generated complex models assist with interdependency analysis, including investigating cascading effects. 	<ul style="list-style-type: none"> Does not integrate geographical and logical interdependencies.

Heracleous et al. [188] propose a modelling and simulation framework based on open hybrid automata, in an endeavour to analyse interdependencies within critical infrastructure systems. Modelling accurate infrastructure components and interlinking them based on their dependencies, which creates a complex model that incorporates interdependencies.

3.2.6 Detecting Complexity

The term complexity cannot simply be defined in regards to SoS, and depends on the researcher's perspective and the circumstances of the analysis. To broadly categorise proposed solutions into two main types, research generally focuses on complexity which is associated with the physical topology of the network and the links between components, and complexity which is associated with the dynamic behaviour and operational functions of the network. The following researched solutions all endeavour to model complex networks, a summary of the methods is provided in Table 3.13:

Mane et al. [189] motivated by the PageRank algorithm and Markov analysis, propose a methodology which measures the complexity of networks during initial development. By modelling development disruption propagation as a Markov chain, they define states as the constituent systems and transition probabilities as system interdependency characteristics. Via the application of their schema, the authors have the capability to distinguish between alternate networks, demonstrating its appropriateness to manage risks during design and development of interdependent systems. While the authors provide a method to aggregate interdependent features, they fail to identify how features will be quantified and provided no guidance.

The work by Liu et al. [190] outlines a technique for generating attack graphs within complex networks. The proposed method first analyses and searches for key nodes, and examines the framework using loophole scanning. Then, via the amalgamation of forward and backward searching combined with the greedy policy generates an attack graph. This method uses Nessus to scan the network, hence, is limited by the tool's functionality and failings. Scanning tools can be inaccurate due to their inability to identify vulnerabilities associated with remote services and network connectivity. These elements are only identifiable via the examination of the host's configuration. While the method claims to reduce complexity, the proposed technique could be considered simplistic and it is uncertain of its true effectiveness. The method perceives many paths as useless and simply ignores them, yet these paths could potentially expose the network and can't simply be just ignored. It is unclear how accurate the algorithm is in regards to ignoring so called 'false paths'.

As the popularity of social networks continues to increase and these platforms grow in size, research has been conducted into processing these complex networks. Graph partitions have struggled to partition complex networks; Meyerhenke et al. [191] present an approach which aims to overcome the

limitations of graph clustering by parallelizing and adapting a label propagation technique, in an attempt to facilitate a trade-off between solution quantity and processing time. Their proposed framework effectively utilises hundreds of cores, but as the size and complexity of networks increases and as super computers currently use the processing power of millions of cores, the technique will need to incorporate other methods rather than rely upon the application of 1D partitioning.

Ranking nodes and edges in complex network is a challenge that must be overcome to ensure that data and services can be sufficiently accessed via the internet for example. Liao et al. [192] provide a comprehensive examination of existing ranking algorithms performance, and any biases that affect their effectiveness. Their work explores static and time-aware algorithms, and highlights the impact of network evolution on static algorithms and the benefits of temporal dimension for predicting network traffic, future links, and significant nodes in complex networks.

The work of Li et al. [193] analyses the state estimation problem for stochastic complex networks with switching topology, and proposes a recursive estimator developed by using the extended Kalman filter.

Table 3.13. Complexity Methods Summary

Method	Basic Concept	Pros	Cons
Mane et al. [189].	Measure the complexity of networks in the context of system development time.	<ul style="list-style-type: none"> • Can distinguish between alternate networks, managing risks during design and development. 	<ul style="list-style-type: none"> • Do not clearly state how features will be quantified and provide no guidance. • Assumes that the solution can be applied to SoS.
Liu et al. [190].	Reduce the complexity of attack graphs, by combining greedy policy, forward exploration, and backward searching, allowing the algorithm to be applied to complex networks.	<ul style="list-style-type: none"> • Complexity of attack graph is reduced, allowing for complex network security and network attack analysis to be undertaken. 	<ul style="list-style-type: none"> • Simplistic method that assumes some attack paths are useless, thus ignores them. Therefore the accuracy of the algorithm is unclear in regards to positive and negative rates of these ignored paths.
Meyerhenke et al. [191].	Develop scalable parallelisation of the size-constrained label propagation algorithm and combined into a multilevel solution to enable the partition of large complex networks.	<ul style="list-style-type: none"> • Can evaluate different topologies without modifying the proposed model. 	<ul style="list-style-type: none"> • Bottlenecks introduced by nodes with high degree, cannot be eliminated by the methods use of 1D partitioning of the adjacency matrix. • System cannot use more than 1000 cores.
Liao et al. [192].	Comprehensive insight into ranking methods in complex networks.	<ul style="list-style-type: none"> • In depth review of existing ranking algorithms, both static and time-aware, including evaluating their applications to evolving complex networks. 	<ul style="list-style-type: none"> • No proposed solution.
Li et al. [193].	Develop a recursive estimator for stochastic complex coupling networks with switching topology.	<ul style="list-style-type: none"> • Stochastic analysis methods ensure there are adequate conditions to guarantee the boundedness of the estimation errors. 	<ul style="list-style-type: none"> • Highly theoretical, evaluated by a numerical study.

3.2.7 Detecting Emergent Behaviour

The security of SoS which are operational and continually evolving is problematic, negative emergent behaviour evolution can develop at any time after integration has occurred. Emergent Behaviour can also propagate and cause cascading failures to ripple across systems and entire infrastructures. When detecting emergent behaviour within SoS current solutions struggle due to the size and complexity of these infrastructures, and most methods fail to identify abnormal emergent behaviour and cascade failures. The following researched solutions all endeavour to detect and analyse emergent behaviour, a summary of the methods is provided in Table 3.14:

The timing and consequences of emergent behaviour can be unpredictable and damaging, in an attempt to meet the associated challenges O'Toole et al. [194] present a novel distributed algorithm allowing for agents to collaboratively identify emergent events within complex adaptive systems. The decentralised emergence detection technique proposed relies on feedback that occurs as emergent behaviour appears from the component level to the system level. There are evident limitations with this technique and O'Toole et al. acknowledge that they need to extend this framework further by applying the method to a broader range of systems and devices, including differing sizes, and simulate a greater number of diverse types of emergence.

Anomaly detection is also an abstraction of emergent behaviour monitoring. Research is being undertaken to link types of anomalies with known emerging behaviour, in order to gain a better understanding of the impacts and consequences of these anomalies. By improving the categorisation of the identified emergent behaviours, anomalies, and their consequences, it will advance anomaly detection techniques and increase SoS security. Zoppi et al. [195] consider these themes and define a monitoring and anomaly detection framework for SoS. The proposed framework will need to be expanded further to ensure that as systems are phased in and out, the anomaly detection system can adapt.

Khan and Wang [196] examine emergent behaviours in multi-agent systems, which are restricted due to limitations caused by communications and environmental factors. The authors summarise formal specification as it supports system validation and can be utilised within multi-agent systems to describe and implement the manifestation of emergence and temporal logic, allowing the solution to establish properties for formal verification and capture current and future behaviour of a system. The solution presented is a principal method which will be further enhanced to capture collective motion in multi-agent systems.

The work by Shi et al. [197] considers emergent behaviour including swarming, clustering, and consensus, and surveys collaborative and non-collaborative node interactions by means of consensus dynamics. They extend existing research by exploring a relative-state-flipping model for consensus dynamics over signed random networks.

Singh et al. [198] propose a framework for analysing and simulating emergent behaviours in multi-agent systems, and classify identified emergent behaviour into different types based on Fromm's taxonomy, in an endeavour to eventually facilitate the governing of negative emergent behaviour.

Table 3.14. Emergent Behaviour Methods Summary

Method	Basic Concept	Pros	Cons
O'Toole et al. [194].	Develop a distributed algorithm for agents within complex adaptive systems to collaboratively detect emergent events.	<ul style="list-style-type: none"> Identified requirements necessary for emergence detection, and developed a DETect algorithm. This algorithm allows distributed agents to collaboratively detect emergent events. 	<ul style="list-style-type: none"> Relies on feedback that occurs as emergent behaviour appears from the component level to the system level. Assumes that the solution can be applied to SoS. Requires evaluation against differing topology types and systems, and differing types of emergence.
Zoppi et al. [195].	Developed a set of guidelines to assist when designing SoS.	<ul style="list-style-type: none"> In-depth review of issues that impact the design of monitoring and anomaly detection frameworks for SoS. Established a set of 'best practices' as guidelines to be utilised when designing SoS. 	<ul style="list-style-type: none"> Does not evaluate or understand which anomalies in general are generated by emergent behaviour, or their consequences. Does not consider system evolution, and needs to establish how anomaly detection will be maintained as systems are phased in and out of the SoS.
Khan and Wang [196].	Investigate emergent behaviours in multi-agent systems, restricted due to communication and environmental constraints.	<ul style="list-style-type: none"> Formalised the multi-agent model and provided a platform to facilitate accessibility and understanding of emergence. Reduces complexity and eased system validation. 	<ul style="list-style-type: none"> Does not fully ascertain collective motion in multi-agent systems.
Shi et al. [197].	Study asymptotic dynamical patterns that emerge among nodes interacting in a dynamically evolving signed random network.	<ul style="list-style-type: none"> Investigates a relative-state-flipping framework for consensus dynamics in signed random networks. 	<ul style="list-style-type: none"> Highly theoretical, evaluated by a numerical study.
Singh et al. [198].	Provide a structured approach for analysing and simulating emergent behaviour in multi-agent systems.	<ul style="list-style-type: none"> Classifies emergent behaviour. 	<ul style="list-style-type: none"> Needs to consider a larger number of different types of emergent behaviour. Requires further development, solution limited and does not automate mapping from finite state machine to simulation objects.

3.3 Risk Management and Assessment

Risk is unavoidable; organisations will always have to contend with risk, therefore it is vital that security managers understand their networked systems and establish a risk tolerance level. To manage risk effectively and limit its consequences, vulnerabilities must first be identified and risks which have the potential to be exploited must be managed, reduced or eliminated. Failing to identify and manage systems which pose risks to collaborative infrastructures, means it is highly difficult to accurately assess and measure the security of the SoS, and it becomes problematic to ascertain the methods that should be implemented to assure secure and robust systems.

Differing from risk analysis, risk management and assessment is about having identified vulnerabilities within a networked system, how those risks are effectively managed and assessed in order to mitigate the risks that they pose in order to increase the overall security of the infrastructure. The following researched solutions all endeavour to analyse and manage identified vulnerabilities and risks, a summary of the methods is provided in Table 3.15:

If vulnerabilities are identified, a common practice to eliminate them is via the use of patch management, summarised in Section 3.1.4. Kim et al. [153] propose a patch management solution that engages with vulnerability patch sites to verify vulnerability patch integrity, supporting increased security, patch collection capability, and reinforced patch verification.

Other methods include the use of vulnerability containment, in order to limit its effects on the integrated systems. Ahmad et al [199] propose a countermeasure framework to protect networks against fast scanning network worms. The method combined a network layer detection system and a containment system at the data-link layer, which is capable of identifying and containing worm infections with minimal false positives.

Risk containment is a technique which is often overlooked by security managers, and in some instances this solution can be the cheapest and most effective approach to initiate and manage, in comparison to other risk management solutions. Alternatively, reducing the attack surface of the SoS will assist in managing risk further, which can be achieved by removing and blocking unnecessary communication links and disabling unused ports etc., prior to their compromise.

Organisations globally have been forced to develop various risk methodology standards and practices in an attempt to manage risk. These proposed methods are typically comparable in nature, and utilise similar techniques and sequences to both identify and manage risks. Organisations that have developed risk methodologies include the British Standards Institute (BSI), Computer Emergency Readiness Team (CERT), the European Union Agency for Network and Information Security (ENISA), International Organisation for Standardisation (ISO), and the US National Institute of Standards and Technology (NIST). In Section 2.4.1, we critically reviewed several risk management and assessment methods. These methodologies require heavy adaptation for them to be applied against specific SoS, as the principles defined are too broad and non-specific. Our research determines that there is no unique documented risk methodology that can be intuitively applied to any SoS that is currently fully developed and operational without adaptation.

Conducting risk assessment on SoS is highly problematic, great consideration must be undertaken when applying assessment methods directly to systems which are deemed critical, especially if assessment methods have the capacity to impact the collaborative systems' components or affect their ability to meet objectives. In these instances theoretical assessment methodologies are suitable alternatives for conducting risk assessment, and can be the only viable option in some instances.

Ordinarily, theoretical network risk assessments are conducted via questionnaires, and rely upon the expert knowledge and experience of the infrastructure's security managers. Accurate assessment using this technique is not guaranteed or precise, as it is reliant upon the experts' knowledge, experience, and ability to be unbiased. To be successful and increase the accuracy of the method multiple experts representing multiple fields of expertise on the systems, including for example security managers, engineers, health and safety officers, operations, and maintenance, etc., are required as part of the analysis process. Offline assessment often does not reflect the security of the entire SoS, nor does it typically deliver a complete view of the collaborative infrastructure. This method commonly fails to identify communication links with third party systems and applications, sub-systems, and vital components.

Offline assessment is outside the scope of our current research, but it must be noted that there are instances where security managers are forced to use this method due to the assessment processes posing too big a risk, making offline analysis the only viable solution to provide realistic assessment results. As well as considering the direct impact a risk assessment might pose to a network, we also must consider the cost of implementing assessment methods versus any benefit.

When conducting risk assessment it is generally an automated process that utilises software applications, however, to assure network security and identify all risks, the physical components and systems, security controls, documentation, including application, host, and network configuration files will also need to be analysed [12].

The Cyber Security Evaluation Tool (CSET) [200] developed by the US Department of Homeland Security's Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), is a free software tool to assist security managers in assessing their industrial control systems and ICT networks and practices. CSET is an offline assessment method reliant on security managers selecting one or more cybersecurity standards, which generates questions based on those requirements. Once detailed questions have been completed, a series of reports are generated identifying the strengths and weaknesses of the infrastructure. Again this method relies upon the expertise of security managers and their understanding of risk and their systems, in addition to their ability to make strong unbiased assessments. Figure 3.1 visualises the assessment process of CSET.

Yao et al. [201] analyse topological vulnerabilities of advanced metering infrastructures, with a special interest in risks which could be utilised by malicious attackers to steal electricity by directly targeting smart meters. They propose a risk assessment protocol to identify nodes which are vulnerable to modification and exploitation, utilising known information regarding the network and further data which is obtained via existing monitoring solutions. Hence, the method is limited by the functionality and failings of the monitoring tools used to gather accurate network data.

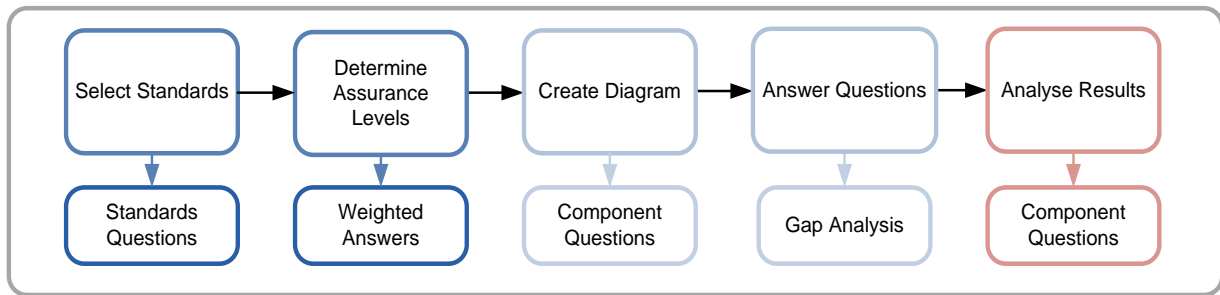


Figure 3.1. Cyber Security Evaluation Tool Assessment Process

Source: Schematic representation of Cyber Security Evaluation Tool Assessment Process, US Department of Homeland Security [200].

The paper by Tanimoto et al. [202] outlines the challenges of assessing and securing MANET, particularly when these networks are formed using personal mobile and smart phone devices. Tanimoto et al. look at the risks which directly expose the personal data on the device to ensure that the MANET is secure in regards to the personal user. This work identified and analysed nineteen risk factors, and the solutions to assist in securing the devices, and future work will examine how to quantitatively assess countermeasures.

Loutchkina et al. [203] outline a System Integration Technical Risk Assessment Model (SITRAM), based upon Bayesian belief networks combined with parametric models. The method was designed to be used during the initial development stages, as research showed limitations for models which examined systems integration technical risks. While system integration provides significant challenges, using hierarchical holographic modelling it was possible for initial factors associated with risk integration to be identified; furthermore it was possible to identify relations between risk factors and risk factor taxonomy. Using the hybrid of Bayesian belief networks and parametric models to represent relations between risk contributing factors and provide input data, a suitable modelling tool was presented along with corresponding software support tools.

The paper by Guzman et al. [204] critically examines Artificial Intelligence algorithms for risk assessment related to safety critical infrastructures. Algorithms were generally categorised into three classifications which were Expert Systems, Artificial Neural Networks, and Hybrid Intelligent Systems, and the authors conducted a comparative analysis of techniques which included Fuzzy-Expert System, Neural Networks, and Adaptive Neuro Fuzzy Inference System.

Cheng et al. [205] provide a comprehensive insight into quantitative risk assessment, and adopt a region based method to assist in quantifying the probability of a target being directly attacked. A clustering method is then applied to generate a region graph, which they consider is superior to the belief propagation method.

Table 3.15. Risk Management and Assessment Methods Summary

Method	Basic Concept	Pros	Cons
Kim et al. [153].	Secure computing environments by increasing the effectiveness of patch management, and mitigate risks.	<ul style="list-style-type: none"> • Mitigates risk and eliminates vulnerabilities. 	<ul style="list-style-type: none"> • Difficult to predict how applying patches will impact integrated systems, and fails to consider the introduction of risk factors, i.e. could patches introduce emergent behaviour or cascading failures.
Ahmad et al [199].	Countermeasure solution, which has the capability to detect and contain identified worm infections.	<ul style="list-style-type: none"> • Detects and contains identified worm infections. 	<ul style="list-style-type: none"> • Requires evaluation against differing infrastructures, background traffic, and other types of vulnerabilities. • Further investigation regarding false positives.
US Department of Homeland Security [200].	Systematic and repeatable set of principles for evaluating industrial control systems and cyber-network security practices.	<ul style="list-style-type: none"> • Offline assessment method that will not be applied directly to the infrastructure, therefore will not introduce vulnerabilities or impact system performance. • Assess systems based on step by step questions regarding the infrastructure based on industry standards, and presents detailed charts showing strengths and weaknesses, along with a prioritised list of recommendations to increase security. 	<ul style="list-style-type: none"> • Reliant on the knowledge, expertise, and understanding of risk by security management, along with their ability to make strong unbiased assessments.
Yao et al. [201].	Risk assessment protocol to identify communication networks infrastructure targets within advanced metering infrastructures.	<ul style="list-style-type: none"> • Compatible with existing system monitoring technologies. • Identifies vulnerable infrastructure targets that could be exploited to steal electricity. 	<ul style="list-style-type: none"> • Limited by the functionality and failings of the monitoring tools utilised to gather network data.
Tanimoto et al. [202].	Assess mobile ad-hoc network risks and propose countermeasures, in order to establish a mobile ad-hoc network that is secure from an individual user's standpoint.	<ul style="list-style-type: none"> • Identified nineteen risk factors and proposed countermeasures. 	<ul style="list-style-type: none"> • Does not quantitatively evaluate countermeasures.
Loutchkina et al. [203].	Provide a modelling framework and support software for system integration risk assessment.	<ul style="list-style-type: none"> • Parametric models deliver project-specific data to Bayesian belief network models. • Interfaces between parametric and Bayesian belief network models, ensures model integration into the risks management processes is simplified. 	<ul style="list-style-type: none"> • Requires model validation and sensitivity assessment.
Guzman et al. [204].	Investigate and compare artificial intelligence algorithms for risk assessment.	<ul style="list-style-type: none"> • In-depth review of existing artificial intelligence methods that improve the accuracy of risks assessment of infrastructures deemed critical. • Performs a comparative analysis of three distinct techniques, which are Fuzzy-Expert System, Neural Networks, and Adaptive Neuro Fuzzy Inference System. 	<ul style="list-style-type: none"> • No proposed solution.
Cheng et al. [205]	Implementation of a region based method to estimate the likelihood of a target being compromised.	<ul style="list-style-type: none"> • Methods performance of region based approximation is more effective than the belief propagation methods. 	<ul style="list-style-type: none"> • Fails to evaluate the algorithm properties and its run time application.

3.4 Risk Modelling

3.4.1 Network Modelling

Due to the sheer complexity of these large heterogeneous SoS and because of society's dependence upon the infrastructure, a wide majority of research is purely theoretical. This is because academics and industry practitioners do not have the required resources available to them, to test and implement their procedures, as recognised by the US army which developed a Laboratory Based Risk Reduction method [4], in an attempt to strengthen the development and deployment process summarised in Section 3.2.2. The following researched solutions all endeavour to model networks and their associated risks, a summary of the methods is provided in Table 3.16:

To defend critical systems Wang et al. [206] propose an attack graph-based probabilistic metric for measuring network security, in an attempt to prevent sophisticated malicious attacks that combine multiple vulnerabilities to reach a specific target state. The method relies upon the assumption that individual scores based on expert knowledge in regards to exploits is accurate, and despite other probabilities existing only consider a fixed probability for measuring vulnerabilities. Reliance upon the human element to assign scores means the method potentially could be highly inefficient and inaccurate, as scores will be influenced by the skill level and training of the individual assigning the score, along with their pre-conceived perception in regards to risk. The authors base scores on the level of difficulty for a vulnerability to be exploited; failing to consider the level and skill of the attacker or how much finance and resources the attacker has access to. Yet, we must admit that these elements are difficult to predict and identify, but are valid points to raise and be aware of.

Feng and Jin-Shu [207] propose a flexible approach using attack graphs to measure the security of the critical resources within the monitored network. A backward iterative algorithm is presented to solve the issues associated with cyclic attack paths within attack graphs. The proposed method does not require a complete input probabilities dataset. While the method is effective at measuring security, the greater the source input the more precise the result of measure. The accuracy of the method is not fully identified when results are generated on scarce input data, and the technique seems to have not been compared to other similar solutions to ascertain its true effectiveness.

A comprehensive insight is provided by Kecskemeti et al. [208] into modelling and simulation challenges faced when attempting to design and deploy IoT systems. Kecskemeti et al. identify that 99% of IoT data is either not collected or not analysed, and as IoT popularity and integration increases so will the size of these data sets and complex heterogeneous networks. Data sources that are not being used for their full potential and sources that are used in real-time control and anomaly detection are failing to be adequately utilised and analysed, therefore can leave systems vulnerable.

The work by Sarigiannidis et al. [209] provides an insight into multiplicative networks specifically focusing on their use in applications for modelling networked systems, in order to defend IoT systems against various security issues. The paper introduces key performance metrics, and presents a security threat model capable of estimating data losses within IoT systems, and quantifies the intensity of attacks in the application domain.

Milanović and Zhu [210] critically examine multiple Complex Network Theory based methodologies which have been developed to model and analyse interconnected networks. The authors present a three-dimensional holistic model based on Complex Network Theory, in an attempt to analyse interdependencies and interactions of the integrated systems.

Table 3.16. Network Modelling Methods Summary

Method	Basic Concept	Pros	Cons
Wang et al. [206].	Attack graph-based probabilistic metric to understand and measure the likelihood of vulnerabilities being combined to reach a goal state.	<ul style="list-style-type: none"> • Metric for measuring security to prevent sophisticated attacks combining multiple vulnerabilities. 	<ul style="list-style-type: none"> • Reliant on security managers to assign scores, therefore influenced by their expertise, understanding of risk, and their ability to make strong unbiased assessments. • Does not consider the attacker's level, skill, and financial resources. • Not capable of measuring the security risk of physical networks.
Feng and Jin-Shu [207].	Flexible attack graph-based approach to measure the security of the critical resources within the monitored network.	<ul style="list-style-type: none"> • Does not require a complete input probabilities dataset. • Backward iterative algorithm to overcome issues associated with cyclic attack paths in attack graphs. 	<ul style="list-style-type: none"> • Requires evaluation against differing techniques. • Further evaluation required to determine the accuracy of the method when results are generated using scarce input data.
Keckemeti et al. [208].	Review of modelling and simulation challenges that impede the design and development IoT.	<ul style="list-style-type: none"> • In-depth review of existing modelling and simulation challenges within the field of IoT. 	<ul style="list-style-type: none"> • No proposed solution.
Sarigiannidis et al. [209].	Analytic model for modelling IoT infrastructure under attack.	<ul style="list-style-type: none"> • Adopts G-network concept as it allows for negative arrivals to be considered when modelling security attacks. • Identifies operation characteristics of the principal IoT systems under both light and heavy attack. 	<ul style="list-style-type: none"> • Requires evaluation against other types of attack and infrastructure. • Does not utilise threat detection systems.
Milanović and Zhu [210].	Based on complex network theory, a three-dimensional model to study interdependencies and interaction of interconnected systems is developed.	<ul style="list-style-type: none"> • Establishes the integration of different topological patterns within the three-dimensional model, produces accurate modelling of interconnected systems with differing behaviours. • The framework's ability to capture different engineering structures enables criticality variation analysis of system components. 	<ul style="list-style-type: none"> • Does not consider functional level modelling. • Requires further development of the control theory.

3.4.2 Attack Graph Generation

While many of the attack graph generation methods state they have been applied in a distributed approach, they tend to only be distributed in the sense that the components are distributed within a networked infrastructure. In general the methods have never been applied to multiple distinct distributed systems that are integrated solely to fulfil an objective, yet remain standalone or perhaps even collaborate via other SoS. We know of no automated method to date that does not use a centralised approach or a method which does not rely upon specific standalone resources (e.g. vulnerability database). Should the centralised management point or those specific resources be targeted, attacked, interrupted or corrupted, then the centralised point and those resources become a SPoF for the method, which will result in CI and SoS defence becoming vulnerable or exposed to potential vectors. The following researched solutions all endeavour to analyse and develop attack graph methods that are capable of demonstrating how vulnerabilities can be combined to reach a goal state and for the steps an attacker needs to take in order penetrate the security and gain access into the infrastructure, a summary of the methods is provided in Table 3.17:

Kaynar and Sivrikaya [211] propose a distributed attack graph generation approach. They focus upon defining a new distributed search-based algorithm, which is implemented across a multi-agent platform using a virtual shared memory abstraction. The method relies upon data from the National Vulnerability Database and Common Weakness Enumeration database, and relies upon the manual generation of pre and post conditions for weaknesses. This indicates that the process is not fully automated. Once all search agents finish their partial graphs, these attack graphs are sent to a pre-assigned leader agent. This agent is responsible for merging all the graphs into a single generated attack graph, meaning the final processing is constructed within a centralised point.

The work of Li et al. [212] presents a searching forward attack graph generation algorithm based on hypergraph partitioning. The framework is applied to large scale complex networks, in an attempt to improve the efficiency and load balancing on each node, devise new attack templates, and to improve attack graph generation. In addition, they explore the use of reversing attack graph generation by generating graphs from the vulnerabilities to the attacker, in an effort to reduce required computing resources.

It is vital that methods for testing network security continue to advance, Nichols et al. [213] focus on the addition of priorities into exploit models of hybrid attack graphs. The method aims to reduce the state explosion problem, while sustaining an adequate amount of relevant data to ensure vital information is not omitted and maintain security.

Chejara et al. [214] base their methodology on conditional probability methods. The proposed schema allocates scores to identified paths within the attack graphs to assist with analysis, identifying the most critical paths. Conditional probability allows them to calculate scores for multiple devices rather

than rely upon the scores assigned by CVSS, which provides scores for individual devices and vulnerabilities, aiming to assist administrators to identify potential risks, and prioritise network hardening. The method is limited in the identification of vulnerabilities and cannot be considered a complete security solution. It also ignores CVSS temporal and environmental metrics (summarised in Section 4.6.1.2), consequently disregarding potential threats or influences by external factors which could impede or expose key systems and their security.

Table 3.17. Attack Graph Methods Summary

Method	Basic Concept	Pros	Cons
Kaynar and Sivrikaya [211].	Parallel and distributed memory-based algorithm, to generate vulnerability-based attack graphs.	<ul style="list-style-type: none"> • Full attack graph generation on multi-agent systems. • Distributed computation overcomes the state space explosion problem during graph generation when components and system sizes increase. 	<ul style="list-style-type: none"> • Will not protect against zero-day attacks, as method is reliant on identified vulnerabilities and utilises the NVD and CWE databases. • Leader agent is solely responsible for merging all partial graphs.
Li et al. [212].	Searching forward complete attack graph generation algorithm based on hypergraph partitioning, for large complex systems.	<ul style="list-style-type: none"> • Improves efficiency and load balancing on computing agents, by dividing subtasks • Generate attack graphs from vulnerabilities to attacker, based on vulnerabilities exploited assumption, therefore do not need to store states of the attacker on nodes. 	<ul style="list-style-type: none"> • Requires evaluation and expansion of vulnerability knowledge, and improve attack templates. • Further evaluation required to determine the influence on graph generation and merging of attack graphs, due to subtasks division and the different partitioning results and load balancing parameters, and different networks.
Nichols et al. [213].	Reduce the state explosion problem by the addition of priorities into exploit models of hybrid attack graphs.	<ul style="list-style-type: none"> • Reduces graph generation time without losing important data. • Advances hybrid attack graphs in regards to modelling reactive behaviour and exploits over multiple time-steps. 	<ul style="list-style-type: none"> • Further research required to evaluate exploits occurring at any time with no time-step. • Requires further research into prioritising exploits, to determine the likelihood, difficulty, or time it would take to exploit the attack path.
Chejara et al. [214].	Based on conditional probability, the method assigns each possible attack path within the attack graph an attack path score.	<ul style="list-style-type: none"> • Assigns attack paths with scores to assist with graph analysis, identifying the most critical paths. 	<ul style="list-style-type: none"> • Requires evaluation against other types of vulnerabilities and attacks. • Data set is not automated to keep track and up to date with latest vulnerabilities.
Polad et al. [215].	Introduce fake vulnerabilities into a system, in order to force an attacker to invest additional resources to reach a goal state.	<ul style="list-style-type: none"> • Establishes a stronger defence mechanism, and forces attackers to expend resources. 	<ul style="list-style-type: none"> • Requires further development to establish the strongest nodes and physical locations to host deceptive vulnerabilities.
Johnson et al. [216].	Attacker-centric probabilistic threat modelling technique for automated risk identification and quantification.	<ul style="list-style-type: none"> • In-built threat analysis. • Quantified attack graphs are populated with probability distributions and time to compromise for each attack step. 	<ul style="list-style-type: none"> • Requires further development to establish the method's accuracy, and implementation against real world systems.
Sun et al. [217].	Graphical model to interconnect mission dependency and Cloud-level attack graphs.	<ul style="list-style-type: none"> • Increases cyber resilience analysis of mission critical systems. 	<ul style="list-style-type: none"> • Extends the attack graph generation MulVal tool, which struggles and is limited by its ability to scale.

Polad et al. [215] utilising an attack graph model, examine the use of injecting false vulnerabilities into a network in order to establish if the process can distract a malicious attacker and increase the resources an attacker would require in order to penetrate the network.

The paper by Johnson et al. [216] outline a technique called pwnPr3d, which is a risk analysis method combining network architecture modelling language and an automated probabilistic inference engine to generate attack graphs. It also provides details on a quantitative estimation method utilised for information security risk.

The work by Sun et al. [217] examines impact assessment and cyber resilience, and proposes a novel mission impact graph model for Cloud environments. The attack graph model extends the MulVal tool and combines mission dependency graphs and Cloud-level attack graphs, in order to increase the resilience of critical systems.

The failings of these existing schemas include the inability to accurately identify the relationships and interdependencies between the risks and the reduction of attack graph size and generation complexity. Many existing methods also fail due to the heavy reliance upon the input, identification of vulnerabilities, and analysis of results by human intervention. When we consider the dynamic nature, size, and complexity of SoS it is unclear if these methods could be applied effectively to secure these infrastructures.

3.5 Information Assurance

Organisations, governments, and individuals have become heavily dependent on ICT, and are reliant on the processing and storage of confidential and critical information on these systems, which in turn has become a target for cyber-attacks. We have critically reviewed information assurance solutions to gain a better understanding of the methods utilised to protect data in both secure and insecure infrastructures, and establish the limitations of solutions that leave data insecure and exposed to risk vectors as data is created, stored, and transmitted within SoS. The following researched methodologies all endeavour to analyse and develop methods that attempt to assure cyber data, a summary of the methods is provided in Table 3.18:

One of the most common protection measures for network security is intrusion detection and prevention systems, which have been increasingly studied and can monitor data being transmitted between integrated systems. These systems are heavily reliant upon the signatures of known attacks, and require an in-depth knowledge of the vulnerabilities associated with the protocols which are to be monitored. Intrusion detection systems also struggle to monitor some protocols, for example protocols such as DNP3 and Modbus which can be relied upon within smart grids and critical infrastructure using SCADA controls [20] [21].

Ashfaq et al. [218] propose a fuzziness based semi-supervised learning algorithm, to improve classifier performance on intrusion detection datasets. This is achieved by applying unlabelled samples assisted by a supervised learning algorithm, in an attempt to increase intrusion detection systems classifier performance.

Cryptography and key management also continue to be heavily researched, as it can assist in overcoming many of the challenges associated with ensuring data integrity. However while many SoS use open standards such as Internet Protocol Security (IPSec) and Secure Socket Layer (SSL) which are compatible with encryption, there are still many protocols and legacy systems currently in use that do not support such features. For example DNP3 does not support authentication or encryption [20]. Won et al. [219] present a rule management protocol for assured mission delivery networks, which is based upon certificateless cryptography. The method endeavours to support authenticated rule registration, update, and deregistration via a one-way transfer message.

Hong and Sun [220] present an approach to guarantee adaptable and secure data sharing in mobile multimedia sensor networks. The attribute-based encryption technique developed utilises a novel efficient key updating mechanism that reduces impact on computational resources during attribute revocation and key exposure.

Understanding how legitimate users can access and control data and components within SoS is vital, insider threats at times can be considered more dangerous than external threats, as insiders with malicious intent will have a greater understanding of, and access to, key systems. Subsequently malicious attacks by insiders will potentially result in deeper impacts due to their knowledge and authorised access [12] [221]. In order to secure networks, data access controls are put in place in order to ensure that access to sensitive data and systems is restricted. The work of Li et al. [222] presents a data access control scheme for multi-authority Cloud storage systems, providing a two-factor protection to ensure the privacy of outsourced data.

Fugkeaw and Sato [223] discuss methods that utilise ciphertext update and proxy re-encryption techniques, then introduce a novel policy updating algorithm and proxy re-encryption method for secure access control in big data environments.

Data flow analysis has also been heavily researched as it is essential that organisations understand the flow of data as it traverses across systems and programs. The work of Sampaio and Garcia [224] provides a comprehensive insight into techniques for detecting security vulnerabilities within software programs, and review late detection and early detection techniques in order to improve secure programming. Exploring context-sensitive data flow analysis which can reduce the restrictions of pattern matching and improve vulnerability detection, the authors focus on early detection and develop a vulnerability detection method applying an abstraction of data flow analysis.

Szabó et al. [225] present a method called MPS-DF, which is an abstraction of the Meta Programming System (MPS) language component which supports data flow analysis. By defining data-flow builders for the analysed language, the builders generate subgraphs which assist with the development of data-flow graphs; these represent the data flow of the program which has been analysed. The MPS-DF methodology then analyses the data-flow graph and computes data-flow specific knowledge in regards to the program, the generated knowledge can then be utilised by existing MPS components.

Maintaining operational relations on a daily basis between distinct systems within SoS is essential. Ensuring security does not negatively impede genuine and time critical communications during operations, as safeguarding data and maintaining an effective communication network is vital. We must also consider how to safeguard these diverse collaborative networked cities, especially in regards to IoT and WSN, along with securing routing within the topology between dissimilar devices. Numerous approaches for secure routing within these types of networks have been researched and proposed. These include developments for WSN within smart grids to support secure communications.

Yan et al. [226] focus on the security requirements of smart grid communications, and present a light-weight and low cost solution, as cryptographic solutions impact sensor nodes limited resources. The methods presented include a digital watermarking algorithm which is an abstraction of alternating electric current, a digital watermarking algorithm which is an abstraction of time window, and a digital watermarking secure framework.

Glissa et al. [227] focus upon data transfer and routing within IoT, and propose a new secure protocol based upon Routing Protocol for Lowpower and Lossy Networks (RPL). Their protocol Secure-RPL attempts to prevent rank manipulation by generating a rank threshold and adopting a hash chain authentication method, limiting the decrease and increase of rank values and impeding malicious nodes from exploiting rank modifications.

A novel approach is also proposed by Wang et al. [228] outlining an addressing-based routing optimisation scheme to be applied to WSN which use IPv6, applying their method within a vehicular scenario.

Farooqi and Khan [229] adapt the low-energy adaptive clustering hierarchy (LEACH) protocol, to include intrusion detection principles in an attempt to protect WSNs from sinkhole, black hole, and selective forwarding attacks.

Table 3.18. Data Assurance Methods Summary

Method	Basic Concept	Pros	Cons
Ashfaq et al. [218].	Improve intrusion detection systems classifier performance.	<ul style="list-style-type: none"> Improves classification accuracy. 	<ul style="list-style-type: none"> Requires further development in order to efficiently detect multiple types of attack.
Won et al. [219].	Rule based management protocol established via the principals of certificateless cryptography.	<ul style="list-style-type: none"> Supports authenticated rule registrations and updates with non-repudiation. Avoids the need for certificates. 	<ul style="list-style-type: none"> Private keys are generated by a key generation centre, which is not the owner of the key yet it can decrypt ciphertext and place data at risk.
Hong and Sun [220].	Attribute-based encryption solution to support flexible and secure data sharing in mobile multimedia sensor networks.	<ul style="list-style-type: none"> Only authorised users can access the encrypted multimedia data. Reduces computation cost and energy consumption. 	<ul style="list-style-type: none"> Requires further development to establish the methods efficiency and operation against physical systems.
Li et al. [222]	Data access control scheme for multi-authority Cloud storage systems.	<ul style="list-style-type: none"> Users are required to hold sufficient attribute secret keys to access policy and authorisation key for the outsourced data. 	<ul style="list-style-type: none"> Only support the AND_m access policy.
Fugkea w and Sato [223].	Policy updating algorithm and proxy re-encryption solution secures and supports access policy evolution in big data Cloud environments.	<ul style="list-style-type: none"> Reduces computational cost. 	<ul style="list-style-type: none"> Requires evaluation against larger physical topologies, with increased number of attributes and transactions. Requires additional analyses of decryption performance in regards to constant size ciphertext.
Sampai o and Garcia [224].	Improve vulnerability detection and mitigate the limitations of pattern matching.	<ul style="list-style-type: none"> Identifies eleven security vulnerabilities that stem from input and output not being cleaned. Heuristics can be added or removed without interfering with the other heuristics, and heuristics can be adapted and implemented to other programing languages. 	<ul style="list-style-type: none"> Solution has high memory usage. Does not consider containers, reflection, and InnerClasses, thus generates false negatives.
Szabó et al. [225].	Analyses data-flow graphs and computes data-flow specific knowledge in regards to the program, generated knowledge can be utilised by existing meta programming components.	<ul style="list-style-type: none"> Can be used with several open-source and commercial projects based on domain-specific languages for embedded systems, insurance, and high performance computing. 	<ul style="list-style-type: none"> The inter analysis is eleven times slower in comparison to the intra mode.
Yan et al. [226].	Light-weight and low-cost security solution based on digital watermarking for home area networks and WSN in smart grids.	<ul style="list-style-type: none"> Algorithmic security is better than that based on alternating electric current. 	<ul style="list-style-type: none"> Compared to similar solution computational complexity increased Fails to determine the optimal number of watermark digits for the time window based watermarking algorithm.
Glissa et al. [227].	Routing Protocol, which introduces hash chain authentication and rank threshold to limit the effect of rank manipulation.	<ul style="list-style-type: none"> Mitigates the gravity of attacks in terms of resource depletion, node saturation, topology disruption, and network unreliability. 	<ul style="list-style-type: none"> Leaves systems vulnerable to attacks that are not based on rank.
Wang et al. [228].	Addressing based routing optimisation method for Low-Power Wireless Personal Area Networks.	<ul style="list-style-type: none"> Utilising one addressing process, each nodes can be configured with an address, thus addressing cost and latency are reduced. 	<ul style="list-style-type: none"> Expands IEEE 802.15.4 command frames, requires additional analysis against differing link protocols.
Farooqi and Khan [229].	Adaption of the low-energy adaptive clustering hierarchy protocol, to assist with intrusion detection.	<ul style="list-style-type: none"> Detect sinkhole, black hole, and selective forwarding attacks in WSN. 	<ul style="list-style-type: none"> Increases throughput, further analysis should be undertaken to determine its impact on system resources.

3.6 Network Optimisation

As ICT continues to be rapidly deployed and utilised, network optimisation approaches have been heavily researched in an attempt to assure network enhancement. Optimisation methodologies can be beneficial for large infrastructures and systems as they can be utilised for example to identify alternative network configurations. The following researched methodologies all endeavour to analyse and develop optimisation methods that attempt to improve communications within cyber networks, a summary of the methods is provided in Table 3.19:

Kumrai et al. [230] focus on the development of a multi-objective particle swarm optimisation (MOPSO) method to assist with Cloud brokering. The method identifies appropriate links between clients and service providers, this optimisation provides a solution for Cloud brokering by assisting to find solutions that lower the energy consumption of the service provider and response time requests, in addition to optimising profit for the Cloud broker.

In an endeavour to identify defective wireless access points and optimise interference within wireless local area networks, Yao et al. [231] propose a self-organising feature map (SOM) neural network model, using simulation techniques to generate results and evaluate their methods against nineteen access points.

The work of Rullo et al. [232] focuses on the security of IoT networks and the allocation of security resources. Using an abstraction of game theory they propose a Pareto-optimal solution, endeavouring to reduce the cost of infrastructure security, energy consumption, and probability of attack.

Zhao et al. [233] explain their interpretation of service risk assessment, and examine external risk factors which can impede communication links and nodes. They explain how they use services, link, and nodes to generate a risk model for key services, and then present their optimisation techniques which are an abstraction of Dijkstra algorithms, with different weights to reduce key service and network risks. In this method Zhao et al. do not analyse risk equalisation between link risk and node risk.

Yun et al. [234] discuss high performance networks which have been specifically developed to overcome the associated issues with the transfer of big data. They develop a cohesive framework to identify systems and network resources, and generate end-to-end paths for big data to traverse. The developed optimisation algorithms are evaluated by simulating and comparing them against a greedy approach, with several experiments being conducted against specific sections of a physical network.

The work by Alfarhan and Alsohaily [235] critically analyses self-organising wireless networks, they consider long-term evolution systems, and identify several network parameter optimisation challenges associated with the development of these types of network. Alfarhan and Alsohaily propose a Mixed Integer Quadratic Program optimisation technique for each of the identified challenges (optimisation

of frequency channel assignments, tracking area codes, physical cell identifiers, and long-term evolution).

Li et al. [236] outline the importance of complex network clustering, and present a novel quantum-behaved discrete multi-objective particle swarm optimization (QDM-PSO) algorithm. The aim of this research is to improve the performance of parallelisation for discrete particle swarm optimisation, then apply the improved technique to assist with complex network clustering in large scale networks.

Table 3.19. Network Optimisation Methods Summary

Method	Basic Concept	Pros	Cons
Kumrai et al. [230].	Ascertain the appropriate connections between clients and service providers, in order to improve service provider's energy consumption, Cloud broker's profit, and client request response times.	<ul style="list-style-type: none"> Reduces the response time and energy systems energy consumption. 	<ul style="list-style-type: none"> Requires evaluation against larger networked systems.
Yao et al. [231].	Self-configuring based method, utilising self-organising feature map neural network model, which trains the model in order to optimally solve interference in wireless local area networks.	<ul style="list-style-type: none"> Quickly identifies faulty access points in different conditions. 	<ul style="list-style-type: none"> Only configures nineteen access points for evaluation. Requires evaluation against larger networked systems.
Rullo et al. [232].	Game-theoretical model to minimise security cost, energy consumption, and the probability of attack.	<ul style="list-style-type: none"> Computes best defender strategy, allowing for requirements to be met in regards to resource allocation. 	<ul style="list-style-type: none"> Assumes the attacker will compromise at least one security resource, is aware of the defence strategy, and targets the most critical security resource.
Zhao et al. [233].	Reduce service risk in smart grid communication network.	<ul style="list-style-type: none"> Reduces both service risks and network risk. Optimised paths meet the time delayed standard. 	<ul style="list-style-type: none"> Does not consider risk equalisation between link risks and node risk.
Yun et al. [234].	Supports big data transfer in large-scale scientific applications within wide-area networks.	<ul style="list-style-type: none"> Generates an optimal end to end data transfer path for user requests, accurately modelling existing services, and improves big data transfer. 	<ul style="list-style-type: none"> Does not consider multiple conflicting user requests. Requires further evaluation against differing high performance networks.
Alfarhan and Alsohaily [235].	Identify network parameter optimisation issues within Long-Term Evolution systems, and develop Mixed Integer Quadratic Program optimisation models for identified issue.	<ul style="list-style-type: none"> Reduces optimisation cost, best lower bounds, and relative gaps. 	<ul style="list-style-type: none"> Some tracking area codes remained unassigned to cells during the optimisation process.
Li et al. [236].	Extend MapReduce, integrating quantum-behaved particle swarm optimisation, to achieve parallel and distributed quantum-behaved particle swarm optimization.	<ul style="list-style-type: none"> Increased solution performance, and reduced running time cost. 	<ul style="list-style-type: none"> Requires further evaluation against larger networks, and constructed using additional servers.

3.7 Summary of Existing Methodologies

Table 3.20 provides a comparison of the theoretical and applied solutions analysed and presented in this section against notable criteria in the aims, objectives, and challenges discussed in Sections 1.2 and 1.3. A summary of the current theoretical and applied solutions critically analysed in this section

is provided in Table 3.21, highlighting the main advantages and disadvantages of their application within SoS environments.

Table 3.20. Comparison of Analysed Methods Against Solution Requirements

	Network Assessment							Risk Assessment						Risk Management					
	Network Discovery	Vulnerability Analysis	Vulnerability Scoring Methods	CVSS v3	Vulnerability & Exploit Repositories	NVD	Network Security Assessment	Data Access Control	Risk Analysis	Attack Graph Generation & Analysis	Topological Vulnerabilities (Centralities)	Communication Security	SoS Robustness Identification	Risk Mitigation	Optimisation Methods	Genetic Algorithms	Ant Colony Optimisation	Local Search	Tabu Search
Accurate.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Assures data.	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓
Automated process.	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Considers multiple attack vectors.	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
Considers wider environmental factors.	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Defines relationships.	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Does not increase computational complexity.	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗
Does not impact systems when applied.	✗	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Does not introduce or expose systems to additional risk.	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Easy to implement.	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Evaluated within large SoS environments.	✗	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Expandable solution.	✗	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
Identifies interdependencies.	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mitigates risk.	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓
Non-domain specific.	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Non-reliance upon expert knowledge or perspective.	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
Non-reliance upon external methods.	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
Non-reliance upon single agents for graph generation.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Protection against zero-day attacks.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Quantify security for entire collaborative infrastructure.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Quantify security using multiple methods & factors.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Secures infrastructure utilising existing resources & systems only.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

Requirement to some extent met or technique capable = ✓, requirement not adequately met or technique not capable = ✗.

Table 3.21. Summary of Reviewed Theoretical and Applied Solutions

Method	Description	Subcategories	Generalised Pros	Generalised Cons
Security	Defends cyber based networks against malicious attacks, by detecting and responding to assaults.	<ul style="list-style-type: none"> • Malicious attack. • Detection methods. • Intrusion detection. • Intrusion prevention. • Development life cycle security. • Network security. 	<ul style="list-style-type: none"> • Different alert levels. • Real time analysis. • Accurately detects specific attack types. • Can maintain high system performance. • Can reduce false positive rates. • Reduces internal and external attacks. 	<ul style="list-style-type: none"> • Theoretical. • Requires evaluation against different configurations, topologies, and SoS. • Fails to consider other types of attack. • High computational overhead. • Domain specific. • Learner based solutions can be trained by attackers to ignore behaviour. • Signature based solutions are reliant on known attacks.
Risk Analysis	Process to identify and assess systems for risk.	<ul style="list-style-type: none"> • Risk analysis based techniques. • SPoF prevention and detection. • Cascading failure prevention and detection. • Interdependency detection. • Complexity detection. • Emergent behaviour detection. 	<ul style="list-style-type: none"> • Automated detection. • Considers different attack types. • Can classify attackers. • Customisable analysis. • Penetration testing methods provide support for improved security. • Supports decision making. 	<ul style="list-style-type: none"> • Limitations with vulnerability identification and securing risks. • High false positive rate. • Reliant on perspective and knowledge of administrators. • Requires evaluation against different topologies and SoS environments. • Theoretical. • Limited by reliance on external methods and databases and their associated limitations and failings. • Can negatively impact analysed systems. • Struggles to define relationships. • Difficult to implement, time consuming, and can be expensive • Fails to consider wider environmental factors and other infrastructures.
Risk Management and Assessment	Having identified vulnerabilities within networked systems, the process is responsible for how risks are efficiently managed and assessed, in order to mitigate those risks and increase system security.		<ul style="list-style-type: none"> • Offline assessment methods. • Compatible with existing monitoring technologies. • Endeavours to mitigate risk and eliminate vulnerabilities. • Quantitative and qualitative network assessment. 	<ul style="list-style-type: none"> • Theoretical. • When managing risks, difficult to predict how solutions will impact systems. • Reliant on human assigned vulnerability scores, limited by the perspective, bias, and knowledge of experts. • Limited by the functionality of the systems monitoring tools. • High false positive rates. • Requires evaluation against different topologies, SoS, and vulnerabilities.
Risk Modelling	Allows for networked systems to be represented as a network model, and can assist to quantify associated risks and vulnerabilities, along with representing exploitable paths.	<ul style="list-style-type: none"> • Network Modelling. • Attack graph generation. 	<ul style="list-style-type: none"> • Generate attack graphs from vulnerability to attacker, based on exploitable vulnerabilities. • Assigns paths with scores to assist with analysis. • Assists to establish stronger defence and force attackers to expand resources. 	<ul style="list-style-type: none"> • Reliant on external vulnerability databases. • Does not protect against zero-day attacks. • Requires further evaluation against other types of vulnerabilities and attacks. • Requires further evaluation regarding merging of graphs, often single agents are responsible for graph generation. • Reliant on human assigned vulnerability scores, limited by the perspective, bias, and knowledge of experts.
Information Assurance	Process secures and mitigates risks that pose a threat to data as it is created, stored, processed and transmitted.		<ul style="list-style-type: none"> • Reduce computation, cost, and energy consumption. • Mitigate attacks. 	<ul style="list-style-type: none"> • Theoretical • Requires development in order to detect and protect against multiple attacks. • Not applied to SoS. • High memory usage and increases computation complexity. • Can leave systems vulnerable to attack.
Optimisation	Method of improving or identifying optimum solutions and configurations.		<ul style="list-style-type: none"> • Reduce network risk. • Reduce computation. • Increase solution/system performance 	<ul style="list-style-type: none"> • Not applied to large SoS environments. • Theoretical. • Focuses on specific resources and risks, and ignores multiple risk vectors.

Reflecting on the work we critically analysed against our solutions criteria and its suitability to be applied within large multi-level SoS environments, it is evident that there are significant limitations with existing methods and techniques. Tables 3.20 and 3.21 determine these limitations and failings, and identify that no single method or technique is capable of identifying vulnerabilities and mitigating risks within multi-level SoS in order to increase communication security and robustness.

3.8 Summary

This chapter provides a critical review of related work that has been researched and developed in order to improve the security and robustness of SoS. In this chapter we have reviewed the methodologies developed to secure and defend cyber networks from internal and external risks, and provide an overview of the limitations and challenges that impact the methods when considering their application within SoS environments. It has also examined a variety of approaches that can be utilised to identify and quantify a variety of risk factors within SoS. These approaches were examined closely to identify their effectiveness to ensure that organisations applying these techniques can identify vulnerabilities that expose systems and understand the risks and potential consequences that can occur.

This chapter also reviewed risk management, assessment and modelling techniques, identifying the scope of current research and assessing the problematic application and limitations of these frameworks. It also analysed the concept of information assurance, and existing methods that have been researched and developed to ensure that access to sensitive data and systems is restricted within SoS. Lastly, the chapter summarised the field of network optimisation and relevant methodologies that attempt to identify alternative network configurations and enhance network security and robustness.

A common recurring issue in all the areas critically assessed in this chapter is that current approaches tend to be highly theoretical or implemented on small to medium standalone networks, failing to be applied to large dynamic SoS that have been formed using a varied combination of devices with varying security levels, multiple access points, and are geographically dispersed. We perceive that all unidentified cyber vulnerabilities and risk between integrated components which form part of an SoS have the potential to leave all collaborative systems and devices exposed and vulnerable to attack vectors.

In general, proposed research and solutions attempt to rectify and overcome a single specific challenge, but are ineffective due to their limitations, and the challenges and complexity of the environment in which they are applied. Having conducted an in-depth literature review, we conclude that there is no single solution or method that can conduct risk analysis and calculate the security level

for the entire multi-level SoS environment, utilising vulnerability analysis, node property aspects, topology data, and other factors, in order to mitigate risks without introducing additional resources into the multi-level SoS infrastructure

This chapter corroborates why a novel approach to optimise the level of security risk and mitigate risks within multi-level SoS is vital, and the deficiencies of existing methodologies form the inspiration and help ascertain the motivation and objectives for the proposed solution in this thesis.

Chapter 4

SeCurity Risk Analysis and Mitigation

(SCRAM) Framework

Information and Communication Technology (ICT) advancements have assisted distinct organisations to ‘pool resources’ during crisis situations, and these dynamic ad-hoc communication networks are abstractions of Systems-of-Systems. Working in partnership, diverse organisations have endeavoured to provide vital services to accomplish complex tasks that they could not individually achieve, often in challenging conditions and environments. In these crisis situations, organisations that typically do not interact on a daily basis such as emergency services (i.e. police, fire, and ambulance services), hospitals, voluntary groups, military, government, and non-government organisations, can find that they are heavily reliant upon communications and the exchange of information in regards to events, hazards, and even the locations of citizens, for example.

While the paradigm of the Internet of Things, Smart Cities, social media, and a variety of diverse SoS in coming years will assist the effectiveness of responders, when dissimilar technologies are combined during crisis management operations, access safeguards between these systems will be crucial, as any failure or delay which impedes communication can result in severe consequences. When integrating technologies which are distributed, formed from varying components, with differing security levels, it is vital that we consider how new policies and standards will be forced upon the systems forming the collaborative infrastructure. As insufficient or conflicting security policies, and unrestricted or inappropriate security levels for legitimate users, could expose the entire SoS, measuring the security properties of these types of infrastructures is highly difficult, and it is one of the most important challenges which must be overcome to prevent serious flaws exposing entire infrastructures, organisations, governments, or even towns and cities [237] [238].

Existing security and risk methodologies fail to adequately protect all components and networked systems which have been integrated together to form the SoS [5] [239], leaving the entire collaborative infrastructure exposed to vulnerabilities, failings, and potential attack vectors. The research presented in the previous chapters corroborates that existing research and developments fail to overcome the many challenges which impact SoS, and supports the need for a new novel solution that is capable of identifying risks and vulnerabilities within multi-level SoS, along with quantifying the security of the entire multi-level SoS, which in addition has the ability to overcome the challenges previously discussed, such as system complexity and evolutionary development.

Having critically reviewed existing solutions, and discussed the failings and limitations of developed techniques currently utilised to secure networked systems and identify risks within collaborative environments, to advance this research, we propose in this chapter a novel framework for measuring and optimising security, and mitigating risk within SoS environments, without introducing additional resources into the infrastructure, overcoming the associated challenges of measuring security between interconnected components and systems, the identification of risks and interdependencies within dynamic SoS, and data assurance within insecure networked environments.

The chapter contents are as follows. Section 4.1 presents a problem analysis that synthesises the limitations and issues identified from the conducted review of existing research methods and developed techniques. In Section 4.2 a high-level overview of the SCRAM framework discussed, whereas Section 4.3 provides an in-depth overview of the proposed solution's design. In Section 4.4 a detailed explanation of the SCRAM framework's runtime operation is provided. Section 4.5 summarises the data access control problem and management, whilst Section 4.6 discusses network centralities and the algorithms used to create the graph centralities within the framework. In Section 4.7 we outline the risk mitigation process, and provide a detailed description of node security grade assignment, including vulnerability analysis and scoring, the robustness function algorithm, and the evolutionary risk mitigation algorithm and comparative algorithms. Finally, in Section 4.8 a summary of the proposed SCRAM framework is given.

4.1 Problem Analysis

Due to the real-world failings of SoS we conducted numerous case studies in an attempt to identify the prominent issues impacting collaborative infrastructures. Reviewing not only issues associated with SoS, but also environmental impacts that can impede cyber systems. Through critical evaluation of these case studies, we ascertained that the majority of failings could be directly attributed to organisations failing to perceive or identify risk(s) which leave their networked systems exposed and vulnerable (as discussed in Section 2). Research corroborated that the theoretical and applied methods that attempt to overcome and assure systems against such issues are inadequate (discussed in Sections 2 and 3). We ascertained that organisations struggle to implement these solutions in complex collaborative environments, with methods being vague and generalised, domain specific, and often unsuitable or incapable of being applied to entire infrastructures or all collaborative networked systems which are often managerially independent.

Organisations find it problematic to manage their own security and risk effectively, never mind that of their collaborative partners. They can also be limited by financial restrictions, and can be forced to conduct expensive and vigorous testing prior to implementing risk and security solutions to prevent negative impacts occurring. The use of network and risk assessment methods can be just as hazardous

as the introduction or removal of key systems, as both can easily introduce additional risks that expose the infrastructure and any collaborative systems to unknown vulnerabilities and potential SPoF. The complexity and size of SoS means it is also difficult to quantify if the SoS is optimally configured in terms of its communication security, i.e. difficult to quantify if the SoS is as secure as it can be. Meaning it is difficult to ascertain if the security of data is being assured as it traverses across the collaborative environment.

Within our undertaken research we have found no single solution that is capable of successfully securing a complex multi-level SoS for its entire life cycle, and which is broad enough to be applied to dissimilar and dynamic environments and be non-specific to a particular vulnerability or attack. These rigid and unsuitable methods have directly resulted in SoS being susceptible to zero-day attacks and failings, with critical vulnerabilities remaining unidentified and system security not reflecting the SoS true status. Similarly, solutions are also failing to mitigate risks, organisations find it problematic to identify their own vulnerabilities accurately and eliminate them, and it becomes a bigger challenge to identify risks associated with the systems of their collaborative partners; who is responsible for monitoring and managing those risks; and who is accountable for identifying and applying the relevant resolutions.

As risk is unavoidable, there will always be risk factors to contend with, though if risk can clearly and precisely be identified prior to failings or attacks occurring, then this early identification would ensure that we can both secure and manage risk more effectively. This in turn would increase the robustness of the SoS and allow for risks to be mitigated prior to their exploitation or failure. Additionally, if the risk or vulnerability could not be mitigated then early identification ensures that the vulnerability can be monitored and managed more effectively, so in a worst case scenario the appropriate strategy plans are in place to limit its impact.

The work in this thesis is motivated by wanting to address what we perceive to be the three most problematic challenges, which are measuring security between interconnected components and systems, the identification of risks and interdependencies and their mitigation, and data security in unsecure and unencrypted networks (detailed in Section 1.2).

4.1.1 Aims Analysis

It is evident from the sections previously presented that there are significant limitations and failings associated with SoS risk identification, mitigation, and security. These problematic issues summarised in Table 3.21 demonstrate the need for a new novel approach, additionally, the critical review of existing methods and techniques summarised in Table 3.20 corroborate that currently there are no distinct viable solutions capable of identifying and mitigating security risks in large complex SoS.

The purpose of this broad and in-depth review of literature was to assist us in discovering the associated inadequacies of techniques, and support the development of a new solution based on the identified research gaps and the weaknesses that require significant enhancement in order for them to be more appropriate in their application within large SoS environments (i.e. multi-level SoS). Our methodology was also required to observe the aims summarised in Section 1.3, which were determined after reflecting upon this review and identifying the main problematic challenges.

Security – The problematic issues associated with SoS security, lead to our initial aim of developing a solution capable of measuring the security of individual devices and the entire SoS topology. Current techniques cannot state with certainty that communication security for the distinct components, systems, and the entire SoS is secure, and to what level vulnerable nodes weaken security exposing systems to potential attacks and failure. SoS security techniques are generally theoretical, have not been evaluated within these environments, have not been applied to dissimilar systems and are domain specific, focus on specific attacks or vulnerabilities, and signature based security solutions fail to protect against zero-day attacks. Through conducted case studies we have corroborated that if systems have unidentified vulnerabilities, then these insecure nodes have the potential to expose the entire SoS resulting in various consequences, including SPoF, negative emergent behaviour, cascading failure, and entry points for malicious attackers.

Risk and Interdependencies – Our goal is to identify risks and interdependencies that impact and form between collaborative components, this is motivated by both the problematic challenge of measuring security between interconnected components and systems, and in order to improve the problematic issues associated with quantifying SoS security. Current vulnerability, risk, and interdependency identification methods are theoretical, reliant on the perspective and knowledge of administrators, suffer with high false positive rates, can impact analysed systems, are difficult to implement, time consuming, expensive, have not been evaluated against large complex SoS, struggle to define relationships, are reliant on external methods and databases and are therefore impacted by their associated failings, and often are domain or vulnerability specific with the methods requiring heavy adaptation to be applied to differing factors. Improved risk and interdependency identification would increase the accuracy of communication security measurements for both devices and systems collaborating within the SoS, could reduce false positive rates, would assist to support the maintenance of high system performance, and the identification of interdependencies would assist to reduce SPoF and could reduce potential partial and full cascading failures.

Risk Mitigation – Similarly, there have been issues with the mitigation of identified risks and interdependencies. In order to improve the security of multi-level SoS and reduce risk vectors we aim to mitigate risks motivated by the limitations of risk management and assessment methods, which fail to efficiently manage and assess risks which in turn decreases communication security. Existing risk mitigation methods are generally theoretical, domain specific, do not consider the ramifications of

changes to systems, can be influenced negatively due to human bias, perspective, and knowledge, and typically focus on specific resources and risk, ignoring multiple risk vectors. Mitigating risks would improve the security of both the distinct device and systems within the SoS environment, and enhance the overall multi-level SoS security and robustness. Developing a solution that mitigated risks utilising the existing topological systems means that we could increase the security of the collaborative environment utilising only the existing networked resources. This means not only will vulnerabilities and the risks that they pose to the infrastructure decrease, but we can assure that the collaborative environment is as secure as it possibly can be.

Robustness – As stated, security and risk management and assessment methods have limitations, and struggle to identify the security of communication and provide an appropriate assessment of the network's systems in terms of the risks which expose the infrastructure. In order to improve security and mitigate risk effectively we endeavour to measure the robustness of the environment. Having the capability to quantify the robustness level of a networked infrastructure is important, this measure could help to ascertain in the event of failure or attack, how well the infrastructure will stand, i.e. will the dynamic nature and built-in redundancy of the SoS support the infrastructure's ability to meet objectives, or are there significant risks, potential SPoF, and dependencies within the SoS that will reduce the SoS capability to maintain operation in worst case scenarios. When mitigating risk this robustness level would also allow for the evaluation of the appropriateness of the changes that are applied to the infrastructure.

Data Assurance – SoS are reliant upon the transfer of data in order for them to maintain collaboration; as SoS can be formed from numerous dissimilar systems with conflicting data access requirements, securing data as it traverses across insecure and unencrypted networks is problematic. Therefore, we aim to develop a solution that overcomes the limitations of existing solutions that fail to assure data, to reduce potential risks that expose this element. Current solutions are in general theoretical, have high computational overhead, leave systems vulnerable to attack, are developed to protect against specific attacks and require development in order to detect other assaults, and have not been evaluated or applied to large complex SoS. Assuring data is of utmost importance within SoS, we consider data to be the biggest SPoF within these environments, as, if data cannot be created, stored, or transmitted within the SoS then systems could fail and full or partial failings could ripple across the entire collaborative environment.

By increasing communication security, accurately identifying risks and interdependencies that expose systems, quantifying infrastructure robustness levels, mitigating risks, and improving data assurance, we would have the capacity to secure the collaborative environment utilising only the existing networked resources and increase the cyber resilience of the SoS infrastructure. Which would mean in the event of component failure or network attack, impacts should be minimised due to the applied solution, or the solution can provide an early warning mechanism which would allow for unmitigated

risks to be clearly identified allowing for them to be managed more effectively thus reducing their impact.

4.1.2 Objectives Analysis

The objectives presented in Section 1.3 heavily contribute to solving the problematic challenges associated with measuring security between interconnected ‘things’, the identification and mitigation of risks and interdependencies, and data security in insecure and unencrypted networks, in addition the objectives assist us in adhering to the established aims of the research presented in Section 1.3 and discussed above in Section 4.1.1.

4.1.2.1 Background Literature Research

Our primary objective was to conduct detailed background literature research into the challenges and risks that expose SoS, the issues impacting the ability for current solutions to secure these dynamic networked infrastructures, and the methodologies that fail to identify and quantify risks within SoS.

In order to solve the problematic challenges associated with the measurement of security within SoS, it was imperative that the review of literature and case studies into real-world failings covered a diverse range of security aspects, risk factors, risk methodologies, and network topologies, including cyber issues and wider environmental factors which can impede SoS security and functionality.

For instance, when reviewing the issues that had caused critical failings within numerous UK banking infrastructures, we were able to determine that the problematic issues and cascading failures that had ensued could be attributed to varying risks and attacks, yet each of the separate issues had all resulted in similar consequences and loss of service, and each of the organisations had failed to identify the vulnerabilities in advance and could not guarantee that these risks had been effectively managed and would not occur again. After identifying the types of risks and attacks that had exposed these infrastructures, we reviewed the current methods and theoretical techniques that endeavour to secure these SoS against such vulnerabilities, which included risk identification, risk assessment, risk management, vulnerability analysis, vulnerability assessment, vulnerability scoring methods, exploit databases, attack graphs, intrusion detection systems, etc. In addition, we also investigated previous enquires that had been undertaken that questioned the financial institutes and banks, which allowed us to gain a better understanding of how previous banking failures had been managed and weakness that these institutes are still struggling to manage effectively.

This objective assists to ascertain the risks that expose SoS, issues impacting the functionality of current solutions to secure SoS, and weaknesses that prevent solutions from accurately identifying and

quantifying risks within SoS. This in turn, corroborates the difficulty in measuring the security between interconnected components and systems, assists to determine flaws with existing methodologies, and assists to identify promising techniques that could be useful if developed to process multiple risk vectors or function in SoS environments. In addition, our primary objective validates the need for the development of a solution to identify and mitigate security risks within SoS and multi-level SoS.

4.1.2.2 Security Risk Analysis

Our second objective was to develop an SoS security risk analysis solution to calculate the security level of the entire SoS using vulnerability analysis, node property aspects, topology data, and other factors, to improve and mitigate risks without introducing additional resources into the SoS infrastructure. This objective contributes to solving issues associated with measuring security between interconnected components and systems, the identification of risks and interdependencies and their mitigation, and data security in insecure and unencrypted networks.

Risk and vulnerability identification, assessment, and management methods are in general component, domain type, and risk specific, depending on a single specific methodology or vulnerability database, and are typically theoretical or not applied and evaluated within large complex SoS and multi-level SoS. The real-world case studies clearly establish that vulnerabilities remain unidentified, and organisations are failing to competently analyse their networked infrastructures and accurately quantify their communication security. Therefore, by organisations failing to accurately identify vulnerabilities, security scores do not reflect the true level of how secure their systems are.

This objective contributes to solving the problematic issues associated with the quantification of accurate security levels, achieved by implementing better identification, analysis, and reporting of risks. This is attained through the incorporation of numerous network and vulnerability analysis methods, including collating data in regards to node property aspects, topological vulnerabilities, and other factors which can be categorised as a risk as they pose a threat to data, devices, systems, or the entire SoS. By developing an adaptable solution that can combine a wide number of security risks from dissimilar methods and systems, we can gain a more realistic overview of the vulnerabilities which expose systems, this in turn will support their management and mitigation, i.e. the more sources we include identifying risks the more accurate scores will be, and we can be confident that they reflect the true security state of both the distinct nodes and the SoS.

A broader view of what is categorised as a risk within these large complex infrastructures, extending the number of risk parameters to be analysed (i.e. not focusing on a single type of attack or vulnerability), and improving the accuracy of risk identification, also supports improvements to risk

mitigation and securing the infrastructure without introducing additional resources into these SoS. Improvements to risk identification assist to quantify more accurate communication security scores that reflect the true status of the topology, meaning as changes are made to communication links in order to mitigate risks between insecure and vulnerable devices, reassessment of the network utilising the same methods will quantify if modifications have negatively or positively impacted the overall security of the infrastructure.

Furthermore, by quantifying and analysing topological vulnerabilities such as high connectivity vulnerabilities, shortest path vulnerabilities, SPoF, weighted high connectivity, and dependent communication vulnerabilities (discussed in Section 4.5), when we analyse the topology and attempt to mitigate risks within the SoS, we will be able to identify important relationships between nodes that expose the SoS and ascertain network behaviour characteristics. This objective assists to overcome the problematic challenges associated with risk identification and mitigation, and in part assist to support data security in insecure and unencrypted networks, by ensuring that the SoS being evaluated can be accurately measured in terms of its communication security.

4.1.2.3 Robustness Analysis

Our third objective was to develop a solution that can analyse and quantify the robustness of the SoS environment based on the relevant data captured from the application of the security risks analysis solution.

In order to overcome the problematic challenges associated with mitigating risk and increasing communication security, it was imperative to establish a means to determine if networked infrastructures were optimally configured. Robustness scores can be quantified utilising key criteria identified during the risk analysis process, and can assist to determine the network's appropriateness. This is important as SoS with inadequate topology configurations can expose data to differing risk vectors as data traverses across the infrastructure, and can increase topological vulnerabilities including SPoF and interdependencies, for example.

As risk mitigation methods are applied to the SoS under evaluation, the robustness level can also act as a comparable vector demonstrating the appropriateness of the overall infrastructure's security and network security configuration. Robustness scores can also assist to produce the next generation of improved solutions, and can be used in comparison to monitor modification impacts. It is vital to have the capability to evaluate how alterations negatively and positively impact the overall suitability of the infrastructure, to ensure that topology and security enhancement is achieved while preventing reverse evolution.

In addition, risk is unavoidable therefore it is important to have the capability to know in the event of failure or attack how robust and secure systems truly are, thus the robustness level can be used in conjunction with security reports during decision making processes and when establishing management plans when risks cannot be effectively mitigated further.

4.1.2.4 Optimisation Evaluation

Our fourth objective was to conduct a detailed investigation into optimisation techniques and algorithms in order to identify which solutions suit SoS to mitigate the risks.

In order to solve the problematic challenges associated with mitigating interdependencies and risks, and securing data in insecure and unencrypted networks, it was necessary to implement different optimisation techniques and algorithms into the developed framework in order to evaluate both the solution itself in regards to mitigating risks and quantifying SoS topology robustness levels and security scores, and in order to evaluate the difference in risk mitigation methods suitability and functionality when being applied into large complex SoS environments.

For instance, not all optimisation methods can be applied to large complex SoS, as they can struggle with complexity, scaling, require heavy adaptation, increase computational overhead, and can be too rigid for the dynamic environment, restricting the process and limiting end results. This objective corroborated the difficulty in evaluating the effectiveness of optimisation techniques within complex SoS environments, and assisted us to determine those methods that when combined into the risk mitigation function showed promise in enhancing networked infrastructures, mitigating associated risks, and increasing the overall security and robustness level of the topology, as well as demonstrating the methods that are adequately capable of generating and reporting new enhanced SoS topologies that were secure and more appropriate.

In addition, not only did this objective assist us in identifying the limitations of existing optimisation methods and the areas that require improvement, it supported the identification of the most effective techniques and algorithms applied against SoS, and corroborated the need for the development of a solution to mitigate risks and increase the robustness and security of the SoS utilising only the existing infrastructure's resources. For instance, WSN can be deployed in remote areas and can be inaccessible, if we evaluated this SoS and determined that the security and the robustness level was inadequate, then our proposed methods could be a suitable, inexpensive, and realistic approach to improving the appropriateness of the WSN without having to physically access the devices or deploy further infrastructure. Thus, the objective would solve the issue of ascertaining the best solution to mitigate risks to ensure that SoS are optimally configured and secure, which is vital to guarantee functionality is maintained, data is assured, and the infrastructure's robustness is increased.

4.1.2.5 Case Study

Our fifth objective was to conduct a case study on a specific network type such as WSN, and expand the solution to encompass a different risk vector utilising the same developed risk analysis framework and robustness techniques.

This objective contributes heavily to solving the issues that impact the ability of existing solutions to secure dynamic SoS, and the methodologies that fail to identify and quantify risks within SoS, as it supports the evaluation of our proposed methods, corroborates their capabilities, and ascertains the appropriateness of applying them within SoS environments. It was imperative to validate not only the proposed solutions, but our decision to use simulation rather than applying the methods directly to a physical SoS. The case studies conducted to meet this objective allowed us to experiment and enhance the techniques and algorithms that are integrated within the solutions framework without introducing vulnerabilities into the SoS being evaluated or cause critical consequences to occur. Instead the framework and proposed algorithms could be adequately tested and evaluated, and potential issues associated with the application of the framework could be monitored and assessed within the simulation environment.

These case studies also allowed us to run the same experiments against different optimisation methods, and enhanced techniques against the same topologies. This ensured that the results that were attained were accurate and a true reflection of the SoS topology (i.e. we could repeat the test multiple times and the end results would be consistent). This objective in turn corroborates the difficulty in measuring security, identifying vulnerabilities, quantifying the robustness of the entire topology, and mitigating risks within SoS, and validates that our solution can be an effective and appropriate methodology for mitigating risks and improving communication security within SoS environments.

In order to establish the effectiveness of the proposed solution's ability to incorporate different risk vectors and its usefulness, this objective aims to expand the solution to encompass a different risk vector. This will assist to determine the dynamic nature of the proposed solution, and corroborate the framework's ability to effectively function when new risk parameters are identified then incorporated into the solution, ensuring that the security, risk, and mitigation methods continue to accurately quantify scores and mitigate risks. The objective to encompass a different risk vector within the framework overcomes the problematic issues of other methods and techniques that fail to consider alternative vulnerabilities or are too rigid and require heavy adaptation. An adaptable solution will ensure that as new vulnerabilities are identified the method will be able to incorporate them, along with the solution being forward compatible allowing for new risks that are unimaginable now to be incorporated in the future.

4.1.2.6 Multi-Level SoS Analysis

Our final objective was to validate that the algorithms and principles are effective for identifying and mitigating risks within multi-level SoS, in order to increase multi-level SoS security and robustness.

This objective heavily contributes to solving the problematic challenges of measuring the security between interconnected components and systems, identifying risks and their mitigation, and data security in insecure and unencrypted networks. SoS are highly complex and difficult to accurately evaluate and secure; when we begin to consider multi-level SoS which are an accumulation of SoS, these infrastructures will be just as problematic and more complex.

From our case studies and review of associated literature we have identified problematic issues that struggle to identify vulnerabilities within distinct systems and secure them within distinct SoS. Our literature review also corroborated that many of the proposed solutions not only have severe limitations and require adaptation, but have never been applied to SoS that contain a significant number of distinct networked systems.

Evaluating the proposed framework against multi-level SoS will corroborate the usefulness of the techniques to measure the security of distinct devices, and the entire collaborative environment. In addition, it will provide the means to evaluate security grades and ascertain the security impacts that occur, when SoS are forced to collaborate with external SoS under independent management. This objective will also allow us to assess the suitability of the proposed algorithms when applied to multi-level SoS, specifically their ability to manage with the complexity and size of multi-level SoS topologies, their accuracy to quantify the security and robustness levels of the entire collaborative infrastructure, and corroborate their effectiveness to identify risks and mitigate them when considering the accumulation of SoS as a single entity.

4.1.3 Methodology

SoS failure is problematic despite great investment, research, and development. The purpose of this research is to develop a methodology that overcomes the limitations of existing solutions and advance the identification and mitigation of security risks in multi-level SoS environments, to achieve this we present in this section the methods adopted in order to undertake this research. Discussing in detail the research method chosen, followed by the research design, the data collection method, and conclude by providing a detailed explanation of the data analysis method selected.

4.1.3.1 Research Method

To achieve our objectives the research in this thesis uses a quantitative approach. In contrast to qualitative research, quantitative research allows for the generation of numerical data in order to statistically analyse results and quantify the problem. For example, in this instance the quantitative method will support the analysis of identified vulnerabilities and provide a numerical estimate of the risks they pose, achieved using predefined mathematical formula.

Whereas qualitative research would deliver a descriptive estimate of the risk that the identified vulnerability would pose to the device or infrastructure. While qualitative assessment is often utilised by risk assessment methods in order to limit the potential impact that the methods can have upon the physical infrastructure, and will assign risks based upon traditional collective measures such as questionnaires, this type of assessment can fail to perceive risks and produce poor estimates, as numerical evaluations would be based on the expert's skill level, training, and their ability to not influence scores and to remove their own bias.

Overestimated expert scores would result in vulnerabilities being identified as insecure due to analysts being overcautious, this would not reflect the true status of the infrastructure's security as it would appear weaker, and organisations could waste resources and time safeguarding vulnerabilities that do not pose such a threat. Alternatively, experts' underestimation of scores would result in vulnerabilities being identified as secure due to analysts' poor comprehension of threat severity, this would reflect an inaccurate level of high security, and organisations would fail to take action against vulnerabilities leaving their devices and systems exposed to potential failings and points of attack. This in turn could expose the entire collaborative infrastructure.

Research corroborated that experts' opinions can greatly differ, with vulnerability identification and assessment methods in recent years struggling with discrepancies to assigned vulnerabilities. In order for results to be accurate we feel a quantitative approach to be more suitable to limit the reliance on expert opinion and scoring. Computational techniques ensure that the assignment of numerical values to identified vulnerabilities, the quantification of the risk which is posed to the infrastructure, and the measurement of the devices and infrastructures is quantified in a scientific means and one that can be repeated.

Undertaking a quantitative method will allow us to implement and automate processes, which include the collection of data, algorithms to quantify node vulnerabilities, topological vulnerabilities, security grades, robustness scores, minimum path average, and cost in terms of the distance between nodes, etc., and the modelling and analysis of data, including the assessment and management of risk mitigation. Manual assessment, analysis, and management strategy for this research would most likely be impossible to conduct due to the complexity, size, and number of distinct systems that form the collaborative environments that we are going to apply the methods against.

4.1.3.2 Research Design

The primary objective of this thesis was to develop a solution to calculate the security level of SoS using varying vulnerability factors in order to mitigate risks without introducing additional resources into the infrastructure. To achieve this we will design and implement a new risk analysis and mitigation framework using a simulated environment.

Our decision to use simulation is motivated by wanting to develop a solution that can be applied in the future to differing infrastructures and topologies. Had we implemented our solution against a specific physical SoS we would not be able to evaluate the method's suitability to be applied against a range of dissimilar SoS or within multi-level SoS. Simulation provides us the means to replicate a diverse number of different components, system configurations, and SoS topologies, allowing us to quickly generate new environments and run experiments on different adaptable topologies and on different scales, which we could not physically construct due to project limitations and the varied components required.

Besides, we would not be able to apply the experimental and untested solution against a large operational SoS due to societal dependence on the assets of these infrastructures, and to prevent risks being introduced into the networked environment which could have the potential to cause operational failures. We feel this would be irresponsible and unnecessary at this stage of development, simulation is an acceptable and widely adopted method for testing hypotheses in this area of research, and can generate accurate results which our evaluation of the conducted experiment will validate.

Having ascertained the limitations of existing techniques we chose to take a broader view in regards to the types of risk that can expose SoS, choosing to integrate multiple different risk vectors into our solution including node property aspects, topological vulnerabilities, and other factors. This is to ensure that node security grades and SoS communication scores are accurate and a true reflection of the node's and networked system's secure status, and to support the measurement of the infrastructure's robustness. Rigid methods that focus on specific vulnerabilities or attack types are leaving SoS exposed, by collating data on different types of risk not only can we improve vulnerability identification and increase security scores, we can also in the same solution identify topological vulnerabilities which can indicate high connectivity, shortest path, SPoF, weighted high connectivity, and dependent communication vulnerability, and identify nodes that are in breach of data access violations which can expose data, and thus, have a greater understanding of the risks that expose data as it traverses across the infrastructures to maintain collaboration between systems.

To ensure our simulated vulnerabilities are a true reflection of real-world risks, we incorporated vulnerabilities directly reported via NVD applying the published CVSS v3 base scores assigned to each of the vulnerabilities within our method. CVSS and NVD procedures have been widely used in this field, with CVSS providing a standardised vulnerability scoring method and NVD providing an

open repository of analysed vulnerabilities. NVD details security-related software flaws, security check lists, impact metrics, product names, and misconfigurations that can be scanned for against physical cyber networks, for this reason we believe this database will assist to simulate real-world vulnerabilities and validate the method's accuracy for the identification of risks and security analysis.

As a result of this research we will have a framework that can accurately simulate all nodes and systems which form both SoS and multi-level SoS, and will have the capability to produce graphs that represent the SoS topology. Discovered risks and vulnerabilities will be reported within the framework and detailed reports will be constructed and stored, allowing for the infrastructures to be analysed prior to any risk mitigation process being applied against them. To ensure that graphs that represent the topology of the infrastructure can be analysed intuitively we have limited the number of risks that are visualised on the graph. Selected risks visualised include node bridging centrality score, vulnerability scan status, security status and grade, and data access levels and violations. All other risks, including topological vulnerabilities, minimum path average, and SoS communication security, are placed into the reports, with the robustness levels being reported separately in both its own graph and report.

In order to mitigate risks, increase communication security for the entire SoS, and assure data as it traverses across an insecure and unencrypted network without introducing additional resources into the infrastructure, we have implemented a method that reconfigures the network's communication paths using optimisation techniques, and considers node security status, data access violations, and high centrality node risk, choosing to utilise optimisation techniques as they overcome many of the limitations associated with local search techniques.

While our work was influenced by genetic algorithms which replicate the changes made in nature through evolution, to validate the methods and determine the most suited solutions for SoS risk mitigation we implemented two additional optimisation techniques (ant colony optimisation combined with local search and tabu search). The implementation of additional algorithms into the risks mitigation method also assisted in establishing the effectiveness of the applied methods and techniques, as they can validate that the risk identification, analysis, and scoring measurements are not erroneous, achieved by running differing experiments in regards to risk mitigation against the same topologies.

During the risk mitigation process enhancements are made to the topologies evaluated, and random configurations are generated and evaluated, meaning even when applying the same algorithm to the same topology different end configurations can be presented. We have to reflect on results and take a broader view of the analysis, as results will not be perfectly identical, i.e. if the methods are behaving similarly and tracked alterations during the mitigation process are consistent we can validate that the methods are functioning correctly. The robustness level graph and reports are an excellent indicator

for monitoring the effectiveness of our methods as when the risk mitigation technique is applied and enhancements are made to the SoS, they not only log every improved candidate, they report every candidate generated including those rejected.

Through the risk mitigation process the framework monitors the other risks that expose the infrastructure, by tracking node security grades, data access levels, and topological vulnerabilities, etc., if changes were made to the variables that should be static an error would be immediately identifiable and a good indicator that results were invalid. Furthermore, to ensure that methods are generating factual and valid measurements, we manually checked and quantified all factors and scores in regards to both the original SoS topology and the enhanced reported candidates to ensure that the results manually quantified are identical to those generated within the simulated framework

4.1.3.3 Data Collection Considerations

For the purpose of this research, we wanted to examine both distinct SoS and multi-level SoS that are an accumulation of SoS. To ensure that we could manually analyse vulnerabilities, and quantify measurements and scores for every distinct device, system, and infrastructure, we limited initial simulations to no more than 12 devices for SoS experiments, and 12 devices and 12 SoS for multi-level SoS experiments. When we apply the methods to large complex infrastructures that cannot be manually assessed we can be confident that quantified factors and scores are valid, due to our critical evaluation and comparison of manual and framework generated results.

While environmental factors can impede SoS and increase risk factors, we have omitted these types of risks at this stage of our research. Though it must be noted, to fulfil our objectives we have expanded our solution to incorporate a different risk vector in the form of a case study. This establishes while we have chosen to focus on cyber vulnerabilities and risks, the methods are not limited and could easily encompass other identifiable risks and produce more detailed and accurate security scores and assessments. We have chosen a range of risks to incorporate into the framework which assist to identify the security of the individual devices, the security of the entire infrastructure, and support the identification of devices which expose the infrastructures and risks which form between collaborative components.

The simulated framework can generate a range of devices and SoS topologies, the risk factors we chose to simulate and collate as part of the simulated network assessment and risk identification include connections, path lengths, data access levels, and if applicable operating system, firewall status, IDS, encryption details, associated staff levels, system update status, antivirus and security, if the device is connected to the internet, vulnerability scan conducted status, and whether the device has identified vulnerabilities.

We desire to simulate a diverse number of components and differing topologies, therefore not every node will be assigned vulnerabilities as part of the simulation, instead nodes are randomly assigned to have either been scanned with no vulnerabilities, scanned with identified vulnerabilities, or unscanned. Nodes that are identified as having been scanned with identified vulnerabilities are assigned a random number, this figure becomes the number of vulnerabilities that are then assigned to the node based upon what type of device it is and its operating system. These vulnerabilities are real-world vulnerabilities that have been imported into the simulated environment, which were reported on the NVD website in June 2016. This enables more accurate topologies to be generated, and supports the validation of the method's ability to incorporate different vulnerabilities into the quantification of node security grades, and its ability to compute more reflective infrastructure security scores and robustness levels.

These risk variables are then used for several purposes; firstly they are used to quantify node security grades, SoS communication security scores, the robustness level, topological vulnerabilities, minimum path average, and cost in terms of distance between nodes. Once we have a complete overview of the topology, undirected graphs can be generated to provide visual representation of the environment and the relevant risks as required.

Details are also reported into two different records for analysis and use, this includes the generation of a configuration file containing topological details such as node coordinates, connections, path lengths, data access levels, security grades, etc. The configuration files can be used to analyse the topology changes between the original network and the optimum reported enhanced candidate, but more importantly it can be used to restore the original network in order to evaluate different methods and validate the effectiveness of alternative risk mitigation techniques. Secondly, topology data, variables, and quantified results can be presented within the framework's report window for immediate analysis, reporting key risks and measurements to the individual nodes and entire infrastructure. All topological data, variables, quantified risk and security factors are stored within security reports; if the risk mitigation process is applied to the network then these reports are extended with the enhanced candidate details. This allows for a comparative analysis from the original network through the evolution of the topology, which demonstrates how the reconfiguration of the communication paths impacts not only the topology but each distinct node, and quantifies risk and security measurements.

When the risk mitigation method is applied to the original infrastructure topology, the process begins to enhance the infrastructure's security and reduce risks by reconfiguring the communication links between nodes. At this stage we decided not to keep detailed reports of every generated candidate, as data sets would quickly increase in size, especially as each risk mitigation process will generate 20,000 candidates for compression. Instead, full topology data, risk factors, and quantified results are added to the security reports for each of the reported enhanced candidates only, allowing for quick

comparative analysis to be undertaken while still providing enough details for critical evaluation of risk mitigation and any negative impacts to devices or the topology.

The generated robustness score graph produced by the framework similarly only shows the robustness scores quantified by the method for each of the reported enhanced candidates, this allows for intuitive evaluation of the appropriateness of the reported candidates. This graph excludes the negative evolvments that are rejected during the risk mitigation process; however, we do collate every robustness score for all generated candidates. This report ensures that we can evaluate the risk mitigation process itself. The method generates 20,000 network configurations, continually evolving the communication paths and carrying the best reported candidate of each round into the next stage for enhancement. Negative evolvments should be rejected and only the best solution carried forward, the robustness log allows us to evaluate the method by validating that the best candidate is correctly passed into rounds until a more appropriate generation is produced, and that the method does not have limitations with high numbers of false positives and negatives.

4.1.3.4 Data Analysis

The analysis of risks and the appropriateness of the risk mitigation process are conducted by the methods within the framework itself. We then manually replicate the calculations and evaluate both sets against each other, this is to determine the efficiency and accuracy of the application's ability to produce and analyse results, and to validate the effectiveness of the methods themselves to mitigate risks and secure the complex SoS and multi-level SoS environments.

In order to establish the optimum enhanced candidate, the risk mitigation and security enhancement method analyses every generated solution, as when the process adds and removes communication links between devices it must balance connectivity with improvements to the topology's robustness level and security without impacting centrality factors that can introduce additional risks to the network, and must consider the maintenance of a secure route between nodes that are not in violation of data access levels to assure data. The distinct optimisation methods when applied ensure that negative evolvment is eliminated, achieved by comparative analysis of the quantified robustness level and, when applicable, comparative risk parameters. Again the methods can be assessed and validated by analysing the chosen optimum solution and reported enhanced candidates.

Similarly, it is important to continuously analyse communication security throughout security enhancement and risk mitigation, as we are endeavouring to improve the overall topology's security or where applicable maintain the high level of security already achieved.

The security enhancement and risk mitigation process is also analysing the topology of the collaborative infrastructure, ensuring that nodes do not become isolated due to communication links

being removed and added. In addition, the process is analysing the communication links to ensure that paths are maintained between secure nodes and those that do not violate data access requirements. This analysis ensures that a secure path is maintained allowing for data to traverse securely across the networked infrastructure. The method can be manually validated by analysing the configuration file and the undirected graphs generated and reported by the framework.

It must be noted that while the framework analyses all enhanced candidates and reports the optimum solution, we conduct manual analysis against all of the reported candidates to establish their usefulness. For example, are there reported enhanced candidates that, while not considered the optimum solution, assure security and improve the robustness of the infrastructure that could be adequate and should be considered during decision making processes, as they are cheaper to implement and should be considered as an alternative solution if budget restrictions are necessary.

We are also highly interested in analysing the topological vulnerabilities that can be introduced or altered due to reconfiguration of the infrastructure. By examining the quantified centrality values for each node and the aggregated centrality scores for each of the reported enhanced candidates, we can understand the impacts that occur and identify risks that expose the infrastructure as configurations are altered and new relationships are established.

All topology data, variables, and quantified results for both the original collaborative infrastructure and reported candidates, and the robustness scores and relevant data for 20,000 generated solutions are all stored in relevant log files. This allows for data to be filtered as required and presented in the format of tables and graphs. These findings and experiments will be discussed in the remaining sections and chapters of this thesis.

4.1.3.5 Problems and Limitations

As we have been conducting quantitative analysis to ascertain the effectiveness of our proposed methods against the generated results produced without our simulated framework, we have not required the inclusion of human participants for our research. By analysing the results generated by the automated processes of risk identification and security enhancement, we have omitted the opinions of experts (human participants), therefore we did not have to seek ethical approval for our research. However, we understand the ethical issues associated with research and the principles that would need to be adhered to.

Our research has been restricted by time constraints and cost limits, but in the determination to achieve our goals, simulation was the most viable option and most efficient method to produce diverse topologies and generate appropriate data sets for us to experiment upon and analyse. This ensured that the aims and objectives established were accomplished.

The framework incorporates vulnerabilities from multiple sources in order to simulate adequate network discovery and vulnerability analysis, and robustness function constraints appropriate for our environments, meaning the framework and applied methods are reliant upon standardised vulnerability scoring metrics and databases, and our requirements. Meaning accurate vulnerability scoring, node security grades, and infrastructure communication security scores, can be impacted by the associated issues of external techniques and our expertise in establishing the topologies. This is something we could examine as part of our future work in order to advance further the robustness of collaborative infrastructures and protect against zero-day attacks.

In order to mitigate risk and secure the topology we alter communication paths between components forming the SoS and the SoS within multi-level SoS environments. While we have the capability to quantify centralities which are indicators of topological risks, at this stage we do not have the means to identify the consequences or resulting negative behaviour that could arise. At this stage to prove the appropriateness of the methods, resulting consequences or emergent behaviour was not a major factor, but is noteworthy as an issue to consider in the future.

To analyse the methods and validate the effectiveness of the framework we used simulation and experimented on topologies that were suitable in size allowing for risks to be manually quantified and the results physically analysed. Therefore we require further evaluation against larger multi-level SoS and against physical environments in order to validate the framework's ability to apply assessment methods and security risk mitigation techniques on such environments, and to ensure that results do not just reflect our experiments.

4.1.3.6 Method Summary

This section aims to outline and justify the research methodology that we have implemented as part of this thesis in order to meet our outlined aims and objectives. In the following sections within this chapter we discuss in detail the SeCurity Risk Analysis and Mitigation Framework and the methods applied for data access control management, measurements of topological vulnerabilities, and the security enhancement and risk mitigation techniques. This is followed in Chapters 5 and 6 with an outline of the implementation of these methods and their critical evaluations.

4.2 SeCurity Risk Analysis and Mitigation Framework

Maintaining a balance during a crisis situation, to ensure security does not negatively impede genuine and time critical communication during operations, for example, is essential to protect data and maintain an effective communication network. In previous work, researchers studying network

security and network optimisation have focused upon numerous diverse areas and specific challenges. These include endeavours to optimise and analyse the effectiveness of network security hardening [240], network security situation prediction [241], traffic optimisation [242], optimising task scheduling within distributed real-time systems [243], how to optimally deploy security measures [232], and network path optimisation [244].

Having critically reviewed the limitations of existing research and developments within this thesis, including the challenges and risks which impede and expose SoS and their security, reflecting on how an evolutionary algorithm can be applied to optimise the level of security risk in a SoS and mitigate risks, and by considering a number of critical factors in order to determine the security between interconnected components and systems, which include the likelihood of violating access control requirements, risks associated with high-centrality nodes, and the overall cost of the network in terms of distance between nodes, an appropriate SeCurity Risk Analysis and Mitigation Framework has been devised, which incorporates a number of novel techniques, allowing for us to optimise communication security, mitigate security risks, and improve data flow security for SoS infrastructures without introducing additional resources.

When developing SCRAM we had to envision a framework that would be applied to a broad range of ICT components and technologies, and that the SoS could potentially encompass a range of networked systems which could include Local Area Networks (LAN), WSN, remote networks, IoT, modems, firewalls, and networked systems that could be geographically dispersed. In addition, we perceived that the information gained from the analysis of the SoS must be accurate, as the data will be used to quantify the security and robustness of the entire multi-level SoS infrastructure, thus, will assist in identifying dependencies and risks within the topology, and prove the appropriateness of the reconfigured network infrastructure and its security.

SCRAM has been developed to analyse large, complex, distributed, and dynamic SoS and multi-level SoS infrastructures, identifying risks and vulnerabilities which expose each node within the collaborative environment and that leave networked systems vulnerable to failures and potential attack vectors. Unidentified risks pose a great threat to SoS as both the individual systems and the entire collaborative infrastructure could be prevented from meeting their objectives, i.e. risks can directly impact the integrity and availability of components and systems within the SoS if left unidentified and unregulated. Detected and analysed risks, along with risks associated with important nodes, can then be utilised in part to evaluate the security of the networked systems, and can be incorporated into the evolutionary evolution process to ensure that networks are not negatively mutated during the risk mitigation process.

The proposed SCRAM framework is directly connected to the networked infrastructure in which it is to analyse and measure the security between the interconnected components and systems. The

framework has been developed to be non-intrusive and independent in order to assure that it does not impact the processing resources of the devices it resides upon or monitors, and to assure that it does not introduce additional vulnerabilities into the SoS which could expose the collaborative infrastructure to risk or impede the functionality of the independent systems. It has been developed to overcome the identified challenges and limitations of existing solutions, and achieve the aims and objectives presented in Section 1.3. Figure 4.1 is a representation of an SoS environment with the addition of the SeCurity Risk Analysis and Mitigation Framework.

The proposed SCRAM framework measures the security between interconnected components and systems, achieved by identifying the composition of the SoS and conducting risk analysis and assessment in order to establish vulnerabilities that could expose the SoS and reduce secure communication. These methods are discussed in Section 4.7.

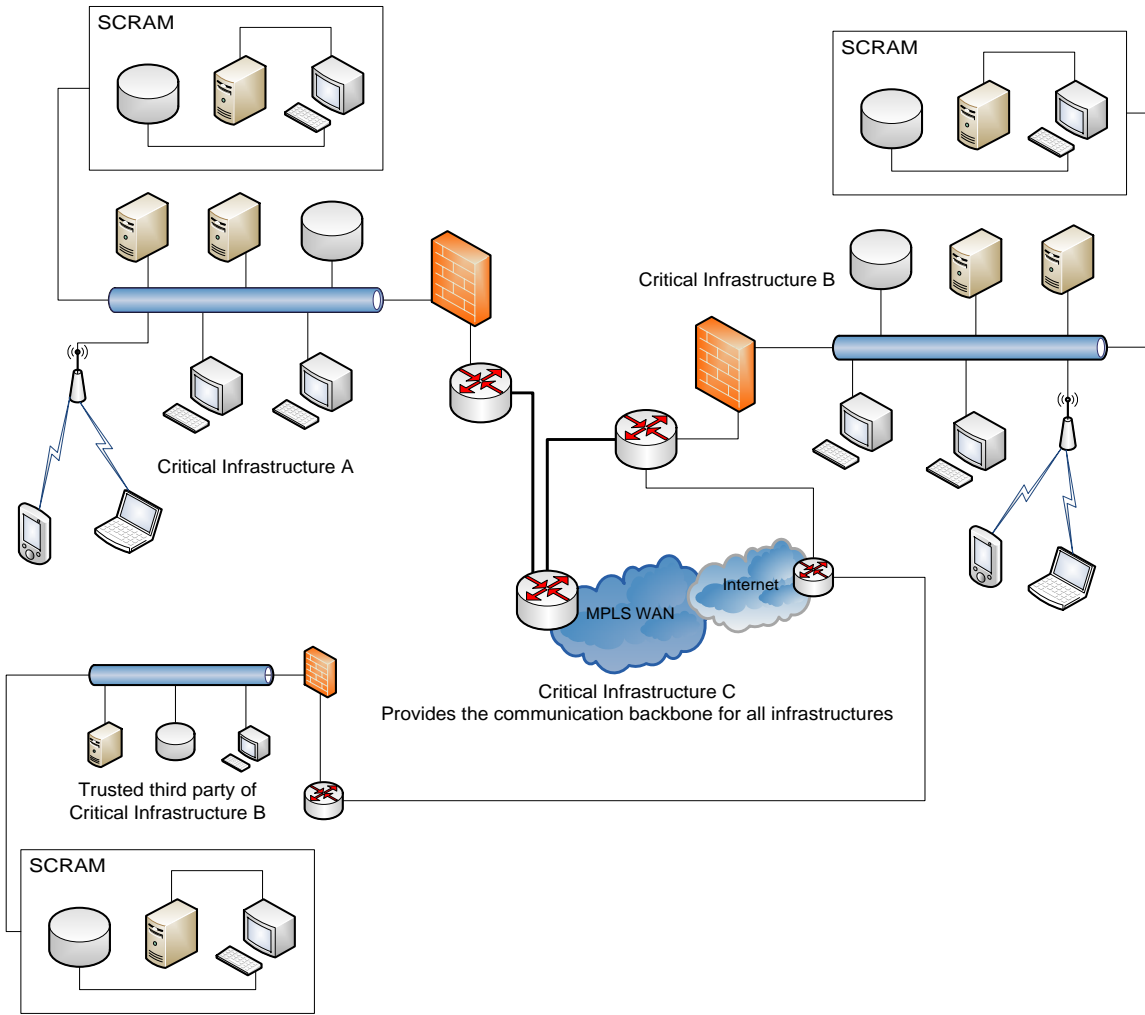


Figure 4.1. SeCurity Risk Analysis and Mitigation Framework Positioning

Identified vulnerabilities with their associated CVSS v3 base scores, and risks associated with a device’s software, hardware, firmware, configuration, and communication connections which have

been assigned the relevant security parameter scores, are utilised by SCRAM to quantify individual node security grades for each networked device. The method for calculating these grades is discussed in Section 4.7.1.4; these generated security grades identify nodes that are secure and insecure, by comparing them against a threshold profile, or identify devices that have not been vulnerability assessed. The grade can be further incorporated within the framework to assist with visualising the node's security status in the undirected graphs, can assist to ascertain the overall communication security level, and is utilised by the proposed robustness function.

The SCRAM framework currently simulates the SoS environments, and can either generate an SoS for analysis or have an existing SoS topology imported into the framework for evaluation. When an SoS is imported or generated within the SCRAM framework, the robustness function quantifies the robustness of the entire SoS topology, by first quantifying the robustness of each node within the SoS based on key parameters generated during the risk analysis process, then generating an overall robustness level for the entire collaborative infrastructure, which represents the appropriateness of the network as described in Section 4.7.3. The robustness level of the network represented in a single parameter means it can be used for decision making processes as a standalone factor, and incorporated into the evolutionary risk mitigation process, as a comparative evaluation number to demonstrate network improvement and evolution as the network is reconfigured into a series of new solutions, in order to mitigate risk and increase SoS security.

In addition, as the SoS are generated or imported, the framework not only performs the initial network discovery and risk assessment, it also visualises each device within the SoS and all communication links between the nodes in an undirected graph. During this import and generation stage, the network and node centralities are quantified by the framework and represent security levels of each system as discussed in Section 4.6, and node data access grades are compared against the data access policy requirements, establishing nodes that are in violation as discussed in Section 4.5. Data access grades and violations are also correspondingly visualised within the generated undirected graph, and these grades will be observed as the network is reconfigured during the risk mitigation process. The risks associated with data access violations will be also represented in the security score of those nodes, as they could introduce additional risks into the SoS by exposing data.

During the risk mitigation process, the SoS will be reconfigured and the framework will re-quantify the network and node centralities, associated costs, communication security, and the robustness level of the SoS for each evolved candidate. The novel evolutionary process will compare the robustness level of reported candidates to ensure that the network does not negatively evolve, and only improved candidates are passed for further evolution, with the best reported individual considered as the optimal reconfigured security solution. This process overcomes many of the limitations associated with local search, and is described in detail in Section 4.7.4.1.

4.3 SeCurity Risk Analysis and Mitigation Framework Design

Overview

In this section we provide a comprehensive overview of the structure and design of the proposed SeCurity Risk Analysis and Mitigation Framework, which is illustrated in Figure 4.2.

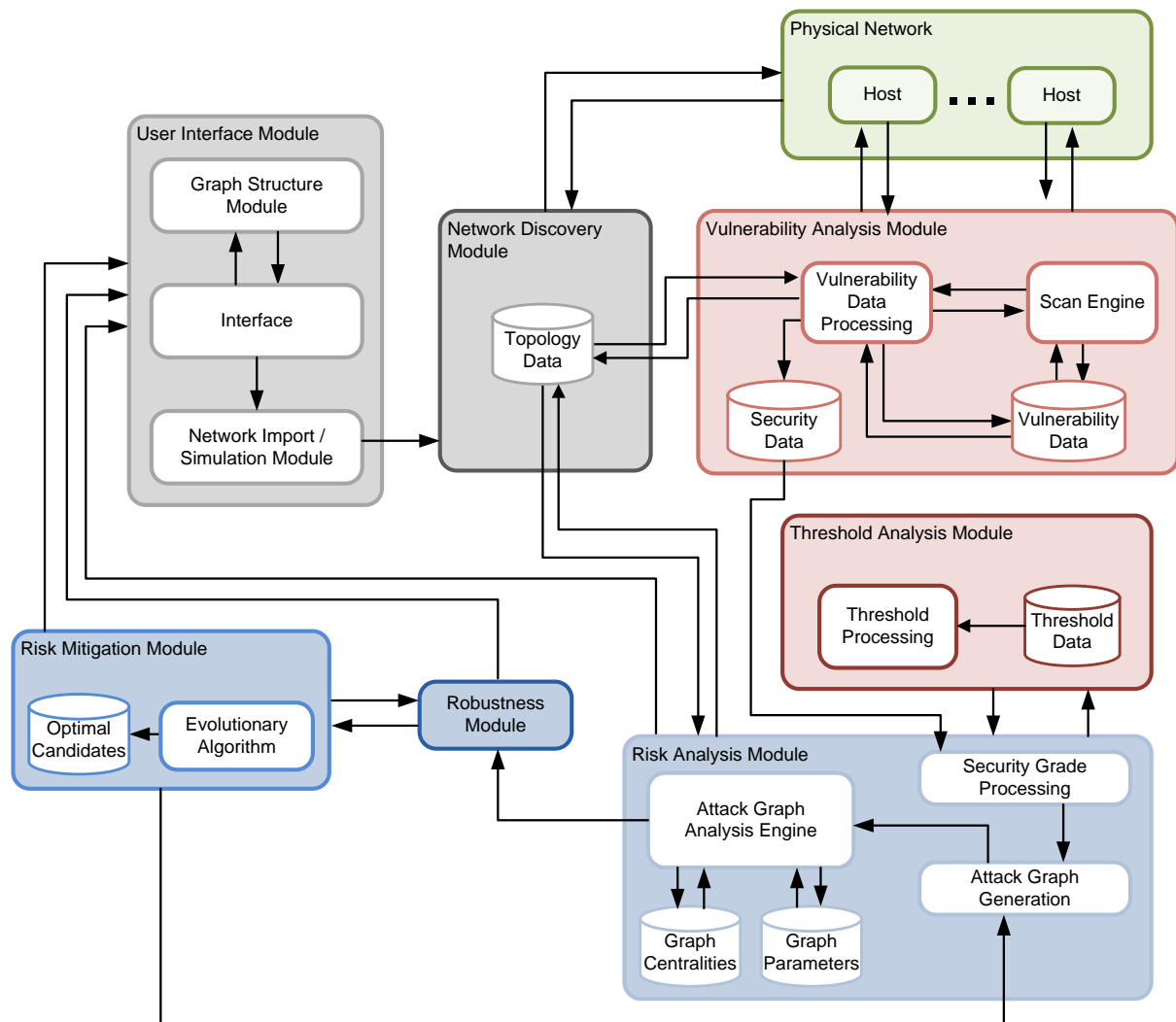


Figure 4.2. Illustrated Overview of the SCRAM Framework

User Interface Module

The *User Interface Module* allows security managers to utilise the *Network Import Module* to either import an existing network into the framework for vulnerability analysis and risk mitigation, or allows for a network to be simulated based on the selected parameters. This can assist with the design and development of ICT infrastructures by simulating networked systems then analysing and reconfiguring the networks to mitigate risks and increase security. The interface allows for a single

SoS infrastructure to be selected and developed for analysis or can initiate multi-level SoS infrastructures for evaluation, via the *Network Import Module*.

The *Graph Structure Module* within the interface allows security managers to select the graph structure type for security optimisation and risk mitigation, including the parameters for prioritisation during the risk mitigation process. For instance, if managers wish to improve security and mitigate risks focusing on the SoS node security grades and robustness, they will not wish to select the graph structure which prioritises and visualises node energy efficiency during the risk mitigation process.

Within the interface window, detailed reports are generated for the SoS and after the collaborative infrastructure has been processed and analysed in regards to risk reduction and security enhancement, the interface window will visualise all generated improved optimal candidates in the form of undirected graphs. In addition, the interface will display detailed reports based on the evolutionary risk mitigation process for all improved optimal networks, including providing details of all node centralities for each evolved candidate.

Network Discovery Module

This *Network Discovery Module* is an automated process that systematically discovers networked devices and assists to map devices identified and their communication links within the *Physical Network* infrastructure, including devices and systems which share a collaborative relationship. Producing a detailed inventory which includes device type, operating system, whether encryption, firewalls, and intrusion detection systems are utilised, if anti-virus and security software is installed on the nodes, if the device has internet access, and the assigned data access for the node, etc. This information is stored within the *Topology Data* database, which can both be accessed by the *Vulnerability Analysis Module* and utilised by the *Risk Analysis Module*. An automated process is essential as manual network mapping would be almost impossible due to the dynamic nature, sheer size, and complexity of the SoS and multi-level SoS environments.

Physical Network

The *Physical Network* infrastructure is the SoS or multi-level SoS that is to be assessed for risks, and if required processed and analysed in regards to risk reduction and security enhancement.

Vulnerability Analysis Module

The *Vulnerability Analysis Module* accesses the *Topology Data* database via the *Vulnerability Data Processing* unit, which is responsible for determining the appropriate vulnerability scans for each node, those that have been identified as unscanned, or if the scan is considered outdated. Once the necessary scans have been conducted utilising the *Scan Engine* unit, *Vulnerability Data* database, and utilising the topology data, the *Vulnerability Data Processing* unit will assess the network's nodes and

evaluate the risks recording the findings and updating information as necessary in the *Vulnerability Data* and *Security Data* databases.

Risk assessment methodologies when applied to networks directly, can impact the functionality of some systems and their components. Therefore, the *Vulnerability Analysis Module* will identify the nodes which are unable to be scanned for vulnerabilities, and the risk that these unscanned nodes pose to the SoS will be quantified as part of the vulnerability analysis.

The vulnerability scoring and exploit databases currently incorporated into the SCRAM frameworks *Vulnerability Analysis Module* are examined in detail in Sections 4.7.1.2 and 4.7.1.3.

Risk Analysis Module

This module serves several purposes; firstly the security data for each node is passed from the *Security Data* database to the *Security Grade Processing* unit. This unit is responsible for quantifying each node's security grade based on the findings of the vulnerability analysis, these grades will then be compared to the relevant thresholds as part of the risk analysis process, and will be utilised as part of the attack graph generation method to assist with visualising node status. Security grade assignment is discussed in detail in Section 4.7.1.

The *Attack Graph Generation* unit within this module utilises the updated topology data stored in the *Topology Data* database, threshold analysis data stored in the *Threshold Data* database, and the quantified security grades to generate an attack graph which will help establish a visualised representation of the network's topology, security status, and data access violations, or can visualise nodes based on energy efficiency levels depending on the graph structure selected. After the evolutionary risk mitigation process has been implemented, the *Attack Graph Generation Module* will also be used to generate the improved optimal candidate graphs.

The *Attack Graph Analysis Engine* evaluates each graph that has been generated, quantifying both network centralities and node centralities, with the results being stored within the *Graph Centralities* database. The importance of network centralities and associated risks are discussed in Section 4.6. In addition, the engine also evaluates other graph parameters such as minimum path average, network communication security, and cost of network communications, these parameters are stored within the *Graph Parameters* database. These parameters are stored as detailed reports that can be accessed via the *User Interface Module*, allowing security managers to analyse the entire SoS in detail.

Threshold Analysis Module

This module contains the *Threshold Data* database, and is primarily used by the *Threshold Processing* unit to identify data access violations and node security status. Thresholds will be established by the network security managers and these profiles will be stored within the *Threshold Data* database. During the risk analysis stage, as security grades are assigned to each node for example, the *Risk*

Analysis Module will pass these grades onto the Threshold Analysis Module for assessment, with results being stored within the *Threshold Data* database. The *Security Grade Processing* unit will then pass on the assessed results to the *Attack Graph Generation* unit which incorporates these results into the graph to ensure that insecure nodes and data access violations can be intuitively identified. The same method is used for storing, quantifying, and visualising node energy efficiency levels.

Robustness Module

This module is responsible for measuring each node within the network by means of a robustness function after an attack graph has been generated and analysed. During the risk mitigation process the *Robustness Module* will quantify the robustness of each node based on five key parameters which have been generated by the *Risk Analysis Module*; this method is described in detail in Section 4.7.3. An overall robustness level is then quantified for the network, and during the risk mitigation process this level assists the evolutionary algorithm to produce a new generation of improved solutions. The robustness score of the network also is of great benefit as it provides an assigned numerical value to the entire network to establish its appropriateness, and can be used as a comparative evaluation number as enhancements are made to the security of the SoS.

Risk Mitigation Module

This module contains an *Evolutionary Algorithm*, to overcome the limitations of local search techniques in large complex networks. Utilising key parameters generated by both the *Risk Analysis Module* and *Robustness Module*, this process generates a new set of potential solutions which are then evolved for comparison to find a set of best solutions, inadequate solutions die out as they are replaced with new better identified solutions. Each solution is fully analysed via the *Risk Analysis Module* and *Robustness Module* to ensure that only the best individuals are directly passed to the next generation of solutions until the end criterion is met. Improved solutions are stored within the *Optimal Candidates* database, and will be passed to the *User Interface Module* to allow for the generated undirected graphs, combined with the reports generated by the *Risk Analysis Module* to be critically assessed by the security managers and decision makers. In Section 4.7.4.1 the evolutionary risk mitigation process is described in detail.

4.4 SeCurity Risk Analysis and Mitigation Framework Operation

In this section we describe in detail the four main process stages of the SCRAM framework that function during its execution. The flowchart in Figure 4.3 visualises these four operational stages during optimal execution only, and does not represent failures occurring during runtime.

Prior to the SCRAM framework being triggered, thresholds will have to be established based on analysis of an organisation's network or default values would be utilised. These constant values will

include security level (i.e. anything below the threshold is considered insecure), highest bridging centrality, centrality degree, minimum path average, and associated network cost in terms of distance between nodes. These values will depend on the importance of the concerned factor and the magnitude, and vary depending on the type of SoS being evaluated or developed.

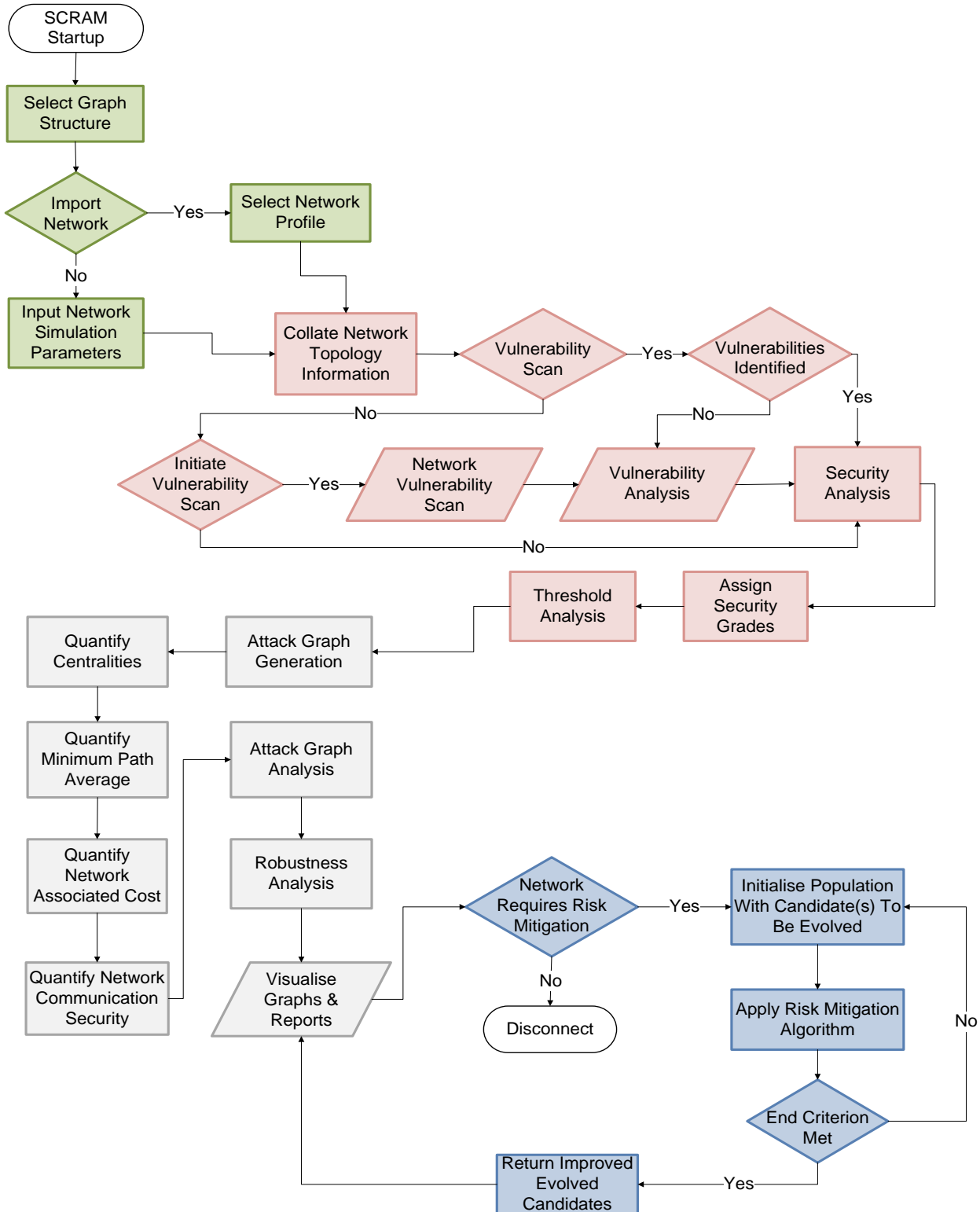


Figure 4.3. SeCurty Risk Analysis and Mitigation Framework Execution Flowchart

While we have programmed SCRAM to simulate several different types of ICT devices, it could be easily extended further based on the requirements of an organisation, prior to the framework being triggered.

Initial Operation Stage (Green)

The initial stage will first require the type of graph structure to be selected, as in this thesis we have implemented two distinct types of monitoring and evaluation, as proof of concept. SCRAM can switch the framework to focus on risk reduction and security enhancement based on security and data access, or can prioritise energy efficiency (i.e. incorporate and focus on different risk vectors) with minimal importance placed on security and data access. Once the graph structure has been established security managers can choose to either import an existing network profile into the framework, or they can input network parameters directly into SCRAM which will assign the relevant attributes to the network which will be simulated and analysed by the system.

Network Discovery Stage (Red)

The relevant network data is collated in order to establish the full SoS environment, including identifying essential communication links between devices and data access grades. It will also establish if nodes within the SoS have recently had a vulnerability scan run against them, require a vulnerability scan to be conducted, cannot be scanned due to the potential negative impact a live scan could cause, or have not been scanned and are outside of the managerial remit so one cannot be conducted. Once the vulnerability scan status has been identified for each node, the relevant vulnerability scans if required can be conducted within the networked environments. In this instance vulnerability scans are simulated within SCRAM against each of the simulated infrastructures.

After all vulnerability scans have been completed, SCRAM conducts a full vulnerability analysis of the results for the entire SoS. Identified vulnerabilities will be assessed and compared against vulnerability databases and assigned risk scores using vulnerability scoring techniques outlined in Sections 4.7.1.2 and 4.7.1.3. Utilising these scores, each node will be graded with an individual security score based on the results from both its identified topology data and the vulnerability analysis results defined in Section 4.7.1.

Each individual node's data access level and security grade will be compared as part of the threshold analysis. This is to identify nodes that should be blocked and not allowed to have unsecure data traverse via the communication paths with other secure nodes, as the node has a lower data access level than the network's assigned level thus violates data access control requirements. Threshold analysis will also identify nodes that have been quantified as insecure, i.e. the assigned score is lower than the permitted minimum security grade assigned and agreed by security managers. This means that critical data will only be allowed to traverse along communication paths via secure nodes that are not in breach of data access policies, with nodes' statuses being documented and reported.

Attack Graph Generation and Analysis (Grey)

The framework will generate an attack graph based upon the analysis of the network's topology, risk assessment, and security analysis. The attack graph will assist to intuitively identify communication links between nodes within the collaborative environment, and will visualise node security status, data access violations, and security grades. Once the graph has been established, SCRAM will quantify the following network centralities (discussed in Section 4.6):

- Degree.
- Betweenness.
- Closeness.
- Eigenvector.
- Bridging.

These centralities are quantified using mathematical formulas, which provide numerical values to assist in the identification of risks, important relationships between nodes, and behavioural characteristics. SCRAM will also then quantify the minimum path average and associated network costs (in terms of the distance between nodes), provide numerical values for these parameters which can help assist in the decision making process when analysing a network, and can reflect both improvements and financial consequences. Finally, SCRAM quantifies the overall network communication security status; this score is a direct reflection of the number of secure routes for encrypted data to traverse.

The undirected graph is further modified to visualise the bridging centrality of each node, achieved by increasing or decreasing node size which is directly correlated to the quantified bridging centrality score. SCRAM will also analyse the attack graph further and produce a series of reports based on the network's topology, risks, and will also include centralities, minimum path average, network cost, and communication security.

The final analysis stage measures each node via the robustness function, this assists in determining the optimal robustness and security level of the network. To quantify the network's robustness the following five criteria are used as part of the robustness function (discussed in Section 4.7.3):

- Communication Security Level.
- Highest Bridging Centrality Score.
- Degree Centrality of the Network.
- Average Minimum Path Length.
- Total Cost (in terms of distance between nodes).

Once completed the undirected graph and reports are sent to the user interface for analysis, prior to any risk mitigation process being applied to the SoS.

Security Risk Mitigation Analysis (Blue)

The framework does not automatically optimise the network's security; this ensures that the original SoS can be reviewed by analysts prior to the collaborative infrastructure's evolution. Instead once the original network has been simulated, analysed, and reported, the process can be discontinued or at this point the security risk mitigation process can be initiated from the user interface. In order to successfully reconfigure the network and determine its optimal security configuration, ensuring that risks are mitigated and the security is enhanced without introducing additional resources into the infrastructure, first the original network is passed into the security risk mitigation process, and then the evolutionary risk mitigation algorithm which is described in detail in Section 4.7.4.1 is applied. Simplified, the security risk mitigation process evolves the network searching for an optimal reconfigured solution, and this evolutionary algorithm measures the appropriateness of each evolved network utilising the robustness function.

If the end criterion is not met, the risk mitigation process is applied to the set of best candidates that are generated via the risk mitigation algorithm. Once the end criterion has been met, the improved evolved candidates are returned to the undirected graph generation process in order for them to be visualised as both graphs and reports in the user interface. At this stage the network could be enhanced again by applying the risk mitigation algorithm if required, or the process can be disconnected.

4.5 Data Access Control Problem and Management

Through surveying the associated literature, we perceive that access safeguards are of vital importance from a security standpoint, and an important challenge that requires further advancement due to the reliance on data transfer during crisis operations, recognising the significance of the data access control problem as surveyed in [237] which outlines a principal model of access control (MATTS), demonstrating data flow between collaborative infrastructures and establishing potential access issues. Using these principal concepts and building upon previous solutions using the MATTS tool to identify such vulnerabilities within crisis management scenarios, we propose a novel solution that identifies and mitigates different types of risks, and enhances both data security and improves data flow security of the overall network.

Smart Cities and crisis operations rely upon the generation and distribution of data, the security of this data and access control is problematic. We use the scenario of a crisis situation occurring within a Smart City to explain the challenges and requirements of access control.

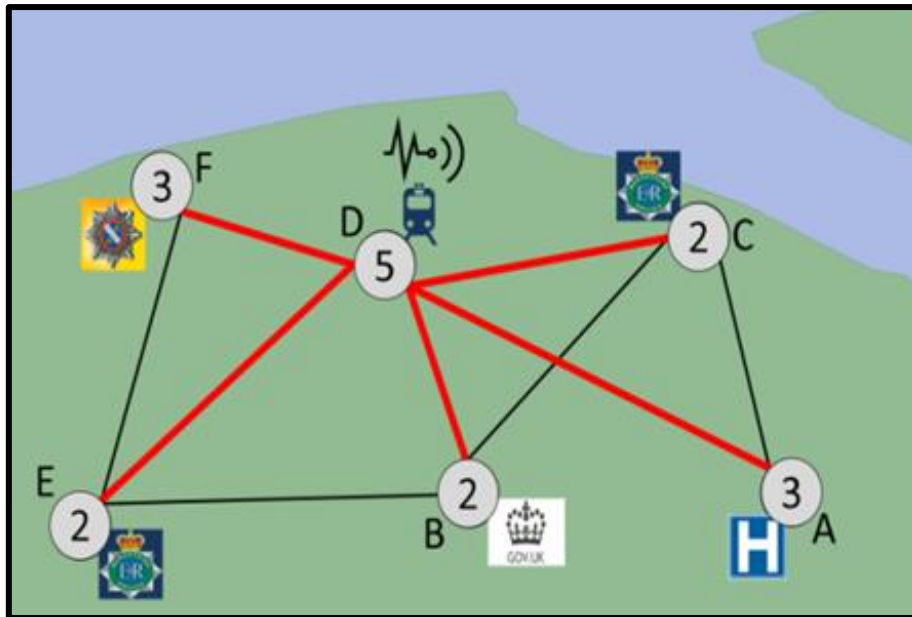


Figure 4.4. Composed Data Access Control Scenario of a Smart City

An easy solution to protect the data would be via the use of traditional cryptographic methods, which secures the data at the cost of increasing computational overhead. As sensors have restricted computational power and memory, these cryptography methods would require heavy adaptation for this environment, and alternative solutions are more desirable [245] [226] [246].

A demonstrative example scenario is provided in Figure 4.4, depicting communications between emergency services (mobile devices), transportation (sensor), and the local government (server) in response to a crisis within a Smart City. Collaborative nodes are connected via varying communication links that include smart devices and a static sensor, with differing security levels. Unencrypted data with a security level of 3 is being forwarded across the collaborative network, between nodes A and F. Figure 4.4 visualises every possible secure and insecure connection in which data with the appropriate security level can traverse between the two nodes, with thick red lines indicating data access violations.

Access control between organisations is demonstrated in Equations 4.1 through to 4.10, where A is the created data set, n is the number organisations, a is the organisation, d is the piece of data (i.e. the data which is to be forwarded), A' is the subset of the created data set A , r is the access rule, D is the piece of data people are entitled to access, P is the people entitled to access data D , m is the mapping of an organisation to its members, u is the user, $\sigma(d)$ is the data sensitivity level, $\alpha(u)$ is the assigned authorisation level for the user, and c represents the complement of the set.

Organisations within the city can be split into sub-units, i.e. the situation involves a set A of n organisations, where $A = \{a_1, a_2, \dots, a_n\}$, for example police operators are represented by more than one element from A . While in general, access control procedures would be operational within organisations, in this crisis scenario, members within each organisation can access all data within their

own organisation. However, should organisations be split into multiple sub-units with individual data scores assigned, it would be possible to model the majority of applied access control procedures [237].

Assume an organisation ai and piece of data d . The organisation ai forms part of a network which is made up from a number of elements from A . Let $A' \subseteq A$ be all organisations downstream from ai for data item d , which is, all these organisations are reachable within the network from ai and permitted to access d [237].

Let

$$r: D \rightarrow P \quad (4.1)$$

represent the access rules that map a piece of data to the people entitled to access it, and

$$m: A \rightarrow P \quad (4.2)$$

represent the mapping of an organisation to its members. Then the general form of the access policy we aim to ensure is that:

$$r(d)^c \cap \bigcup_{a \in A'} m(a) = \emptyset \quad (4.3)$$

where c represents the complement of the set.

Using naïve set theory, this is equivalent to saying that:

$$\bigcup_{a \in A'} m(a) \subseteq r(d). \quad (4.4)$$

People with access to the downstream nodes from ai are within the user set who are entitled to access data d . In distributed environments this requirement can be of high importance [237].

We assume that node A is corresponding with node F, as demonstrated in Figure 4.4, and forwards unencrypted data with a sensitivity level of 3 across the communications network. Conveying our scenario as, every element of data is assigned a sensitivity level between 1 (the most sensitive) and 10 (the least sensitive), with each organisation being assigned an access level from 1 (the greatest access) to 10 (minimum access). Organisation members are authorised to access data of that level or higher. We infer a user u has authorisation level $\alpha(u)$ and an element of data d has a sensitivity level $\sigma(d)$, hence, u can access d if $\alpha(u) \leq \sigma(d)$ [237].

We can derive this scheme from the general case by stating that, for an element of data d has sensitivity level $\sigma(d)$, we define:

$$r(d) = \{p \in P: \sigma(p) \leq \sigma(d)\} \quad (4.5)$$

and for an organisation α with authorisation level $\alpha(a)$ we define:

$$m(a) \subseteq \{p \in P: \alpha(p) \leq \alpha(a)\} \quad (4.6)$$

Given the above definitions, we note that:

$$r(d)^c = \{p \in P: \sigma(p) > \sigma(d)\}, \quad (4.7)$$

and so using these interpretations, our earlier access rule becomes:

$$r(d)^c \cap \bigcup_{a \in A'} m(a) = \emptyset \quad (4.8)$$

which we can guarantee if:

$$\{p \in P: \sigma(p) > \sigma(d)\} \cap \bigcup_{a \in A'} \{p \in P: \alpha(p) \leq \alpha(a)\} = \emptyset, \quad (4.9)$$

which is equivalent to saying that:

$$\alpha(a) \leq \sigma(d) \text{ for all } a \in A'. \quad (4.10)$$

As a result, each downstream node's authorisation level is less than or equal to the sensitivity level of the data d (i.e. the low number represents higher access).

We must also consider that Smart Cities typically contain a series of sensor networks and IoT. When we can overview the topology of such collaborative networks we see that they are a combination of diverse devices that sense data or control and interact with other systems and objects. This type of topology could form a complex series of differing communication links across a city, with devices connecting and transferring different types of data, in a variety of formats, and via various protocols. The scenario in Figure 4.5 is a demonstrative example of such devices and connections. Each device that is connected to the IoT and sensor network within the city has different security grades based on the level of risk it poses to the network, and each device has an assigned data access level. Devices

that can make up such a varied network can include cars, CCTV, transportation, telephones, weather sensors, and smart devices for example.

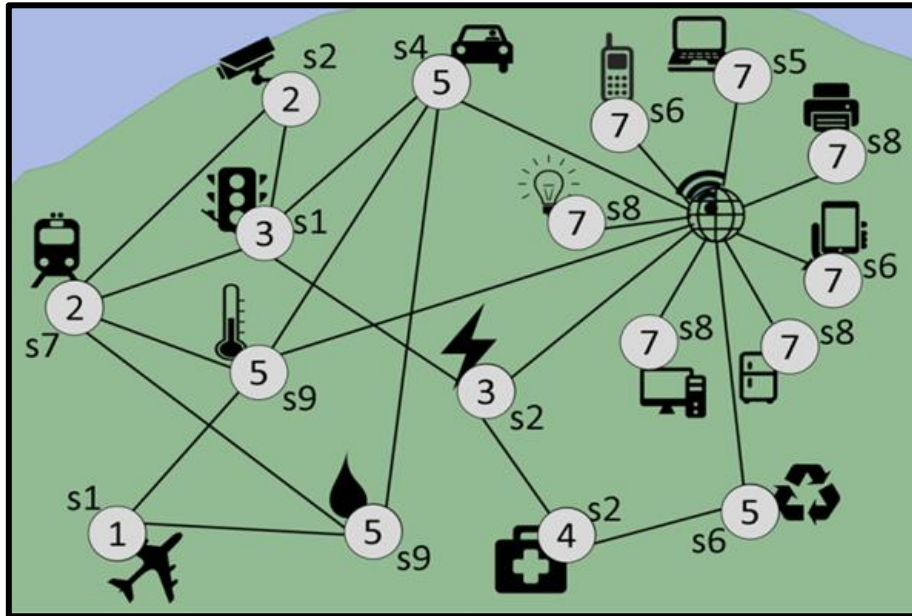


Figure 4.5. Composed Smart City Consisting of Sensitivity Levels and Data Flow Risk

The motivation behind the development of IoT is its potential to connect diverse ‘things’ together, meaning that devices can be accessed or controlled remotely across the communication infrastructure. In this scenario we cannot simply block and reroute data via different communication paths to devices, as the objects that they are connecting with are meant to be accessed or controlled, i.e. devices with lower security and data access levels may be required to interact with devices that have higher security and data access levels within the city.

4.6 Topological Vulnerabilities

In addition to the data access control problem we also consider the problematic relational states between the nodes, in an attempt to identify vulnerabilities and critical risks which have the ability to expose the collaborating systems. Realised through the use of mathematical formulas and the assignment of numeric numbers to risks, which allows for risks to be quantified and network topologies to be visualised. With advancement in the fields of graph theory, network theory, and social network analysis, there has been considerable progress with mathematical and computational tools. This allows for important relationships between nodes to be conveyed, and can assist with ascertaining network behaviour characteristics. For instance, centrality indicators (degree, betweenness, closeness, eigenvector, and bridging) help to assist us with ascertaining a node’s (vertices) importance within a network and identify vulnerabilities associated with connectivity [247] [248].

4.6.1 High Connectivity Vulnerability

Nodes that are highly connected, if attacked or disconnected from the network, can leave low connected nodes isolated and cut off from the remainder of the network and can reduce the number of secure routes available for data transfer. In addition, the removal of highly connected nodes within a Smart City environment could disconnect and split the networked infrastructure into isolated networks and prevent the SoS from meeting objectives, and reduce the number of paths data can traverse. In the worst case scenario, this vulnerability could cause major disruptions and cascade failures as the infrastructure fails to communicate and meet objectives. High connectivity can be identified by the quantification of a node's degree centrality.

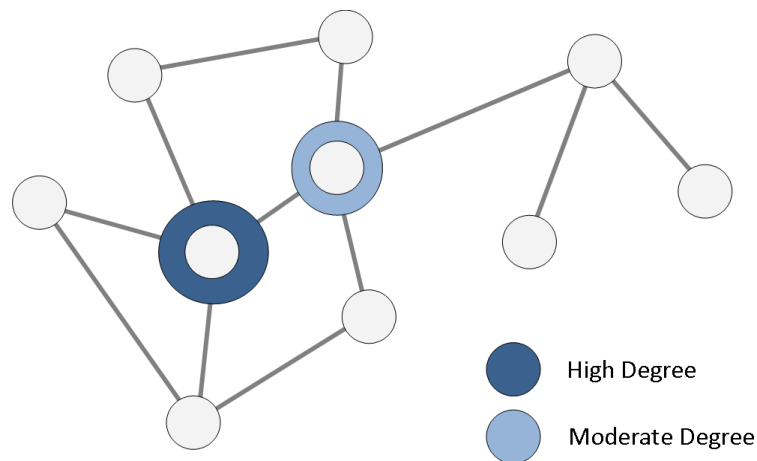


Figure 4.6. Schematic Representation of Degree Centrality

Degree centrality assists with the identification of how popular or active a node is within a network, for example, a high degree value indicates the node's dominance within the network. This is accomplished by quantifying the number of connections (edges) each node has, for a node u is the ratio of the number of incoming edges of node u to the total number of all other nodes in the network [247] [248]:

$$C_{deg}(u) = \frac{deg(u)}{|V| - 1} \quad (4.11)$$

where $deg(u)$ is the number of node u 's edges and V is the set of nodes in the network [247].

4.6.2 Shortest Path Vulnerability

Nodes that are centrally located based on the high number of shortest paths that pass through the node, are more influential than other nodes. Should a Smart City device located on a high degree of shortest paths be attacked or disconnected from the network, then these influential nodes can cause failures to occur across the entire city due to the node being responsible for the transfer of essential data across the networked infrastructure. Failings could cause both minor and major disruptions to the Smart City's services and assets, and could result in economic loss both locally and globally, etc. A node's degree of shortest paths can be identified by the quantification of a node's betweenness centrality.

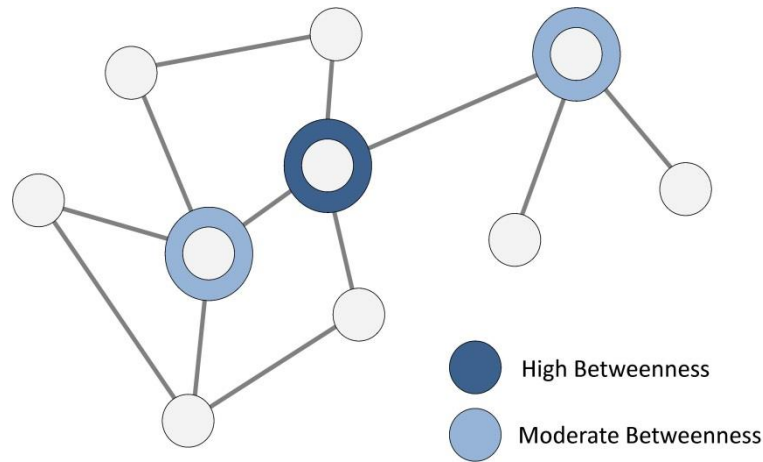


Figure 4.7. Schematic Representation of Betweenness Centrality

Betweenness centrality conveys a node's centrality; the nodes situated on the shortest path route are in a unique position within the network as they are often the nodes most relied upon to transfer data as it navigates across the network. A higher betweenness value indicates the node's importance in regards to data flow, yet also determines its potential to be a single point of failure within the environment. Accomplished by quantifying the number of shortest path connections that traverse through a node, for a node u is the proportional number of shortest paths between all node pairs in the network that pass through u [247] [248]:

$$C_{bet}(u) = \frac{1}{(|V| - 1) \cdot (|V| - 2)} \sum_{s \neq u, t \neq u \in V} \frac{\sigma_{s,t}(u)}{\sigma_{s,t}} \quad (4.12)$$

where $\sigma_{s,t}$ is the total number of shortest paths from source node s to destination node t , and $\sigma_{s,t}(u)$ is the number of shortest paths from source node s to destination node t which actually pass through node u [247].

4.6.3 Single Points of Failure

Nodes that are potential SPoF, if attacked or disconnected from the network, can prevent data from traversing across the networked systems, causing large sections of the network to become fragmented or could incapacitate an entire infrastructure. In addition, SPoF can cause partial or full cascading failures to quickly ripple across the infrastructure. Should SPoF occur within a Smart City, then critical failings could occur across the city, directly resulting in economic loss, damage to systems, prevent access to vital services provided by the city's assets, or could result in the loss of human life, etc. It is essential to identify these interdependent nodes to ensure Smart Cities remain robust. Nodes at risk of becoming SPoF can be identified by the quantification of a node's closeness centrality.

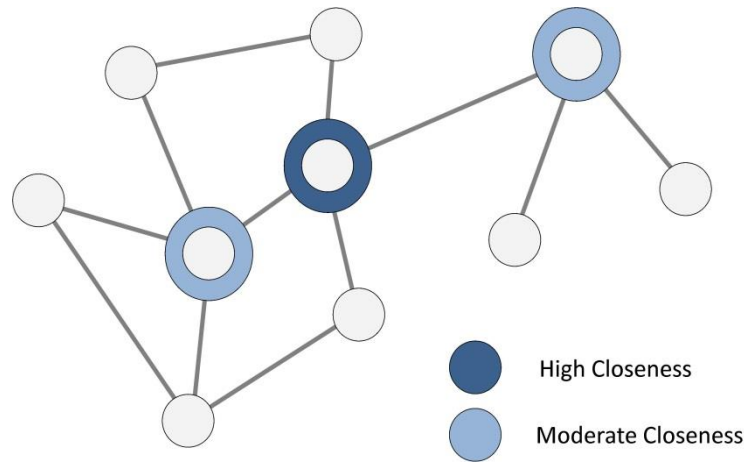


Figure 4.8. Schematic Representation of Closeness Centrality

Closeness centrality assists with the identification of nodes with the shortest paths, and those nodes which are uniquely situated with access to all nodes within the structure either directly or indirectly. Highly centralised networks are in general greatly unstable, should node failures or disconnections occur then the network can quickly become fragmented and failure could ensue. However, low centralised networks in general are not prone to SPoF, meaning in the event of node disablement the network tends to remain functional via the use of alternative edges. This centrality is accomplished by quantifying the node's distance to all other nodes, for a node u is the average inverse shortest path length to all other nodes in the network [247] [248]:

$$C_{clo}(u) = \frac{[\sum_{v \neq u \in V} dist(u, v)]^{-1}}{|V| - 1} \quad (4.13)$$

where $dist(u, v)$ is the length of the shortest path from node u to node v [247].

4.6.4 Weighted High Connectivity

Nodes that are highly connected to other highly connected nodes within the network have an increased influence over the entire infrastructure. High connectivity also increases complexity and introduces additional vulnerabilities to the node increasing associated risk. These nodes if disconnected or attacked within a Smart City environment would quickly result in the network becoming fragmented, and prevent the transfer of communication across the city, resulting in both direct and indirect system failures. SPoF could also arise, with the potential for cascading failures to ripple across the city. Weighted high connectivity can be identified by quantifying a node's eigenvector centrality.

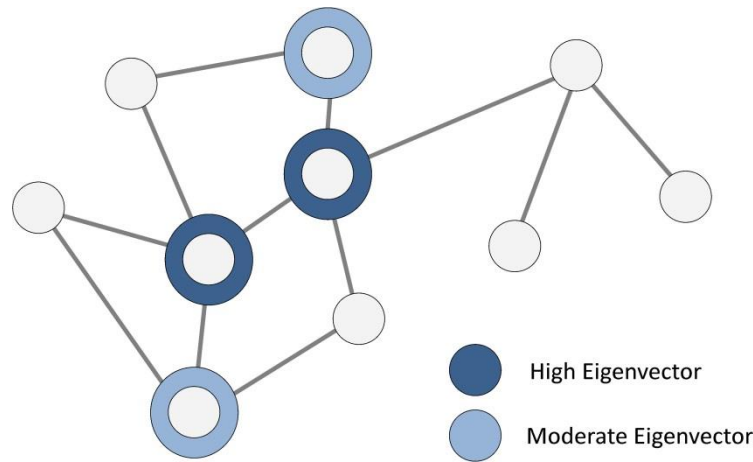


Figure 4.9. Schematic Representation of Eigenvector Centrality

Eigenvector centrality assists with the identification of those nodes which play a more prominent role within the network. This type of centrality is considered to be more advanced than degree centrality, as it differentiates links that are not equal to each other. Eigenvector centrality is accomplished by quantifying and assigning values to nodes based not only on the number of links but also if those links establish a connection to other prominent nodes within a network, for the prominence of a node u is understood to be proportional to the combined prominence of its neighbours [247] [248]:

$$C_{eig}(u) = \frac{1}{\lambda} \sum_{v \in N(u)} W_{u,v} \cdot C_{eig}(v) \quad (4.14)$$

where $N(u)$ is the set of nodes reachable directly from u and λ is a constant. With vector-matrix notation, this equation can be rewritten as $\lambda \cdot C_{eig} = W \cdot C_{eig}$ where $C_{eig} = (C_{eig}(v))_{v \in V}$ and $W = (W_{u,v})_{u,v \in V}$. Therefore C_{eig} is an eigenvector of the weighted adjacency matrix W with eigenvalue λ [247].

4.6.5 Dependent Communication Vulnerability

Nodes that are depended upon to maintain communications within a network, if attacked or disconnected can fragment the network and leave nodes isolated. The removal or failure of nodes which are highly depended upon to maintain communications within a Smart City can prevent objectives from being met as critical data cannot be accessed or transferred. Any interruption to critical data transfer or creation could prevent SoS collaboration, and result in SPoF and cascading failures. Nodes which are highly depended upon can be identified by the quantification of a node's bridging centrality.

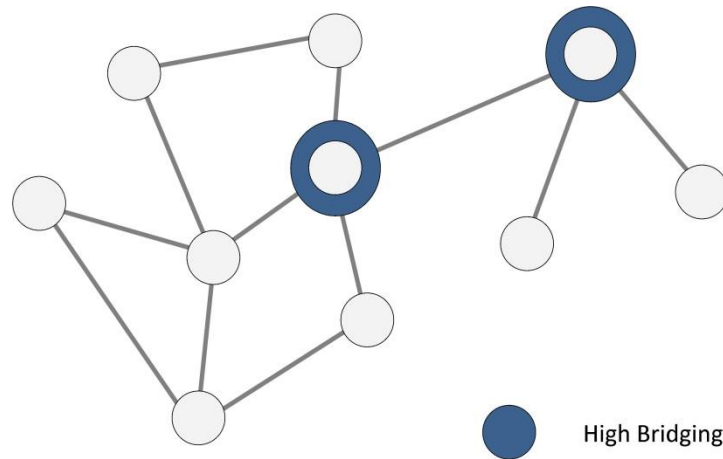


Figure 4.10. Schematic Representation of Bridging Centrality

Bridging centrality conveys whether a node is densely connecting other nodes within a network, and whether the node's topological location and data flow are relied upon by those various connected regions, i.e. identifies nodes within a graph which are positioned between regions, and which are most critical and could interrupt data flow within the network in the event of failure. Identifying nodes with high bridging centrality can assist with network protection and help administrators to improve the overall robustness of the network. Bridging centrality is accomplished by quantifying the network's betweenness centrality C_B and the bridging coefficient BC , thus measures a node's global and local features. The bridging centrality $C_R(v)$ for v of interest is defined by [249]:

$$C_R(v) = BC(v) \times C_B(v) \quad (4.15)$$

4.7 Security Enhancement and Risk Mitigation Techniques

We consider high risk nodes in the network throughout the security enhancement and risk mitigation process, focusing upon nodes with a high degree of connectivity, i.e. nodes measured through bridging centrality. Nodes with high bridging centrality pose a greater threat to networks, as should these nodes be compromised or a failure occur, the impact caused to these critical points has the

capacity to interrupt data flow. To minimise these risks, we mitigate risk and increase security by reconfiguring the network connectivity, achieved by changing connections among the nodes in order to determine the most secure combination of links. In addition to security factors (degree centrality, bridging centrality, and communication security level), we examine two natural factors during the risk mitigation process. These are the average minimum path length, which takes the average of all shortest paths between pairs of nodes within the network, and the cost of communications. This is the sum of all link weights, and is calculated as the geodesic distance between connected nodes.

4.7.1 Node Security Grade Assignment

In order to overcome the limitations of existing solutions it is vital that the SCRAM framework can efficiently quantify an accurate security grade for all nodes within the SoS. This section presents a detailed insight into SoS vulnerability analysis, CVSS scoring system, and NVD vulnerability database, which have advanced the functionality of the framework. Then finally outlines the method for measuring and assigning node security grades.

4.7.1.1 Vulnerability Analysis

Typically, vulnerabilities would be initially identified using a network vulnerability scanner, which allows hosts to be scanned along with the topology of the network. These tools identify and provide specifics on vulnerabilities within the network's topology and hosts, generating details on weaknesses such as open ports, network configurations, system components, operating systems, software applications and services, log-ons, and active IP addresses, etc.

They can also assist in prioritising the implementation of solutions, and have the capability to detect malicious services such as Trojans. Vulnerability scanners though, must be used as part of a risk assessment strategy and not as a full standalone security solution, as they can struggle to identify vulnerabilities resulting in false positives. Unlike firewalls, anti-virus, and intrusion detection systems, vulnerability scanners provide a proactive approach to ICT security rather than purely endeavouring to defend against attacks. Providing an automated platform that identifies vulnerabilities and analysis of network states [250], popular scanners that can be utilised include Nessus [93], Retina [94], Nmap [95], Nspose [97], and MaxPatrol [96].

In addition to network vulnerability scanners, vulnerability scoring and exploit databases can also be incorporated into the risk assessment strategy for the identification and quantification of vulnerabilities. The Common Vulnerability Scoring System (CVSS) [251], National Vulnerability Database (NVD) [92], Common Vulnerabilities and Exposures (CVE) [103], SecurityFocus Forum

[104], Open Source Vulnerability Database (OSVDB) [105], and Bugtraq Security Database [104], for example, have been developed over the last couple of decades to identify and measure vulnerabilities in a variety of ways with differing focuses. Some of the schemas provide threat warning systems, whereas others provide vulnerability databases, while several vulnerability scoring methods assist directly with vulnerability identification [250].

CVSS has heavily influenced our research and the development and implementation of our SCRAM framework, as the algorithms within the methodology [251] have been widely incorporated into many vulnerability applications as they have the capacity to assist with assigning numerical values to risks and vulnerabilities. Scores are composed based on three metric groups (base, temporal, and environmental), and are summarised in Section 4.7.1.2 below. We also incorporate the principles of NVD [92] into SCRAM, as it supports the automation of vulnerability management and security. NVD is an open repository of vulnerabilities, and is summarised in Section 4.7.1.3.

4.7.1.2 Common Vulnerability Scoring System (CVSS)

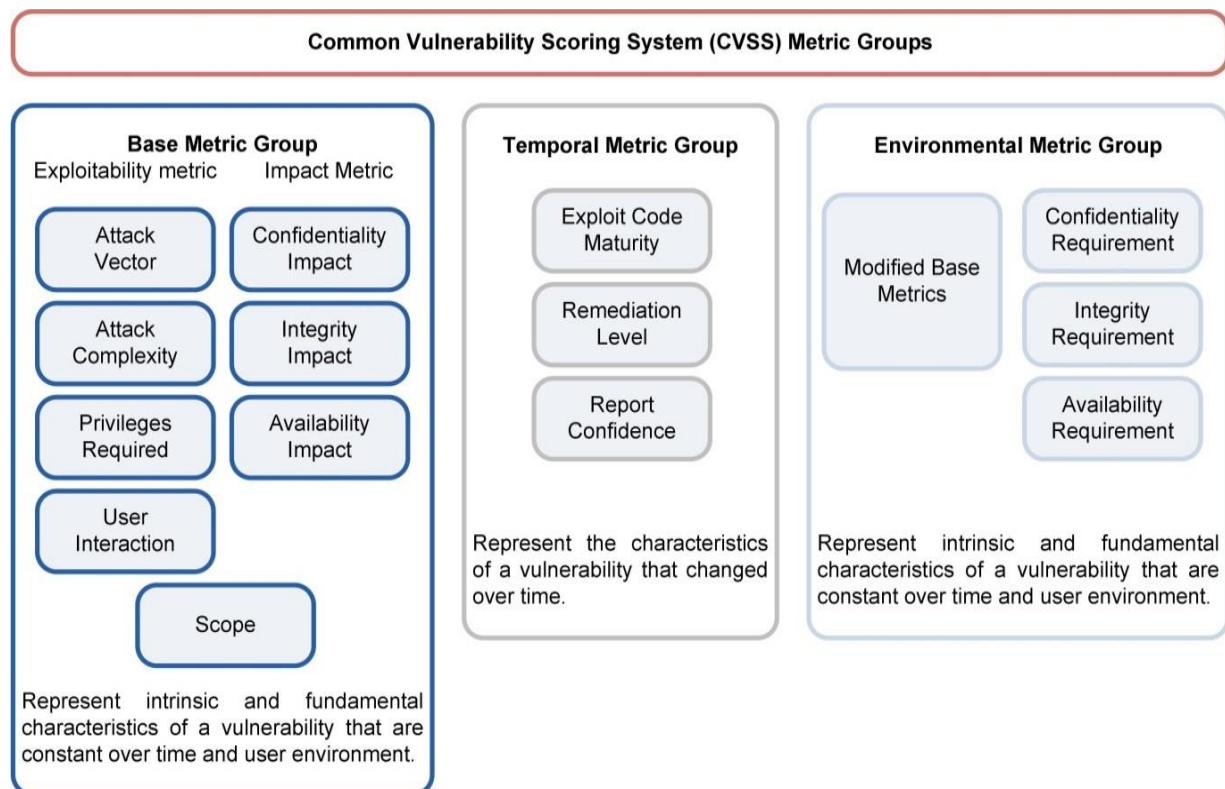


Figure 4.11. Overview of the CVSS v3.0 Metric Groups

Source: Schematic representation of CVSS v3.0 Specification [251].

The Common Vulnerability Scoring System (CVSS) provides standardised vulnerability scoring, an open framework, and contextual scoring, as shown in Figure 4.11. Scores are composed based on three metric groups (base, temporal and environmental). Providing a platform that assigns risk in a

standardised manner, including a schema that has the functionality to accommodate industry specifics [251].

The base equation score is considered the foundation of the scoring schema. Once a base metric has been assigned to a distinct vulnerability, the base equation will calculate a risk score ranging from 0 to 10. Base scores can be advanced by assigning values to the temporal and/or environmental metrics, providing a more accurate score for the vulnerability within its environment. This is not essential as the schema can still quantify the base score vector [251] defined as,

$$\begin{aligned} & \text{If (Impact sub score} \leq 0 \text{ else, } 0) \\ & \text{Scope Unchanged} \quad \text{Roundup}(\text{Minimum}[(\text{Impact} = \text{Exploitability}), 10]) \end{aligned} \quad (4.16)$$

And the Impact sub score (ICS) defined as,

$$\begin{aligned} & \text{Scope Unchanged} \quad 6.42 \times \text{ISC}_{\text{Base}} \\ & \text{Scope Changed} \quad 7.52 \times [\text{ISC}_{\text{Base}} - 0.029] - 3.25 \times [\text{ISC}_{\text{Base}} - 0.02]^{15} \end{aligned} \quad (4.17)$$

Where,

$$\text{ISC}_{\text{Base}} = 1 - [(1 - \text{Impact}_{\text{Conf}}) \times (1 - \text{Impact}_{\text{Integ}}) \times (1 - \text{Impact}_{\text{Avail}})] \quad (4.18)$$

And the Exploitability sub score is,

$$\begin{aligned} & 8.22 \times \text{AttackVector} \times \text{AttackComplexity} \times \text{PrivilegeRequired} \times \\ & \text{UserInteraction} \end{aligned} \quad (4.19)$$

Temporal score is defined as,

$$\begin{aligned} & (\text{BaseScore} \times \text{ExploitCodeMaturity} \times \text{RemediationLevel} \times \\ & \text{ReportConfidence}) \end{aligned} \quad (4.20)$$

The environmental score is defined as,

$$\begin{aligned} & \text{If (Modified Impact} \leq 0 \text{ else,} \\ & \text{Sub score} \leq 0) \end{aligned} \quad (4.21)$$

$$\begin{aligned} & \text{If Modified Scope is} \quad \text{Round up}(\text{Round up}(\text{Minimum} [\\ & \text{Unchanged} \quad (\text{M.Impact} + \text{M.Exploitability}), 10]) \\ & \quad \times \text{Exploit Code Maturity} \end{aligned} \quad (4.22)$$

$$\begin{aligned}
 & x \text{ Remediation Level} \\
 & x \text{ Report Confidence)} \\
 \text{If Modified Scope is} & \text{ Round up(Round up(Minimum [1.08} \\
 \text{Changed} & x (M.Impact + M.Exploitability) , 10]} \\
 & x \text{ Exploit Code Maturity} \\
 & x \text{ Remediation Level} \\
 & x \text{ Report Confidence)}
 \end{aligned} \tag{4.23}$$

And the modified Impact sub score is defined as,

$$\begin{aligned}
 \text{If Modified Scope is} & \quad 6.42 \times [ISC_{\text{Modified}}] \\
 \text{Unchanged} & \tag{4.24}
 \end{aligned}$$

$$\begin{aligned}
 \text{If Modified Scope is} & \quad 7.52 \times [ISC_{\text{Modified}} - 0.029] - 3.25 \times [ISC_{\text{Modified}} - 0.02]15 \\
 \text{Changed} & \tag{4.25}
 \end{aligned}$$

Where,

$$\begin{aligned}
 ISC_{\text{Modified}} = \text{Minimum} & \quad [[1 - (1 - M. I_{\text{Conf}} \times CR) \times (1 - M. I_{\text{Integ}} \times IR) \times \\
 & (1 - M. I_{\text{Avail}} \times AR)], 0.915]
 \end{aligned} \tag{4.26}$$

The Modified Exploitability sub score is,

$$\begin{aligned}
 & 8.22 \times M. \text{AttackVector} \times M. \text{AttackComplexity} \times M. \text{PrivilegeRequired} \\
 & \quad \times M. \text{UserInteraction}
 \end{aligned} \tag{4.27}$$

4.7.1.3 National Vulnerability Database (NVD)

NVD is an open repository of vulnerabilities, including essential details in regards to security-related software flaws, security check lists, impact metrics, product names, and misconfigurations. This database is also reliant upon the CVE repository; nonetheless, NVD augments additional analysis and thus can be considered its superior. While NVD is synchronised to automatically update when new vulnerabilities are identified and published by CVE, it cannot be categorised as a real-time vulnerability and reporting mechanism. This is due to the fact that NVD analysts can take as long as two full working days to analyse the vulnerabilities and augment the vulnerability attributes [92] [250].

Table 4.1 National Vulnerability Database Scoring Methodology Overview

Stage	Process
NVD receives vulnerability information via CVE.	<ul style="list-style-type: none"> • CVE dictionary feeds include: <ol style="list-style-type: none"> a) The unique CVE identifier. b) Description of the vulnerability. c) Links to websites and other references (related to vulnerability).
NVD vulnerability analysts process the information.	<ul style="list-style-type: none"> • Link availability and applicability (verify link publically available and related to vulnerability). • Link verification contains specific data relating to one of the following: <ol style="list-style-type: none"> a) US government resource. b) Advisory notice or bulletin. c) Patch or update for vulnerability. d) Proof of concept or exploit. • CWE identification (determine if the vulnerability description associates with a CWE weakness). • Assign CVSS metrics (assign CVSS base metric values, using scoring templates to ensure consistency among vulnerability analysts).

Source: NVD.nist.gov [92] and CVE.mitre.org [103].

Via third-parties, data is collected in regards to issues; attributes are then assigned to a specific vulnerability. The CVE website provides details in regards to analysis of the vulnerability, and then CVSS metrics are applied to determine the vulnerabilities impact metrics. Common Weakness Enumeration (CWE) determines software weaknesses in a unified measurable format, and Common Platform Enumeration (CPE) ensures applicable statements are included, along with other relevant metadata.

A summary of the scoring methodology used by NVD to identify, analyse, and score vulnerabilities is outlined in Table 4.1. Once vulnerabilities are assigned metrics, they are made available to organisations via XML Data Feeds, with no restrictions placed upon its use [92] [250]. Table 4.2 provides an example summary of vulnerability CVE-2016-7211 which was identified by CVE and has been assigned impact metrics by NVD [92]. Using NVD, these vulnerabilities can be scanned for and identified within the simulation, then quantified into the solution's security grade assignment, thus improve network security and network reconfiguration.

When we review the CVSS severity scores in Table 4.2, we note a marginal difference between the scores, with CVSS version 3 scoring vulnerability CVE-2016-7211 0.1 higher than version 2. This was a notable factor that influenced the development of the SCRAM framework, choosing to utilise CVSS v3 scores over its predecessor. As CVSS v3 has increased the accuracy of its scoring technique by incorporating additional solutions, and having incorporated new metrics which are Scope and User Interaction within the Base Metric group. Additionally, the Authentication metric was replaced with Privileges Required. The Environmental Metrics group was also updated with a new Modified Base

Metrics which supports analysts to customise host scores that have been affected within the organisation. These Metrics and Metrics Groups are visualised in Figure 4.11.

Table 4.2. Identified Vulnerability CVE-2016-7211 Entry

Vulnerability	Overview																																																																																				
CVE-2016-7211	The kernel-mode drivers in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold, 1511, and 1607 allow local users to gain privileges via a crafted application, aka "Win32k Elevation of Privilege Vulnerability." a different vulnerability than CVE-2016-3266, CVE-2016-3376, and CVE-2016-7185.																																																																																				
	<table border="1"> <thead> <tr> <th colspan="2">Impact</th> <th colspan="2">CVSS Severity (version 3.0):</th> <th colspan="2">CVSS Severity (version 2.0):</th> </tr> </thead> <tbody> <tr> <td>CVSS v3 Base Score:</td> <td>7.3 High</td> <td>CVSS v2 Base Score:</td> <td>7.2 HIGH</td> <td></td> <td></td> </tr> <tr> <td>Vector:</td> <td>CVSS:3.0/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H</td> <td>Vector:</td> <td>(AV:L/AC:L/Au:N/C:C/I:C/A:C)</td> <td></td> <td>(legend)</td> </tr> <tr> <td>Impact Score:</td> <td>5.9</td> <td>Impact Subscore:</td> <td>10.0</td> <td></td> <td></td> </tr> <tr> <td>Exploitability Score:</td> <td>1.3</td> <td>Exploitability Score:</td> <td>3.9</td> <td></td> <td></td> </tr> <tr> <td>CVSS Version 3 Metrics:</td> <td></td> <td>CVSS Version 2 Metrics:</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Attack Vector (AV):</td> <td>Local</td> <td>Access Vector:</td> <td>Locally exploitable</td> <td></td> <td></td> </tr> <tr> <td>Attack Complexity (AC):</td> <td>Low</td> <td>Access Complexity:</td> <td>Low</td> <td></td> <td></td> </tr> <tr> <td>Privileges Required (PR):</td> <td>Low</td> <td>Authentication:</td> <td>Not required to exploit</td> <td></td> <td></td> </tr> <tr> <td>User Interaction (UI):</td> <td>Required</td> <td>Impact Type:</td> <td>Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service</td> <td></td> <td></td> </tr> <tr> <td>Scope (S):</td> <td>Unchanged</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Confidentiality (C):</td> <td>High</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Integrity (I):</td> <td>High</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Availability (A):</td> <td>High</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Impact		CVSS Severity (version 3.0):		CVSS Severity (version 2.0):		CVSS v3 Base Score:	7.3 High	CVSS v2 Base Score:	7.2 HIGH			Vector:	CVSS:3.0/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H	Vector:	(AV:L/AC:L/Au:N/C:C/I:C/A:C)		(legend)	Impact Score:	5.9	Impact Subscore:	10.0			Exploitability Score:	1.3	Exploitability Score:	3.9			CVSS Version 3 Metrics:		CVSS Version 2 Metrics:				Attack Vector (AV):	Local	Access Vector:	Locally exploitable			Attack Complexity (AC):	Low	Access Complexity:	Low			Privileges Required (PR):	Low	Authentication:	Not required to exploit			User Interaction (UI):	Required	Impact Type:	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service			Scope (S):	Unchanged					Confidentiality (C):	High					Integrity (I):	High					Availability (A):	High				
Impact		CVSS Severity (version 3.0):		CVSS Severity (version 2.0):																																																																																	
CVSS v3 Base Score:	7.3 High	CVSS v2 Base Score:	7.2 HIGH																																																																																		
Vector:	CVSS:3.0/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H	Vector:	(AV:L/AC:L/Au:N/C:C/I:C/A:C)		(legend)																																																																																
Impact Score:	5.9	Impact Subscore:	10.0																																																																																		
Exploitability Score:	1.3	Exploitability Score:	3.9																																																																																		
CVSS Version 3 Metrics:		CVSS Version 2 Metrics:																																																																																			
Attack Vector (AV):	Local	Access Vector:	Locally exploitable																																																																																		
Attack Complexity (AC):	Low	Access Complexity:	Low																																																																																		
Privileges Required (PR):	Low	Authentication:	Not required to exploit																																																																																		
User Interaction (UI):	Required	Impact Type:	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service																																																																																		
Scope (S):	Unchanged																																																																																				
Confidentiality (C):	High																																																																																				
Integrity (I):	High																																																																																				
Availability (A):	High																																																																																				

Source: NVD.nist.gov [92].

4.7.1.4 Calculating Node Security Grades

Using the metrics within CVSS along with scoring systems and vulnerability data bases, in addition to the data security level and energy efficiency grade, we can quantify a security grade for each node based upon the node's individual software, hardware, and firmware. Security is graded on a scale of 0 to 10, with a security grade of 0 being considered the most secure and 10 least secure. Data types

retrieved for each node include firewall status, intrusion detection system status, encryption status and if used type, operating system, staff skill level, system update status, anti-virus/security, internet access, data security level, etc., with each data type being assigned its related risk probability score.

Table 4.3. Example Parameters and Their Associated Risk Probability Scores

Risk Type	Risk Probability Score									
	Low Risk							High Risk		
	1	2	3	4	5	6	7	8	9	10
Firewall Status	True									False
IDS	True									False
Encryption Status	True									False
Encryption Type	AES - 256	TDES - 168		RC2 - 128			WEP - 114			
Operating System	Linux	Mac OS X			Windows Server 2000	Windows 8	Windows XP		HP-UX11i	Solaris
Staff Skill Level	High				Medium				Low	
System Updated	True									False
Anti-Virus/Security	True									False
Internet Access	False									True
Data Security Level	1	2	3	4	5	6	7	8	9	10

Table 4.3 demonstrates example parameters and their associated risk probability scores; values are assigned based on the specific domain security requirements and expertise of the security managers and administration. In this example scenario we assigned these constants to reflect our initial network environment, and values are assigned depending on the importance of the concerned factor and their magnitude.

We consider the data security level as a risk, therefore it contributes to the quantification of the final security grade for the node. These grades are then incorporated into the risk mitigation process. All parameters used to quantify and assess the appropriateness of the reconfigured network are shown in Table 4.4.

The quantification of each node's security grade is shown in Equations 4.28 and 4.29, where A is the security grade arithmetic mean, n is the number of node vulnerabilities, v is the value of the identified vulnerabilities, \bar{A} is the weighted security grade arithmetic mean, and w is the weight of the vulnerabilities data source.

Table 4.4. Simulated Risk Parameters and Associated Risk Probability Scores

Centralities	Risk Type and Probability Score	Other
Degree (0 – 1)	Firewall Status (0 or 10)	Robustness
Betweenness (0 – 1)	IDS (0 or 10)	Energy Level (0 – 10)
Closeness (0 – 1)	Encryption Status (0 or 10)	Cost
Eigenvector	Encryption Method (0 – 10)	Minimum Path Average
Bridging (0 – 1)	Operating System (0 – 10)	Security Grade (0 – 10)
	Staff Skill Level (0 – 10)	
	System updated (0 or 10)	
	Anti-Virus/Security (0 or 10)	
	Internet Access (0 or 10)	
	Data Security Level (0 – 10)	
	Identified Vulnerabilities (CVSS v3 Base Score)	

In order to establish a single security grade using the scores assigned by the identified risks, as we know the source and reliability of each score there is no need to use a weighted average, instead we can apply arithmetic mean. To determine the security grade arithmetic mean A , assume each node will have n vulnerabilities collated based on its risk assessment. Denote the values of the n vulnerabilities by v_1, v_2, \dots, v_n , which are the assigned values of the identified vulnerabilities [252], defined as:

$$A = \frac{v_1 + v_2 + \dots + v_n}{n} \quad (4.28)$$

Should we be forced to incorporate sources of data that are unreliable or should the data's weight not be equal, then we can quantify a single security grade using weighted arithmetic mean. This would assist to determine the security grade's weighted arithmetic mean \bar{A} assuming that each node has a set of vulnerabilities V_1, V_2, \dots, V_n , which have respective weights of w_1, w_2, \dots, w_n , defined as:

$$\bar{A} = \frac{w_1 V_1 + w_2 V_2 + \dots + w_n V_n}{w_1 + w_2 + \dots + w_n} \quad (4.29)$$

4.7.2 Network Data Flow Security Level

The quantification of the networks data flow security is shown in Equation 4.30, where $S(N)$ is the security level, G is the set of potential grades, N is the Network, V_g is the set of nodes with the required authorisation level, g is the given data sensitivity level, $\delta_{s,t}(g)$ is the step function, and s and t are the nodes between secure paths.

To determine the data security level of a network, we assume that the nodes which form the network are static, yet have dynamic connectivity (i.e. nodes can change communication links). Each node within the network will be assigned an authorisation level, as discussed in Section 4.5. In terms of security, it is vital that elements of data are only passed via nodes along communication links with sufficient authorisation levels for that data flow. On this basis, network data flow security is measured as follows.

$$S(N) = \frac{\sum_{\forall g \in G} \sum_{s,t \in V_g, s \neq t} \delta_{s,t}(g)}{\sum_{\forall g \in G} |V_g| \times (|V_g| - 1)} \quad (4.30)$$

In this equation G is the set of different grades that nodes inside the network N might have assigned, V_g is the set of nodes inside the network that reach the required authorisation level to access the given data at level g , $\delta_{s,t}(g)$ is a step function taking the value 1 if it's possible to find a secure path between s and t , given the sensitivity level g and 0 otherwise; and $n=|N|$ is the number of nodes within the network. In essence $S(N)$ represents the proportion of secure paths between pairs of nodes that are entitled to communicate.

4.6.3 Robustness Function

Once a network has either been simulated or imported into the SCRAM framework, each node is measured by the means of a robustness function (Equation 4.31). To determine the optimal configured network that mitigates risks and increases the overall SoS security, five main criteria are used as a guide. These are the communication security level $S(N)$, highest bridging centrality score $C_R(v^*)$, degree centrality of the network $C_D(G)$, average minimum path length f_{min} , and total cost C . The robustness function is defined as:

$$\phi(i) = [a_1 C_R(v^*) + a_2 C_D(G) + a_3 f_{min} + a_4 C] / S(N) \quad (4.31)$$

Here v^* is the node with the highest bridging centrality. As the robustness function shows, the main factor is the communications security level achieved. Values for the constants are as follows:

$$a1 = 50000,$$

$$a2 = 4000,$$

$$a3 = 60,$$

$$a4 = 10.5$$

Constant values are determined by analysis of an organisation's network and would be assigned by the security experts and administration. In this example scenario, we assigned these constants to reflect our networks environments, $a1$ represents the highest bridging centrality, $a2$ is assigned the centrality degree, $a3$ minimum path average, and $a4$ associated network cost in terms of distance between nodes. The values assigned to these constants not only depend on the importance of the concerned factor, but also on the magnitude. For example, while centralities generate low numbers, the cost tends to be significantly higher. The lower the robustness score, the more appropriate the individual evaluated. It has been ascertained that the robustness increase is inversely proportional to $S(N)$, and that as the other factors increase so does the robustness. The motive being, that we require $S(N)$ to be maximised and all other factors to be minimised, as searching for a lower robustness level, means instigating higher communication security, while preserving low cost, degree centrality, bridging centrality, and average minimum path length.

4.7.4 Risk Mitigation

In order to enhance the level of security risk within SoS environments, we outline an evolutionary algorithm and probabilistic technique that can be applied to the collaborative infrastructure, that reconfigures communication links between networked devices into an optimal configuration. In order to assure data as it traverses across the collaborative infrastructure, mitigate risks, and enhance network security, while observing access control requirements, high centrality node risk, and insecure vulnerable devices, this process overcomes many of the limitations associated with local search techniques.

To evaluate and ascertain the effectiveness of our proposed solution we implemented two additional algorithms into SCRAM (Ant Colony Optimisation combined with Local Search and Tabu Search) discussed below in Sections 4.7.4.2 and 4.7.4.3, allowing us to critically evaluate the technique's efficiency and suitability.

4.7.4.1 Genetic Algorithm

Influenced by nature's capacity to overcome adversity and ability to manage and sense risk, we researched and developed an evolutionary technique in order to establish an effective security enhancement and risk mitigation solution. For a living organism to survive in the depths of the ocean, they must overcome adversity with the capacity to survive countless life threatening risks. Furthermore, light fails to penetrate the largest depths of the ocean, meaning the life that lives close to the bottom of the sea bed is encompassed in darkness. To survive, predators such as deep sea jelly fish

have evolved to be energy efficient due to long periods without being able to feed, and they must also be capable on a different level of sensing the environment, navigating risks effectively, and signalling threats. Through evolution they have adapted to identify risks without the ability to see them or their entire surroundings, which has inspired our research and proposed solution [253].

Genetic Algorithms are based on natural selection, imitating biological evolution. These algorithms overcome many of the limitations associated with local search techniques, when applied to large complex networks. The basis of the algorithm is to take an initial set of potential solutions, then evolve the set to become a set of best solutions. Through the evolutionary process, inadequate solutions die out, whereas the qualities of the superior solutions are amalgamated and disseminated through to new solutions, which are added to the set. Set size remains constant, so as new better solutions are identified they replace the older inadequate solutions. Random mutations are applied to the new generated solutions, ensuring that the new set of best solutions does not evolve into a set of duplicated solutions. The evolutionary process continues until a predetermined end criterion is met [254] [255] [256].

The initial population of individuals used by the Genetic Algorithm (GA) is the original network (encoded as an individual), along with a collection of randomly generated alternatives. For the purpose of our work, the population size is constant and set to be 10 individuals (i.e. the original network plus 9 random networks). Once a population has been generated, every individual is measured by means of the robustness function (described in Section 4.7.3).

After evaluating every individual within the population, the best individual is directly passed to the next generation. Three individuals in the new generation are chosen by contest from the previous generation, the contest passes the best one from these three to the new population. Four individuals in the new generation are chosen by crossing over two different individuals, which have been randomly chosen. Finally, new random individuals are generated and added to the new population, so that the next generation has 10 new populations in total. After running the cross over and random generation processes, the feasibility of the new individual is checked. Unconnected nodes are prohibited, so if any node is identified as isolated, the new individual is mutated until it is feasible.

New generations are built consecutively. At this point we run the evolutionary process for 2,000 rounds, after which we discontinue the application for the GA and the best individual amongst the remaining solutions is selected as the optimal candidate. An outline of the algorithm's pseudo-code is as follows:

Algorithm 1 Pseudo-Code for Genetic Algorithm

```

1: Initialise population with original network (encoded as an individual)  $N_{orig}$ 
2: Next generation array  $N_{Gen}[10]$  equals  $N_{orig}$  plus nine randomly generated populations
3: while stopping criteria is not reached do
4:   for generations  $g$  do
5:     Calculate the robustness of  $N_{Gen}[g]$ 
6:   end for
7:   for generations  $g$  do
8:     if  $R_{best} = 0$  or  $N_{Gen}[g](robustness) < R_{best}$  then
9:        $R_{best} \leftarrow N_{Gen}[g](robustness)$ 
10:    end if
11:  end for
12:   $N_{Gen}[0] \leftarrow R_{best}$  (next population)
13:  Select three random individuals from previous generation, put in random contest with best
  individual passed to next generation (next population)
14:  Four individuals from new generation are chosen by crossing over two different individuals
  which have been randomly chosen, then passed to next population
15:  Generate new random individuals and add to the new generation until next population equals
  10 individuals
16: end while
17: return best individual from improved solutions

```

4.7.4.2 Ant Colony Optimisation Combined with Local Search

The ant colony optimisation algorithm is based on the natural foraging behaviour of ants. While the algorithm can assist greatly when applied as part of an optimisation process, it does have limitations and commonly has to be combined with an alternative algorithm. The basis of the ant colony optimisation algorithm is to initiate a solution and then update the pheromone trails (i.e. update the comparison parameters). Throughout all iterations, as a new solution is constructed, the pheromone trails are compared (i.e. checking for the optimum secure path). After the improved solution is identified the pheromone trail (comparison parameters) is updated with the enhanced parameters. For example, for ants this would be based on the quantity and quality of the food found, trails with a high pheromone would guide ants to a better source. In this scenario, increasing comparison scores would signify that the new solution is impacting comparative centralities and increasing associated risk within the mutated infrastructure. The algorithm continues processes until the predetermined end criterion is met [254] [257].

The local search method is a simplistic algorithm. The basis of the algorithm is to initiate a solution; the solution is then iteratively evolved, i.e. throughout all iterations the algorithm searches for a better solution, until the predetermined end criterion is met [254]. An outline of the algorithm's pseudo-code based on a combination of ant colony optimisation and local search is as follows:

Algorithm 2 Pseudo-Code for Ant Colony Optimisation combined with Local Search

- 1: Initialise population with original network (encoded as an individual)
 - 2: Calculate original populations Robustness R_{old}
 - 3: Initialise parameters
 - 4: Initialise solution trails
 - 5: **while** stopping criteria is not reached **do**
 - 6: Generate a new random solution
 - 7: Calculate new solutions Robustness R_{new}
 - 8: Calculate parameters
 - 9: **if** $R_{new} < R_{old}$ **then**
 - 10: $R_{old} \leftarrow R_{new}$
 - 11: Update solution trails with parameters
 - 12: **end if**
 - 13: Compare all solutions sort into descending order
 - 14: **end while**
 - 15: **return** five improved solutions and identify solutions with their respective costs
-

The initial population of individuals used by this algorithm is the original network (encoded as an individual), along with a collection of randomly generated alternatives. For the purpose of our work, we generate and compare 10 individuals for each cycle of the security process. Once the population has been generated, the solution trail (pheromone trail) is assigned the original network's comparison parameters (i.e. this is the best solution we begin with hence these are the parameters that need to be compared and improved). Every individual is then measured by means of the robustness function (described in Section 4.7.3).

After evaluating every individual within the population, each solution is compared against the best robustness score, in an attempt to find an improved generation. Should the cycle produce a better solution, then the solution trail is updated with the new solution's comparison parameters. After each cycle we compare each improved generation's parameters in the solution trail, placing them into descending order, ensuring that we only keep the 5 most improved solutions. New generations are built consecutively, and the process is run for 2,000 rounds. We discontinue the application of the algorithm and the best individual amongst the remaining solutions is selected, along with reporting the 5 most improved solutions and identifying the solutions with their respective costs.

4.7.4.3 Tabu Search

Tabu search is a meta-heuristic search method, which uses local search methods for optimisation, along with adaptive memory to explore beyond local optimality and to generate dynamic search method performance. The basis of the search is to prevent the method from re-examining solutions that have already been considered, and to ensure that inadequate solutions are not advanced instead only improvements are developed further. Parameters of preference can also be introduced, so that the search can be influenced into producing a more favourable solution. Tabus tend to be only stored with a limited quantity, as typically there are several possibilities and tabu lists can quickly grow in size, making storage of these parameters and comparison expensive, i.e. restricting the tabu list to only recent improvements and preventing reverse evolution to ensure quick and non-costly processing. The security enhancement and risk mitigation process continues until the predetermined end criterion is met [254] [258]. An outline of the search method's pseudo-code is as follows:

Algorithm 3 Pseudo-Code for Tabu Search

```

1: Initialise population with original network (encoded as an individual)
2: Calculate original populations robustness  $R_{old}$ 
3: Initialise parameters  $P_{best}$ 
4: Generate tabu list  $\leftarrow P_{best}$ 
5: while stopping criteria is not reached do
6:   for generations  $g$  do
7:     Let  $g$  construct new random solution
8:     Calculate parameters  $P_{new}$ 
9:     if  $P_{new}$  not tabu then
10:      Calculate new solutions robustness  $R_{new}$ 
11:      if  $R_{new} < R_{old}$  then
12:         $R_{old} \leftarrow R_{new}$ 
13:      end if
14:    end if
15:    Update tabu list
16:  end for
17: end while
18: return best solution  $P_{best}$ 

```

Initial population used by the security enhancement and risk mitigation process is the initial population (encoded as an individual), along with nine randomly generated alternatives, as we compare ten individuals for each cycle of the process for the purpose of our work. Once the population has been generated, the tabu list is assigned our predefined comparison parameters from the original network, as at this stage this is the best solution and we aim to prevent inferior solutions from being generated and considered. Each solution's predefined parameters are then compared against the tabu list, if parameters match the tabu list they are dropped. Else, if parameters are not tabu, then we calculate the solution's robustness by means of the robustness function (described in Section 4.7.3).

We then compare the solution's robustness against the best robustness, to ensure that the generation is improved. Should the cycle produce a better solution, then the robustness score of the new solution replaces the best solution robustness score, and at the end of the cycle the tabu list is updated ensuring that only improved solutions are considered. New generations are built consecutively, and the search is run for 2,000 cycles. The search application is then discontinued, and the best individual amongst any remaining solutions is presented as the optimal candidate.

4.8 Summary

This chapter provides a high-level overview of the SeCurity Risk Analysis and Mitigation Framework (Section 4.2), and includes a comprehensive review of the framework's design (Section 4.3) and runtime operation (Section 4.4). In these sections we discuss the challenges that must be overcome in order to implement our theoretical solution, and disclose operational specifics and the reasoning in regards to the design and composition of the framework.

Additionally, the chapter outlines the data access control problem and management (Section 4.5), and discusses the principal model for access control within SoS. In Section 4.6 Topological Vulnerabilities, the algorithms used to calculate degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and bridging centrality are presented, these centrality indicators are important as they assist in ascertaining a node's importance within the SoS and underlining risks the nodes pose.

In Section 4.7.1 Node Security Grade Assignment, we discuss the vital techniques used to identify vulnerabilities within the topology of the SoS, and the metrics used to assign numerical numbers to those risks identified. The integral principles used to calculate the security grade of all nodes within the SoS are also outlined in detail, the security parameters used to calculate the security grade are also presented, justifying the choice of algorithm used and discussing an alternative algorithm that could be incorporated into the framework if data sources were deemed unreliable. The algorithm used to quantify the data security level for the entire SoS is presented in Section 4.7.2 Network Data Security Level Quantification.

The robustness function algorithm used to quantify the appropriateness of the SoS environment is outlined in Section 4.7.3, the algorithm uses five main criteria in order to determine the optimal network. It combines the security level of the network, highest bridging centrality score, degree centrality of the network, the average minimum path length, and total cost of the communication network in terms of distance between nodes. This section also summarised how the algorithm's constant values are assigned, which depend on the importance of factors and their magnitude.

This chapter also presents the integral security risk mitigation algorithms (Section 4.7.4) used to increase the security of the SoS environment, which evolve the network throughout their applications, considering factors such as access control, risks associated with high centrality nodes, network cost, and node security. These algorithms are applied to the network and overcome the associated limitations of local search techniques, and prevent evolvment from duplicating solutions and ensuring inadequate solutions die out. Meaning only superior solutions continue to be developed, thus, enhance the robustness, security, and structure of the SoS. These three algorithms are evaluated against each other in order to establish the most effective means for measuring and increasing security between interconnected devices and mitigating risks within both SoS and multi-level SoS environment.

Chapter 5

Implementation and Evaluations

To positively evaluate the effectiveness of the SCRAM framework presented in this thesis, we have developed and implemented our solution into an operational application. This implementation ensures we can corroborate the theoretical principles proposed, and confirms that SCRAM conforms to the aims, objectives, and requirements we established. In this chapter, we convey the application of the principles employed by the SCRAM framework (presented in Section 4), the simulation environment, and the evaluation practices.

5.1 SCRAM Framework

As SCRAM is to be applied to diverse and dynamic SoS and multi-level SoS it is essential that the framework does not negatively impede the resources of the network, which are often restricted due to the types of devices that they are formed from (e.g. WSN, IoT, etc.). The complex topologies of multi-level SoS mean that, in order to monitor an entire infrastructure and identify associated node vulnerabilities, while meeting the identified objectives discussed in this thesis, we needed to implement SCRAM in a widely used programming language; therefore we decided to write the framework using Java. This programming language can be applied to most operating systems and platforms, is object oriented, high performance, dynamic, and designed to have minor implementation dependencies.

The simulated SCRAM framework is implemented in NetBeans IDE, which provides a good, lightweight, open sourced, integrated development environment. As stated we developed the framework in Java, but other programming languages are supported, these include for example, C/C++ and PHP. The SCRAM framework is self-contained and is programmed to replicate networks comprising of different devices, each with varying software, hardware, and firmware configurations, and connected via a series of varying communication links. SCRAM is not designed to be applied to a specific operating system, this ensures that the framework is suitable and can easily be applied to various platforms.

The SCRAM framework was implemented with a user interface (as illustrated in Figure 5.1) to allow an existing network to be imported into the framework, or for a user defined network to be selected and generated within the environment. The initial simulation environments did not simulate vulnerabilities; instead nodes were assigned with random security grades to represent the node's risk

within the topology. This allowed for us to quickly develop the key principles and generate a working simulation environment, and provided a platform for us to evaluate the main techniques.

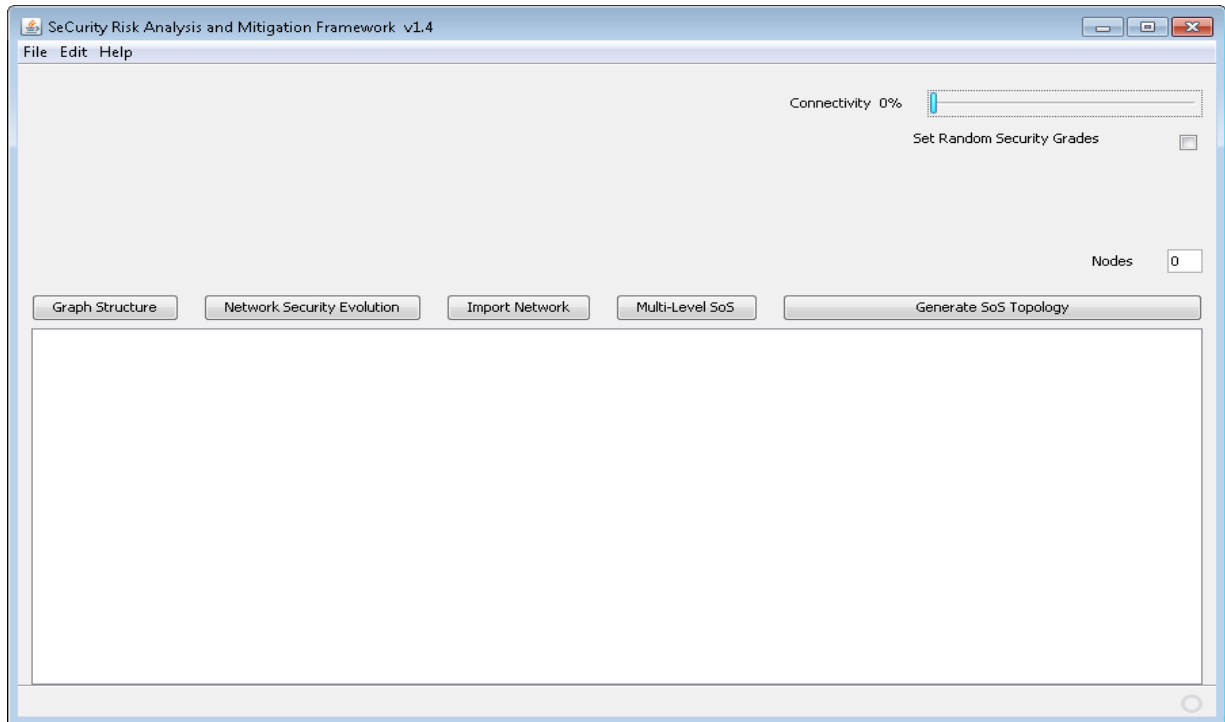


Figure 5.1. SCRAM Screenshot

In order to improve the accuracy of node security grades and improve the overall network security we evolved the simulation to assign vulnerabilities to each node based upon the type of device being simulated and its software, hardware, and firmware configuration. Finally, SCRAM was advanced further and reported vulnerabilities and their CVSS v3 risk probability scores were incorporated into the framework and associated principles, including the security grade. This was accomplished by importing numerous vulnerabilities into the SCRAM framework that were identified then analysed and reported via the NVD website with their associated CVSS v3 risk probability scores.

These vulnerability scores are quantified along with the risks posed by the node's software, firmware and hardware, into the node's final security grade, establishing a more accurate scoring practice for the nodes and improving the accuracy of the overall network communication security. All simulated vulnerabilities correspond to the devices that are simulated within SCRAM. The number of vulnerabilities simulated is limited within the framework as the database sizes within SCRAM would quickly grow in size depending on the type of simulated device and the scale of the SoS.

The security enhancement and risk mitigation algorithms that support the reconfiguration and security improvements of the network have the ability to manage large complex optimisation problems. The proposed security risk mitigation and robustness function (described in Section 4.6) has been implemented as part of SCRAM in order to mitigate risks within the multi-level SoS, and improve security without introducing additional resources into the infrastructure. This is achieved by

identifying vulnerabilities within nodes and data access violation, and reconfiguring network communication paths between nodes based on the conducted multi-level risk assessment. By means of this framework we have successfully conducted a series of experiments to evaluate their effectiveness when applied to SoS environments.

5.1.1 Network Generation

The SeCurity Risk Analysis and Mitigation Framework has the functionality to either generate a random network or import an existing network topology. Random networks are simulated by first selecting the number of nodes the network will be composed from and its initial connectivity, the framework then produces the SoS infrastructure and generates an SoS configuration file (see example Figure 5.2). Our primary test network as visualised in Figure 5.3 visualises our SoS which is composed of 8 static nodes with a low connectivity of 30%. The functionality of SCRAM randomly assigned all nodes with a security level, data grade, and connected nodes with a series of primary links. It then quantifies the network's degree, betweenness, closeness, eigenvector, and bridging centralities, the communications security, minimum path average, and the network's associated cost. The framework selected a random network data level, in this instance the data level has been assigned Level 4, which all nodes will be compared against, in order to replicate data access control requirements.

The SoS network configuration file ensures that we can import the same networked infrastructure within the framework for evaluation against different security enhancement and risk mitigation algorithms. The network configuration file also directly supports the generation of the undirected graph, as it contains several key parameters including node coordinates, communication paths and lengths, which are used to position the nodes and generate all communication links between components forming the SoS infrastructure. The file also details the assigned node identification number, grade which is used during risk mitigation processing, the quantified security grade used by both the graph generation processes and risk mitigation algorithms, the node security status which is used when generating undirected graphs, and the data grade used by both the graph generation processes and risk mitigation algorithms.

```

SoS_NETWORK_GRAPH
@nodes8
@size500,500
# ,278,174
# ,446,47
# ,26,354
# ,189,60
# ,258,309
# ,402,216
# ,25,180
# ,382,186
% ,0,1,211.0
% ,0,2,0.0
% ,0,3,145.0
% ,0,4,137.0
% ,0,5,131.0
% ,0,6,0.0
% ,0,7,105.0
% ,1,2,0.0
% ,1,3,0.0
% ,1,4,0.0
% ,1,5,175.0
% ,1,6,0.0
% ,1,7,154.0
% ,2,3,0.0
% ,2,4,0.0
% ,2,5,0.0
% ,2,6,175.0
% ,2,7,0.0
% ,3,4,0.0
% ,3,5,0.0
% ,3,6,204.0
% ,3,7,0.0
% ,4,5,172.0
% ,4,6,0.0
% ,4,7,175.0
% ,5,6,0.0
% ,5,7,37.0
% ,6,7,0.0
$ ,0,4,4,4,4
$ ,1,4,4,4,4
$ ,2,4,4,4,4
$ ,3,10,3,10,5
$ ,4,10,4,10,8
$ ,5,10,5,10,9
$ ,6,5,5,5,3
$ ,7,10,6,10,6
@security,52,

```

Node Coordinates
#, <x>, <y>

Communication Paths between Nodes
%, < node id >, < node id >, < path length >

Node Security Parameters
\$, < node id >, < grade >, < security grade number >, < security status number >, < data grade >

SoS Communication Security
@security , < coms security % >

Figure 5.2. Primary Test SoS Network Configuration File

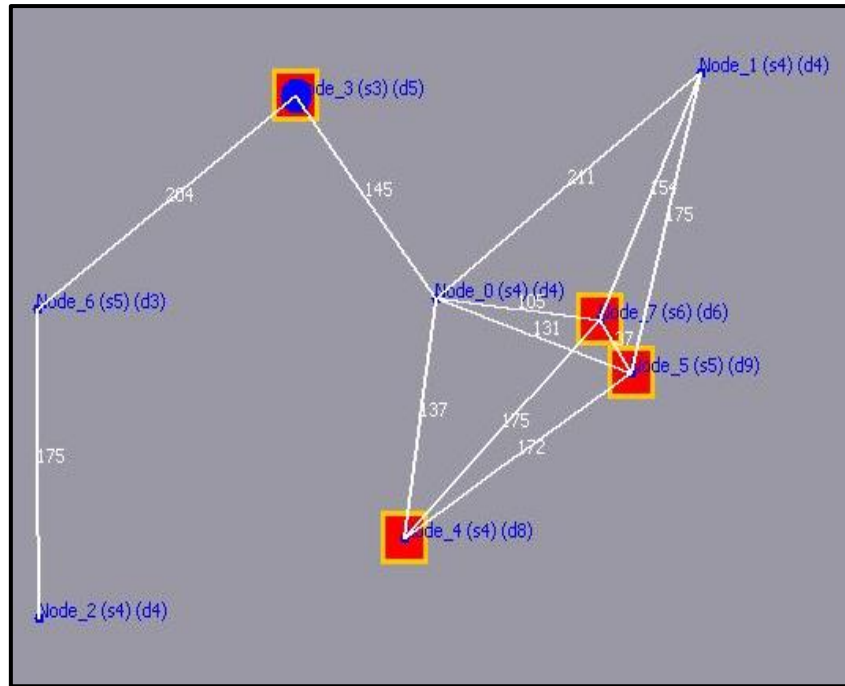






Figure 5.3. Primary Simulated Network Environment

Key parameters are visualised in the graph such as node identification, security grade, data access level, and node status, and an initial report is constructed and visualised in the user interface containing centrality values and security details (Figure 5.4). The framework identifies and visualises nodes which are in ready status, blocked or are considered insecure. Nodes identified as ready have no warning markers, as these nodes have either been assigned with an equal or higher data access level above the assigned network data level. Table 5.1 depicts the visualised parameters used to generate the initial undirected graphs.

Table 5.1 Initial Visualised Security Vulnerabilities and Parameters

Graph	Parameter	Symbol	Description
	Node within the SoS.		Blue node/tag.
	Node size represents quantified bridging centrality score, i.e. the width of the node is proportion to its bridging centrality value.		
Security	Insecure node.		Node encased with a solid orange box.
	Blocked node.		Node encased with a solid red box.
	Blocked and Insecure node.		Node encased with a solid red box with orange border.

- Ready nodes have had their security quantified as secure, meaning their security risk has been scored low thus have high security statuses.
- Blocked nodes are visualised with a red box and an orange border, as these nodes have been assigned with a data access level lower than the allocated network data level, and data flow should be blocked from traversing via these nodes. Nodes with unauthorised data levels are also considered to not only pose risks to the data flow but also could expose the network topology, thus are considered to be insecure in terms of a node's security.
- Insecure nodes are visualised with a large orange block, these nodes have had their security quantified as insecure, meaning their security risk has scored high thus has a low security level. These nodes pose risks to the network as they leave it exposed to potential vulnerabilities and attack vectors. Critical data should not traverse through these nodes, but instead be routed through ready nodes with high security and appropriate data access.
- Nodes quantified with higher bridging centralities are represented with wider nodes; an example of this is node 3 in our primary network (see Figure 5.3), the width of the node is in proportion to its bridging centrality value.

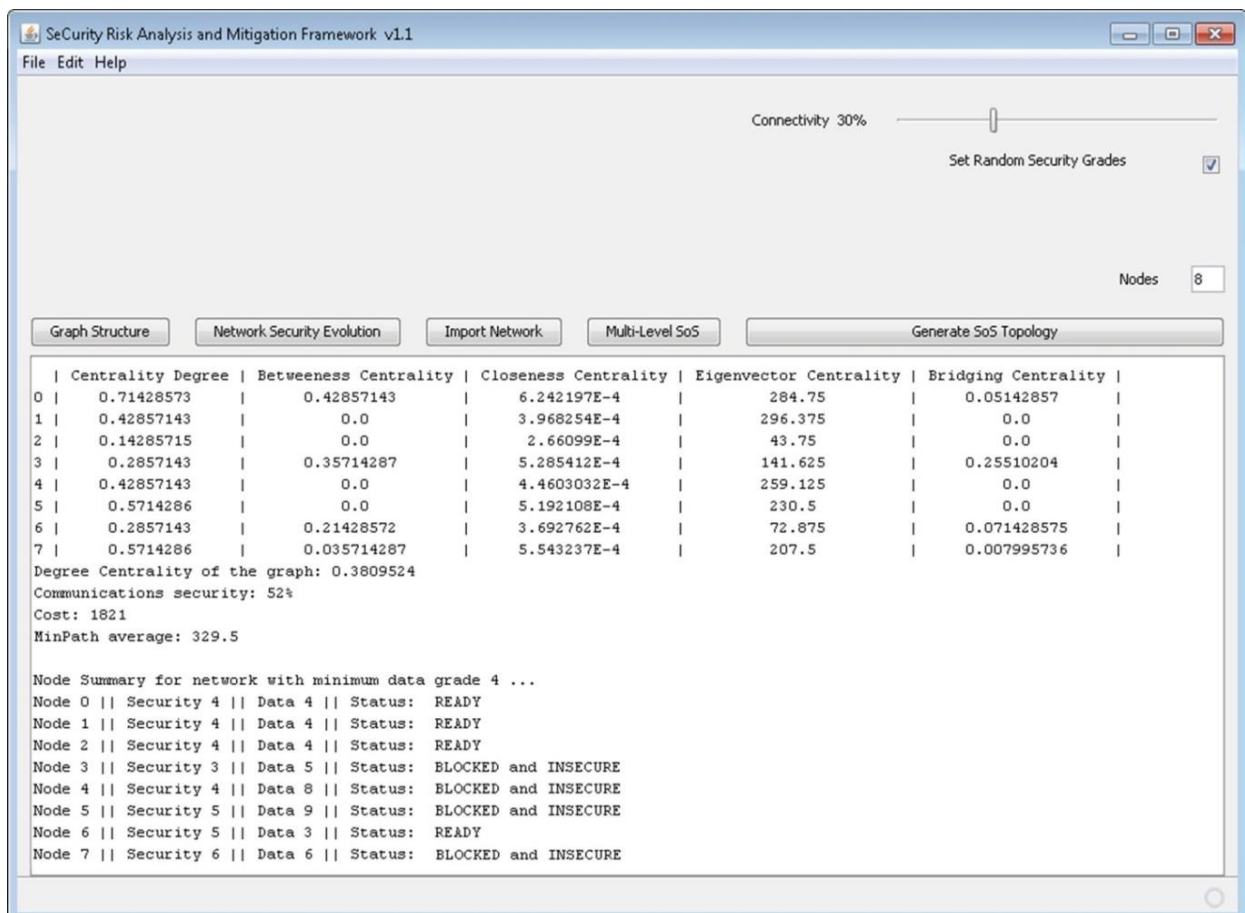


Figure 5.4. Primary Simulated Network Environment User Interface Report

5.1.2 Topological Vulnerabilities Analysis

When a network is generated or imported into the SCRAM framework, and during the security risk mitigation process, graphs are generated that represent the topology of the SoS and the process initiates the centrality methods described in Section 4.5. These centrality methods measure degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and bridging centrality. Each centrality when analysed can indicate the appropriateness of the networks communication path configuration identifying configurations that increase the SoS to additional risk(s), and can assist to indicate nodes that are more influential within the SoS, are potential SPoF, and are greatly depended upon by other nodes, etc., and increase risk within the SoS due to their topological vulnerabilities.

```

public class DegreeCentrality implements CentralityMeasure {
    private Graph _graph;
    public DegreeCentrality(Graph graph){
        _graph = graph;
    }
    public float getVertexDegree(Node n){
        int degree;
        int size;
        degree = _graph.getLinkDegree(n.getId());
        size = _graph.getnNodes();
        return (float)((float)degree / (float)(size - 1));
    }
    public float getGraphDegree(){
        float aux[];
        float sum;
        int max;
        sum = 0;
        aux = new float[_graph.getnNodes()];
        max = 0;
        for(int i = 0; i < _graph.getnNodes(); i++){
            aux[i] = this.getVertexDegree(_graph.getNodes()[i]);
            if(aux[i] > aux[max])
                max = i;
        }
        for(int i = 0; i < _graph.getnNodes(); i++){
            sum += aux[max] - aux[i];
        }
        return (float)((float)sum / (float)(_graph.getnNodes() - 2));
    }
    public int getMeasurementType() {
        return NetworkGraph.CENTRALITY_DEGREE;
    }
}

```

Figure 5.5. Code Excerpt Showing Degree Centrality Method

Each centrality has been coded into the SCRAM framework, and a code expert for generating the degree centrality for each node is shown in Figure 5.5 as an example. Degree, betweenness, closeness, and bridging centrality are all scored on a scale of 0 to 1, with eigenvector being scored from 0+-. Centralities will differ and be influenced by topological data such as the number of nodes that form the SoS, node locations, connectivity percentage, and communication links, etc. Consequently, we cannot have a predefined maximum or minimum comparative value as the network is manipulated into a more secure solution, i.e. if the centrality value does not fall within a set range it would be considered a negative impediment to the centrality. Therefore, for each experiment and reported evolved candidate we review the centrality score itself and the percentage difference between the original network and evolved reported network candidate.

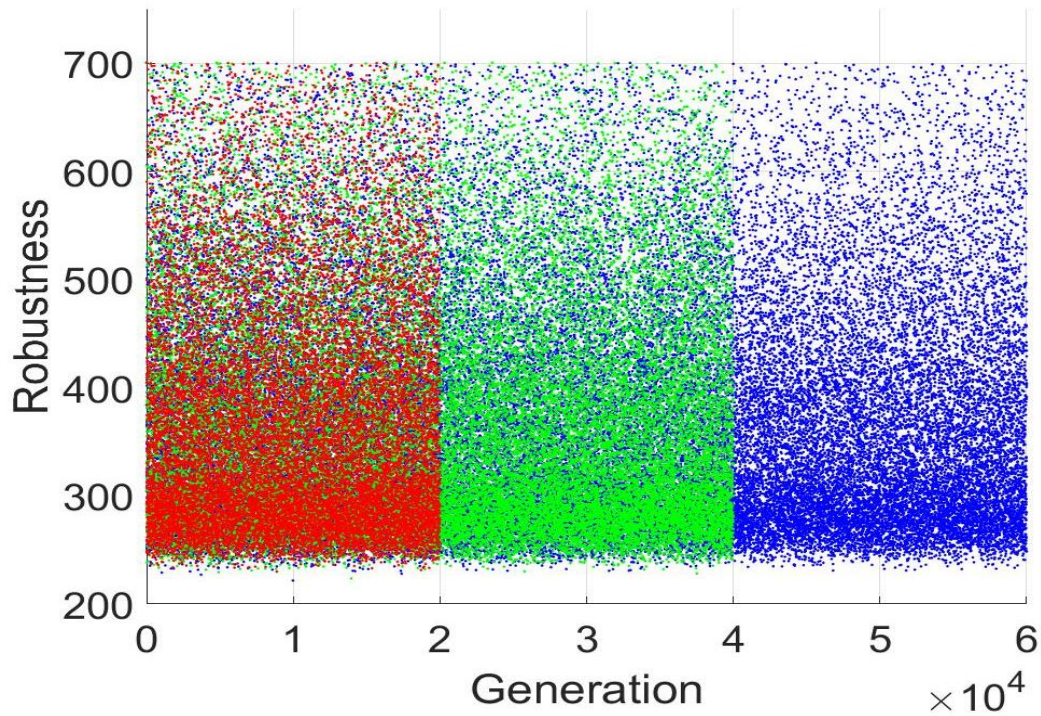
Depending on the network's security requirements and administrators' perceived tolerance to risk, centrality tolerance could be increased or decreased to reflect the organisation's systems and needs, or even frequently reviewed and altered to keep up with alterations and enhancements to the SoS. For example, we could classify anything that increases degree centrality by over 100% is a negative impediment, while any increase below 100% is within a tolerable range and considered acceptable. For infrastructures deemed critical, the tolerance could be reduced for example to 50% or even a lower and stricter criteria level. Centrality values and their associated topological vulnerabilities are discussed throughout our undertaken experiments in the following sections.

5.1.3 SCRAM Framework Cycle Analysis

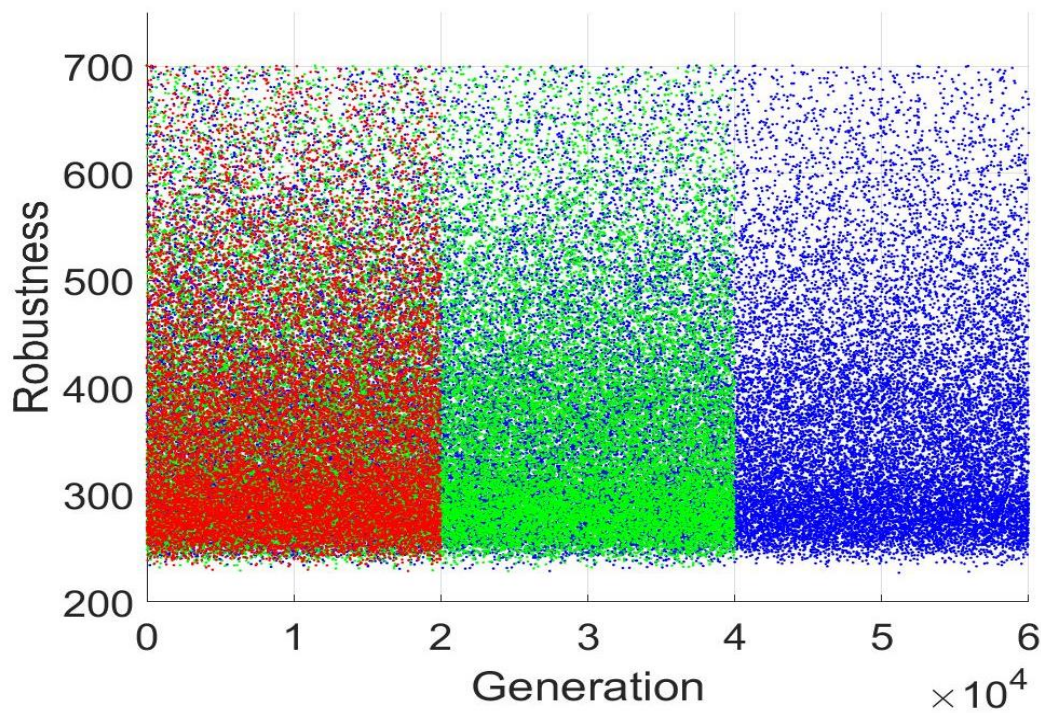
When incorporating the algorithms into the risk mitigation process, it was imperative to ascertain the optimum number of cycles for the length of the evolutionary process. The framework's risk mitigation process deletes and replaces communication links between nodes during evolutionary stages searching for an optimal combination. Should the process be run for too few cycles, while potentially we could still find a more optimum solution, it might not have improved significantly enough to have an impact on the robustness and security of the network. If the evolutionary process is run for too many cycles, then the method's run time will increase which could impact real time analysis, required operation resources, operation costs, and could directly impede distinct nodes operation and resources. There is no guarantee the longer the process runs the better the solution will be, i.e. due to random mutations in the evolutionary risk mitigation process the best robustness level could be found very early in the cycle.

As evident from the network evolutionary process analysis in Figure 5.6, without the use of a suitable optimisation algorithm the process to evolve the network and proposed robustness function generates an excessive volume of new generations that do not greatly improve the network's security and robustness, visible for all generated cycles in both Assessments 1 and 2. For both assessments we

used the same example network (see Figure 5.3), and ran the robustness function starting with 2,000 cycles, then 4,000 cycles, and finally for 6,000 cycles. In this instance, we have excluded all solutions that negatively evolved the network within the graph structure, i.e. every reported candidate with an increased robustness score, to reduce graph size.



(a) Improved Network Security and Robustness: Assessment 1



(b) Improved Network Security and Robustness: Assessment 2

Figure 5.6. Analysis of the Network Evolutionary Process

We compare 10 generations for comparison in each cycle; this gives us 20,000, 40,000 and 60,000 new randomly evolved networks respectively. While the largest concentrations of newly evolved networks have a lower robustness level as evidenced by Figure 5.6, it is difficult to intuitively identify the lowest robustness score in any of the cycles for both assessments, and there is no distinguishable robustness network evolution progression.

When we critically analyse the data and filter out duplicated evolved solutions keeping the best robustness score in each assessment (see Table 5.2), in Assessment 1 the best robustness level achieved for 2,000, 4,000, and 6,000 cycles was generated within the first 2,000 cycles. When we analysed the second assessment data, specifically if we look at the best robustness for 6,000 cycles, we see different best robustness scores during its first 2,000 generations through to its final 6,000 generations, where it finds the optimum solution at generation 51,169. In the first 2,000 cycles of Assessment 2, the best solutions robustness score is 230.944 which is a 67.04% decrease compared to the original robustness level of the network (i.e. the lower the robustness level, the more appropriate the individual evaluated), the best robustness for 4,000 cycles was 228.865 which is a 67.33% decrease, and the best solution for 6,000 cycles was 227.283 which is a decrease of 67.56%.

Table 5.2. Comparing Improved Solutions Robustness During Evolutionary Process Cycles

Round	Evolution Results									
	Number of cycles	Original network robustness	Number of improved solutions	Number of improved solutions (duplicates removed)	2000 Cycles		4000 Cycles		6000 Cycles	
				Generation and Best robustness	Generation and Best robustness	Generation and Best robustness	Generation and Best robustness	Generation and Best robustness	Generation and Best robustness	Generation and Best robustness
1 - 2000	700.623	17,999	17,274	8,876	229.946	-	-	-	-	-
1 - 4000	700.623	38,489	34,489	13,953	223.597	13,953	223.597	-	-	-
1 - 6000	700.623	57,914	51,910	9,976	221.721	9,976	221.721	9,976	221.721	221.721
2 - 2000	700.623	19,268	17,269	17,770	228.860	-	-	-	-	-
2 - 4000	700.623	38,510	34,506	6,355	229.440	24,585	228.854	-	-	-
2 - 6000	700.623	51,774	51,774	17,195	230.944	27,407	228.865	51,169	227.283	227.283

Therefore, running the application for an additional 2,000 cycles provides only a 0.9% better solution with a robustness decrease of 2.079, and the additional 4,000 cycles only provides a 1.59% better solution in the second round with a robustness decrease of 3.661. When we consider the additional time taken to process the extra solutions and the processing requirements needed, against the actual result improvements, we decided that running the application for 2,000 cycles would be optimum at this stage to test the evolutionary risk mitigation and optimisation algorithms as proof on concept.

This will be of vital importance when we increase the size of networks to be examined, as larger simulations will take considerably longer to process, and therefore it is essential that we run tests making use of the optimal resources available, ensuring that we don't negatively impede memory and processing power which could impact simulation times and hinder future real-time analysis on

complex multi-level SoS. Table 5.3 demonstrates the maximum CPU time and memory used for both of the conducted assessments.

Table 5.3. Resource Usage During Evolutionary Process Cycle.

Rounds	Number of cycles	Number of solutions	Time Taken for Security Risk Mitigation Process	Maximum CPU Time	Maximum Heap Memory Used
1	2000	20,000	00:16:00	17.3%	4.83%
1	4000	40,000	00:30:00	18.6%	7.25%
1	6000	60,000	00:48:00	16.5%	5.51%
2	2000	20,000	00:16:00	22.3%	5.19%
2	4000	40,000	00:39:00	16.9%	5.16%
2	6000	60,000	00:45:00	17.1%	8.11%

5.1.4 Applied Network Security Risk Mitigation Principals

Using our primary simulated environment (Figure 5.3), the network has been reimplemented into the SCRAM framework and the three different algorithms presented in Section 4.6.4 have been applied to the network's security risk mitigation process, which influences the evolution stages of the process by searching for an optimal security solution to ensure the network's reconfiguration did not negatively impede but enhanced the communication security between interconnected devices and mitigated risk factors, without having to introduce additional resources into the infrastructure.

5.1.4.1 Genetic Algorithm Evaluation

When the Genetic Algorithm (GA) is applied to the primary network, throughout the evolutionary process random evolvments are made to the new generated solutions. Evident in Figure 5.8, this shows every subsequent security enhanced candidate found from the original network, in a series of undirected graphs. The network was evolved into a set of best solutions as described in Section 4.6.4.1, with the final reported evolvment (Figure 5.8-k) being the optimum configured solution. These configurations are generated from a single run of the GA, which took 21 seconds for completion.

During evolution stages the SCRAM framework searches for an optimal secure network combination, while the security risk mitigation process removes and replaces links. Figure 5.8 Evolution a, is the first reported improved solution and shows an increase in the number of established links, in an attempt to assure communication security. We note that during the algorithm's run time, the evolutions fluctuate between an increase and decrease in communication links until the last configuration is approached.

The robustness monitor in Figure 5.7 records a notable reduction in the network's robustness score, meaning the evolved random solutions are more appropriate and secure. The robustness level of the original network was 700.6233, while the final optimal solution scores a robustness level of 224.9813, achieving a 67.89% improvement. We also note there is a 62.16% decrease in robustness from the first evolved candidate which is an immediate significant improvement at the beginning of the process, which only continues to positively advance throughout the security risk mitigations evolution process.

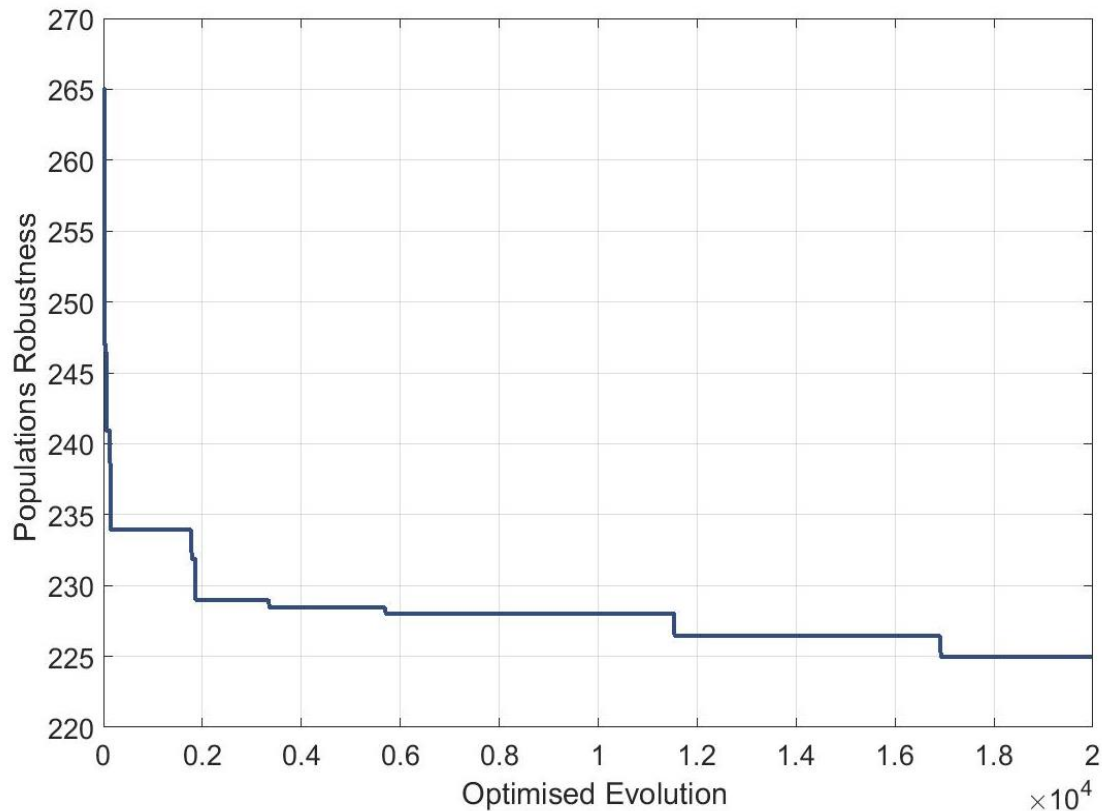


Figure 5.7. Robustness Monitor for the Applied Genetic Algorithm

Through the improved robustness, we succeed in maximising communication security and mitigating risk, while sustaining low bridging centrality (92.22% decrease), and average minimum path length (19.34% decrease). Unfortunately to achieve this, there has been a significant increase to overall network cost, which rose from 1821 to 3801, meaning there is a cost increase of 108.73% to assure a 92.31% improvement for SoS communication security.

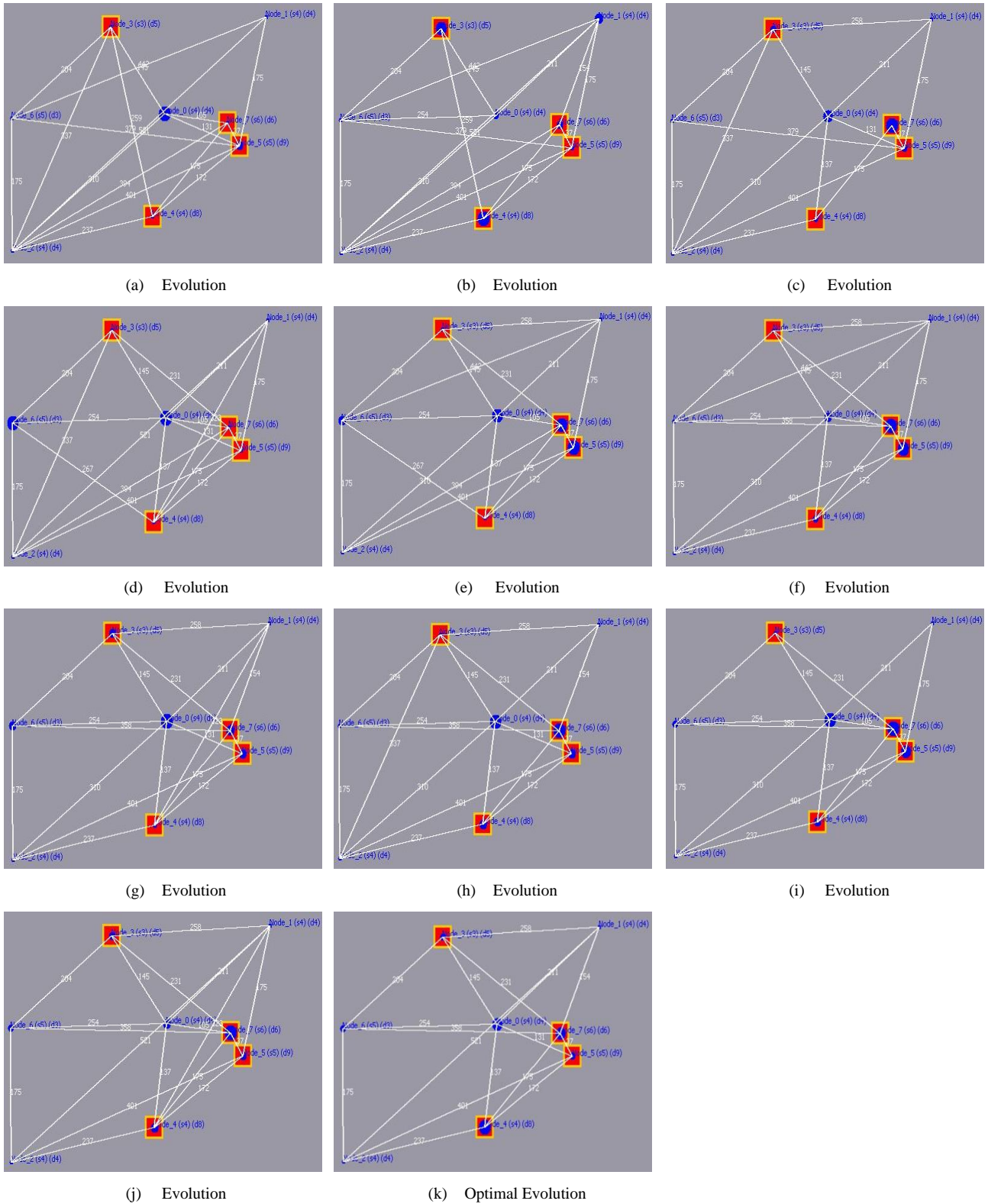


Figure 5.8. Comparison of Genetic Algorithm Improved Security Solutions

5.1.4.2 Ant Colony Optimisation Combined with Local Search Evaluation

When the Ant Colony Optimisation combined with Local Search (ANT) algorithm was applied to the primary network (Figure 5.3), random evolvments were made to the newly generated solutions. Evident in Figure 5.9 which shows every subsequent security enhanced candidate found from the original network, in a series of undirected graphs. The network was evolved into a set of best solutions as described in Section 4.6.4.2, with the final generated evolvment (Figure 5.9-e) being the reported optimum solution. These configurations are generated from a single run of the ANT algorithm, which took 20 seconds for completion.

During evolution stages the SCRAM framework searched for an optimal secure network combination, while the security risk mitigation process removed and replaced links. Figure 5.9 Evolution a, is the first improved evolved solution using ANT and shows an increase in the number of established links, in an attempt to assure communication security. Again we note that during the algorithm's run time, the evolutions fluctuate between an increase and decrease in links until the last configuration is approached.

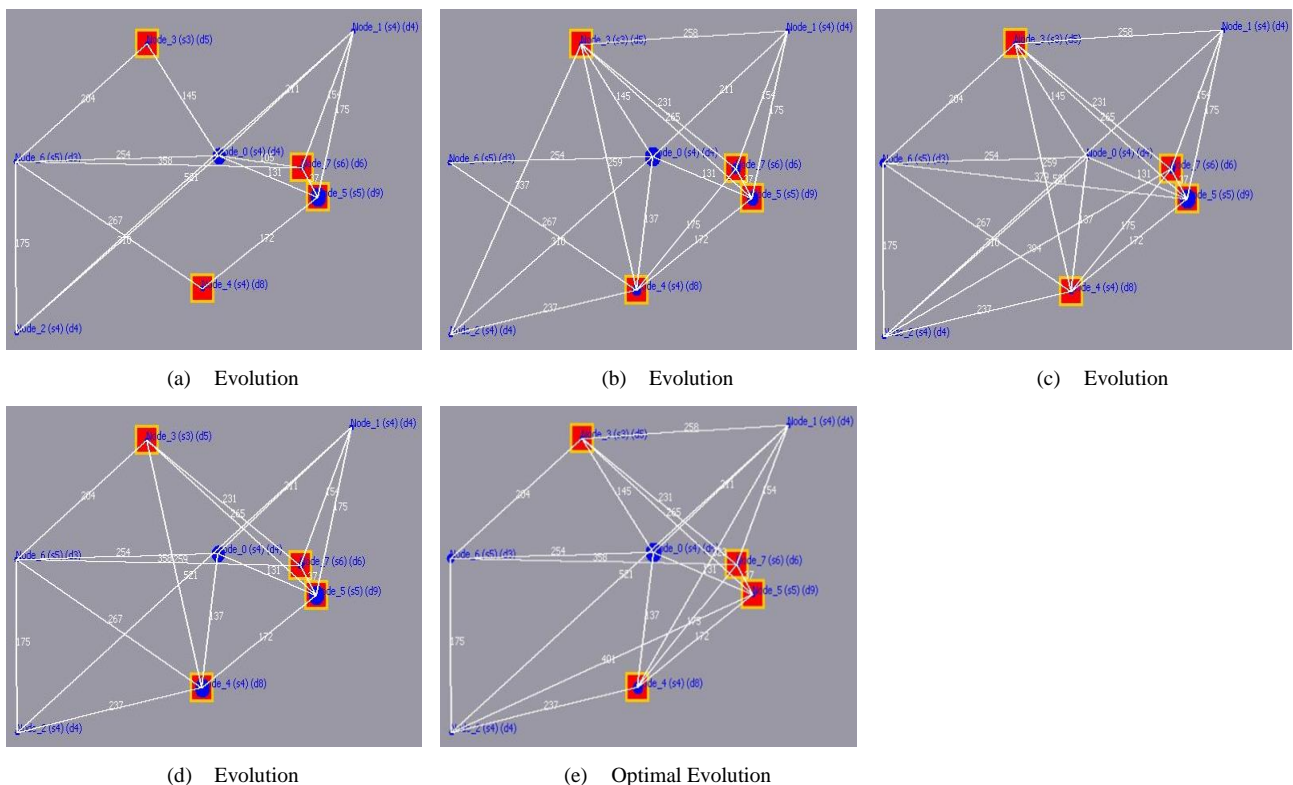


Figure 5.9. Comparison of ANT Algorithm Improved Security Solutions

The robustness monitor in Figure 5.10 records a notable reduction in the network's robustness level, meaning the evolved random solutions are more appropriate and secure. The robustness level of the original network is 700.6233 while the final optimal solution scores a robustness score of 228.9274,

achieving a 67.33% improvement. We also note that there is a 65.07% decrease in robustness from the first evolved candidate, significantly improving the network's robustness which continues to positively evolve throughout the rest of the security risk mitigation's evolution process.

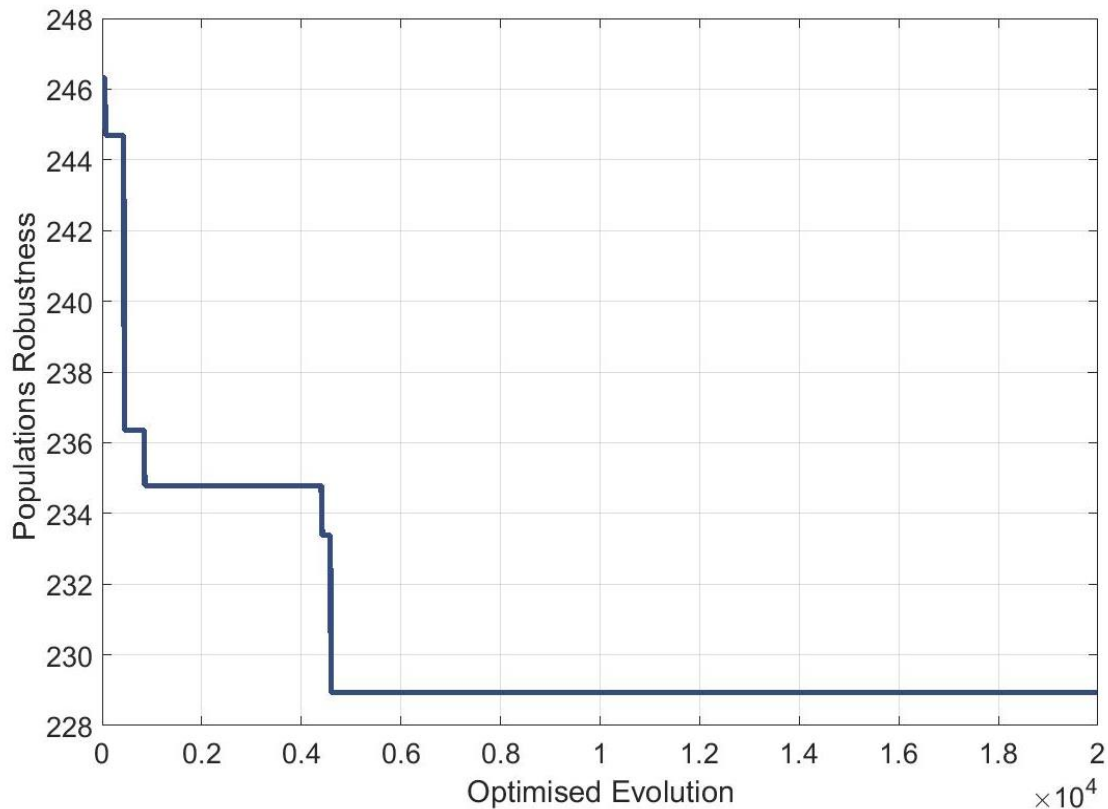


Figure 5.10. Robustness Monitor for the Applied ANT Algorithm

Through the improved robustness, we succeed in maximising communication security and mitigating risk, while sustaining low degree centrality (87.5% decrease), bridging centrality (92% decrease), and average minimum path length (19.43% decrease). Unfortunately to achieve this, there has been a significant increase to overall network cost, which rose from 1821 to 4389, meaning there is a cost increase of 141.02% to assure a 92.31% improvement for SoS communication security.

The generations were built consecutively, and at the end of the risk mitigation process the framework confirmed that it had generated five improved solutions, reporting them in the SCRAM user interface window detailing their associated costs and key parameters (see Figure 5.11). For this single run, while the final solution (Candidate 5) is reported with the lowest robustness score identifying it as the most optimal configured secure solution, evolved Candidate 1 is the lowest costing best solution at only 3219, meaning there would only be a cost increase of 76.77% to assure a 92.31% improvement for communication security. Candidate 1 scored a robustness of 244.6947, achieving a 65.07% improvement.

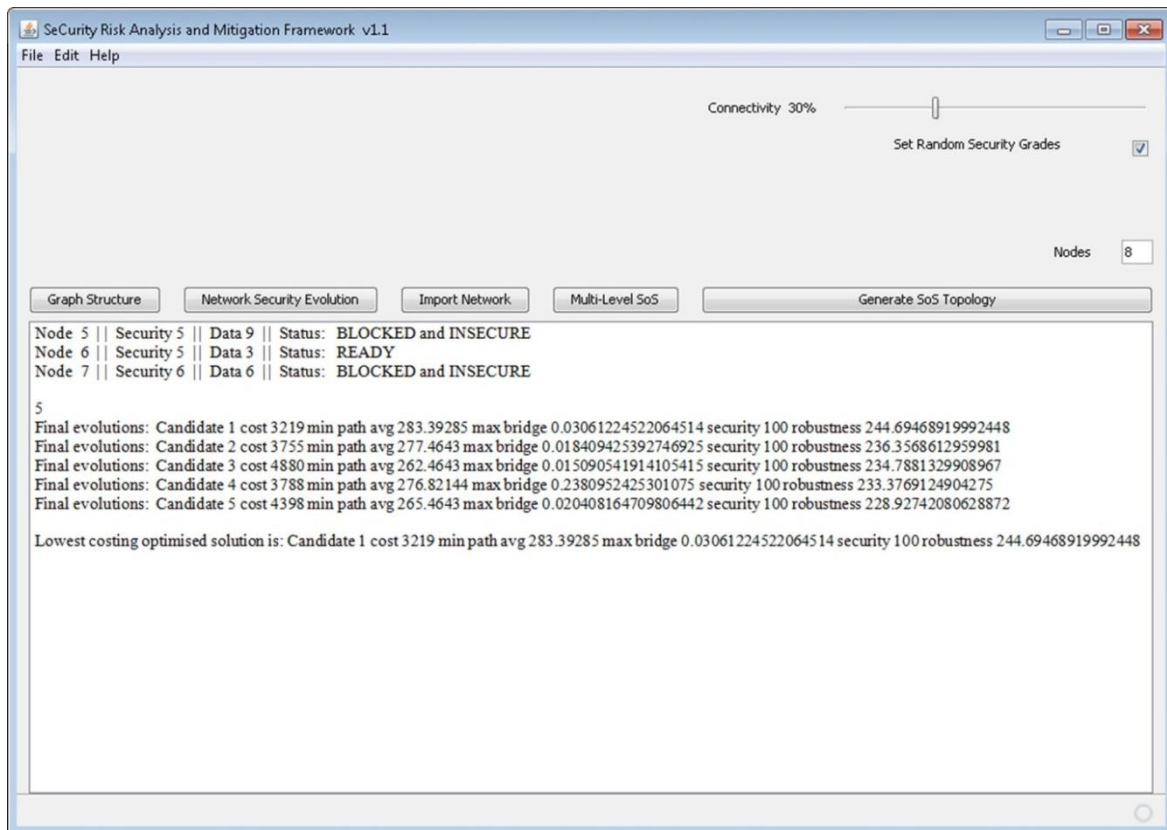


Figure 5.11. SCRAM Output Window Identifying ANT Best Solutions

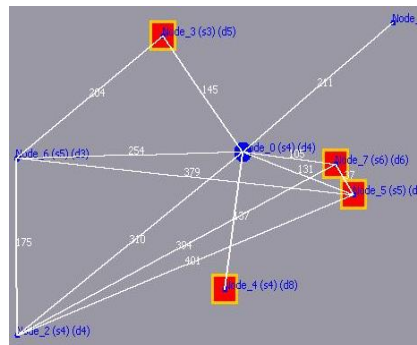
Administrators could consider this solution while not the optimum, as an appropriate cheaper alternative should financial restrictions be a factor. This candidate sustains low bridging centrality (88% decrease), and average minimum path length (13.99% decrease), noting an increase to degree centrality by 12.5% from 0.38 to 0.42, these scores are acceptable as evidenced by the candidate's low robustness score.

5.1.4.3 Tabu Search Optimisation Evaluation

Applying the Tabu Search Optimisation algorithm (TABU) to the primary network (Figure 5.3), random evolvments were made against the original network which resulted in new solutions being generated. The network was evolved into a set of best solutions as described in Section 4.6.4.3, with the final generated evolvment being Figure 5.12.

This reported candidate is the only evolved candidate measured as improved, and has been identified as the optimum solution found using TABU. This undirected graph is the only configuration returned from a single run of the TABU algorithm, which took 9 seconds for completion. No other candidates were returned as the algorithm will not consider an inadequate solution and only improved solutions are developed further. Comparison parameters introduced, influence the tabu list after each cycle, preventing reverse evolvments from being considered to ensure quick and non-costly optimisation.

The strict tabu list which must be observed also limits the process, as, if any parameter is considered tabu even if it is only marginally different, this prevents the candidates from being reported and considered as strong secure solutions.



(a) Optimal Evolution

Figure 5.12. Comparison of Tabu Algorithms Improved Solutions

The tool searches for an optimal secure combination, while the process removes and replaces links. This single reported candidate shows an increase in the number of established links, in an attempt to assure communication security. The robustness monitor failed entirely, as there are no other improved reported solutions for the robustness progression to be mapped. The robustness level of the original network is 700.6233 while the final optimal solution scores a robustness score of 250.4453, achieving a 64.25% improvement. This candidate is a more appropriate solution than the original network topology.

Through the improved robustness, we succeed in maximising security, while sustaining low bridging centrality (92.19% decrease) and average minimum path length (11.93% decrease). This solution notes degree centrality is increased by 87.5 % from 0.38 to 0.71, with all centralities being identified as having acceptable increases and decreases as evidenced by the candidate's low robustness score. However, individual centrality scores should still be analysed as they could be a major factor to consider, depending on the requirements of the network and the perceived accepted level of risk. To achieve this, there has been a significant increase to overall network cost, which rose from 1821 to 2883, meaning there is a cost increase of 58.32% to assure a 92.31% improvement for communications security.

5.1.4.4 Network Security Enhancement and Risk Mitigation Evaluation

Using the SeCurity Risk Analysis and Mitigation Framework we have successfully conducted a series of simulations using the three presented evolutionary security risk mitigation and optimisation algorithms. While it is difficult to compare the three algorithms' effective performance, in the sense

that we can run the robustness algorithm on the same network multiple times and due to the random evolution of the network can be presented with different results each time, we provide a comparative analysis of their effectiveness in a broader sense.

In this instance we have based the experiment on the scenario outlined in Section 4.2, and assume a random 30% connected network comprising of 8 static nodes. The network was established within the tools environment and the three different algorithms described above were applied to the SoS consecutively.

All the algorithms work effectively, but it is immediately noticeable that the TABU algorithm generated the fewest improved solutions, in comparison to the other two algorithms. While TABU ensures a quick and non-costly optimisation process, completing its run in less than half the time when compared to GA and ANT, it fails to consider any solution unduly impacting centralities for example. Due to its restricted comparison parameters that must be matched or improved, even if the solution improves security while unduly impacting centralities then it is considered inadequate and discarded. The tabu list successfully influences cycles preventing reverse evolution from being considered, and this directly improves the overall processing time and costs. But as we analyse results for the other two algorithms we note that alternative reported network configurations potentially could provide security managers with alternative solutions for implementation.

The GA and ANT processes generated multiple improved candidates, and allow for us to not only see the optimum solution but detail multiple other evolved networks that improve the overall security and robustness of the network. This assists in identifying alternative solutions for example that while they might not be quantified as the optimum, could be applied to the network, increasing network communication security, mitigating risk, and improving robustness to a degree, but perhaps cost less to implement than the final optimum candidate. This is highly beneficial when forced to consider and adhere to tight financial requirements and budgets.

The SCRAM security risk mitigation process removed and replaced links during evolution stages searching for an optimal combination. In all three simulations, we see a notable increase in established links between the original evolution and the first candidate, with the evolutions fluctuating between an increase and decrease in links until the last configurations are approached. While this high connectivity among nodes is considered secure in terms of centrality and data flow because of the variety of alternative paths from one node to another, the significant cost increase of the network reflects this increase of new links to assure communication security, i.e. highly connected networks typically result in higher costs. While TABU shows only a single increase in the cost, GA and ANT show while there is a large increase in network cost over all mutated evolutions, the optimum candidates for both solutions are not the most expensive to implement (see Figure 5.13-a).

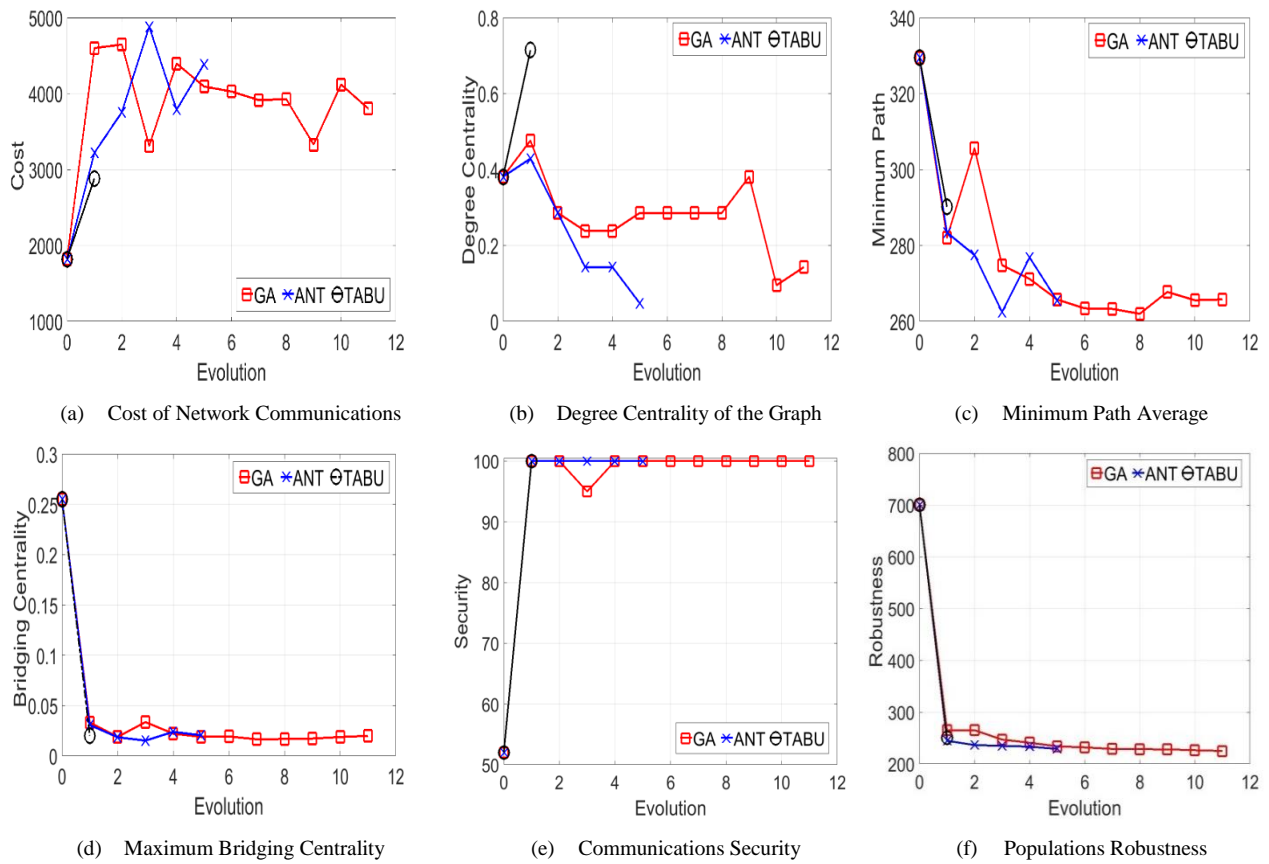


Figure 5.13. Comparison of the Applied Risk Mitigation Algorithms

Networks established with a large number of links are not necessarily establishing a higher security level; other configurations potentially could be more affordable (low cost) and sustain a similar security level. In order to reduce cost, a reduction in the number of connections and path lengths among the nodes inside the network would be expected. In this scenario however, a reduction in links is not viable as the primary network has only 52% communication security, and further link reduction would negatively impact security.

All algorithms show a significant security increase from the first mutated evolvment (Figure 5.13-e), and all apart from GA maintain 100% communication security for each reported solution. GA does maintain high security and only one evolvment drops to 95% secure (Evolvment 3), which is still an 82.69% increase comparable to the original network’s security level of 52%.

As stated due to the random mutations of evolvment it is difficult to compare analysis, and when we look at degree centrality (Figure 5.13-b), for all three generations we see that TABU is the only algorithm that increased degree centrality by 87.5% from 0.38095 to 0.71429. The evolutions of GA and ANT show a significant reduction in degree centrality within each new evolvment. GA reduces degree centrality by 62.5% from 0.38095 to 0.14286, and ANT reduces degree centrality by 87.5% from 0.38095 to 0.04762.

Analysing minimum path average (Figure 5.13-c) and maximum bridging centrality (Figure 5.13-d), we note that all three algorithms report a vast reduction in both areas. Minimum path average is decreased 11.93% by TABU, 19.34% by GA, and 19.43% by ANT, and bridging centrality was decreased by 92.19% via TABU, 92.22% by GA, and 92% by ANT respectively. The overall robustness (Figure 5.13-f) for the reported candidates is also significantly reduced by all three algorithms. Reducing by 64.25% via TABU, 67.89% by GA, and 67.33% by ANT, this means that the optimum candidate for each algorithm is more appropriate and improved due to the significant robustness level decrease compared to the original network.

When we directly analyse the undirected graphs, we see that the combined robustness function with each of the three algorithms, ensures that as the network has been evolved nodes identified as in a ready state and not blocked, have a maintained clear series of links established between them. These evolved links ensure that ready nodes that share the same data level access or higher, have direct communication links between them, rather than data being passed via a node which is considered insecure, i.e. data will not traverse via a node with a lower data level access which should be blocked.

Analysis establishes that for all three algorithms the increase in costs establishes more links, which in turn increases communication security, and as a direct result of the increase in links significantly decreases the minimum path average, while reducing maximum bridging centrality, degree centrality (with the exception of TABU), while vastly improving the population's robustness levels. These results suggest an evolutionary approach is practical for evolving a relatively small network in a small number of steps, and these algorithms can be applied to improve the level of security and mitigate risk in a network, while considering a number of factors such as violation of access control requirements, high centrality node risks, and costs associated with distance between nodes. In addition, while TABU completes its evolutionary process in only 9 seconds and both GA and ANT execute in similar times, there is little difference in the maximum CPU usage and memory used for all three applied methods evident in Table 5.4.

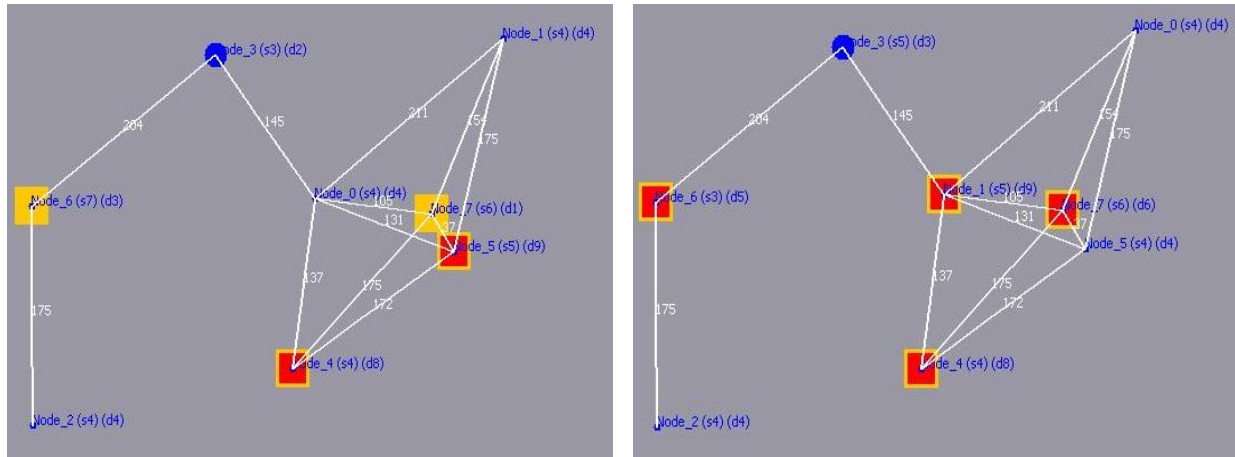
Table 5.4. SCRAM Resource Usage for Applied Algorithms

Algorithm	Number of cycles	Time Taken for Security Risk Mitigation Process	Maximum CPU Time	Maximum Heap Memory Used
GA	2000	00:21:00	18.3%	7.22%
ANT	2000	00:20:00	16.4%	8.22%
TABU	2000	00:09:00	18.3%	7.55%

5.1.5 Evaluating Dynamic Systems-of-Systems

To examine the effectiveness of the techniques implemented within SCRAM, we modified the primary network into two new different forms. First we wanted to examine the effects of altering three

different nodes' data access levels and node security grades, simulating what would occur if nodes that have remained in situ were re-categorised replicating the dynamic nature of SoS. Secondly, we wanted to keep the same topology but establish the network from a different node starting point, replicating communication failure between nodes and network reestablishment, due to a network update. With nodes being re-categorised and communication links restored, i.e. as part of the update security and data access levels have been re-quantified, meaning several nodes have become isolated and cut off from previously secure communication routes within the network.



(a) Network A Reconfigured Security and Data Access Levels

(b) Network B Network Failure and Reconfiguration

Figure 5.14. Modified Topologies of the Primary Network

The first modified network as visualised in Figure 5.14-a, shows the alterations made to the network's node data access levels and security grades. We have reassigned node 3 and 7 so that they have higher data access levels, increasing node 3 from access level 5 to 2 and node 7 from access level 6 to 1. This identifies that node 3 is now ready and secure and node 7 is only insecure. We also re-assigned the security grade for node 6, from 5 to 7, meaning that node 6 is identified as insecure, which directly impedes communication to node 2 making this node isolated. Simulating if node 2 now wished to communicate across the network it would have to transmit data via an insecure node.

Our second network as visualised in Figure 5.14-b, shows the changes made to the network's starting point and changes to several node data access levels and security grades. While we have kept the network's topology the same (i.e. replicating a network configured with static nodes), we have swapped the assignment of node 0 and 1, this simulates that the network terminated and was re-established by node 1, which consequently becomes node 0 (the primary node) and the new starting position for the network. Node 3 and 5 have been re-assigned higher data access levels, increasing node 3, from 5 to 3, and node 5, from 9 to 4. Node 5 also had an increase in security from grade 5 to 4, this means both nodes are now considered ready and secure, as they have an equal or higher data level access than the network's assigned data level access which is 4, and a higher security grade of 5 or higher. Node 1 and 6 have been re-assigned with lower data access levels, decreasing node 1, from 4 to 9, and node 6, from 3 to 5. This designates node 1 and 6 as blocked due to them having a lower

data access level than the network’s assigned level, and node 1 has been re-assigned with a security grade of 5 instead of 4.

Due to the alterations made to these nodes, it has allowed for us to isolate parts of the network from each other by ensuring nodes have blocked communication paths. As visualised in Figure 5.14-b, nodes 2 and 3 have been completely isolated from the remaining two ready nodes (nodes 0 and 5), owing to the topology of the network.

Individually the modified networks were imported into SCRAM and the three outlined algorithms were applied to the networks consecutively. Figure 5.15 and Figure 5.16 visualise the final optimum solutions for each algorithm, and demonstrate the effectiveness of the security risk mitigation processes. This is achieved by displaying every possible secure and vulnerable connection in which data with a security grade that is equal to or higher than the networks assigned data level can traverse. Green lines indicate no data access violations or vulnerabilities, yellow lines indicate potential vulnerabilities but no data access violations, and lines remaining uncoloured indicate potential data access violations and security vulnerabilities (i.e. communication links to blocked nodes).

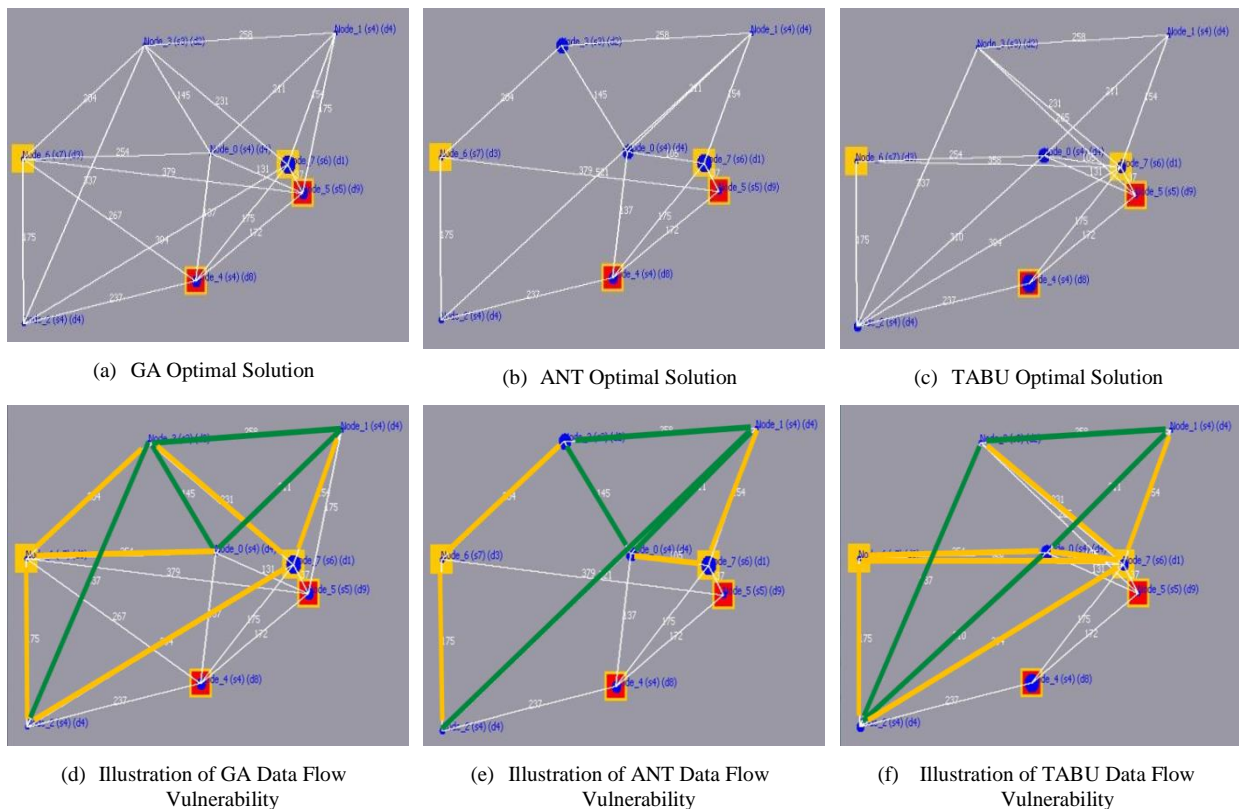


Figure 5.15. Evaluating Modified Network A Reconfiguration

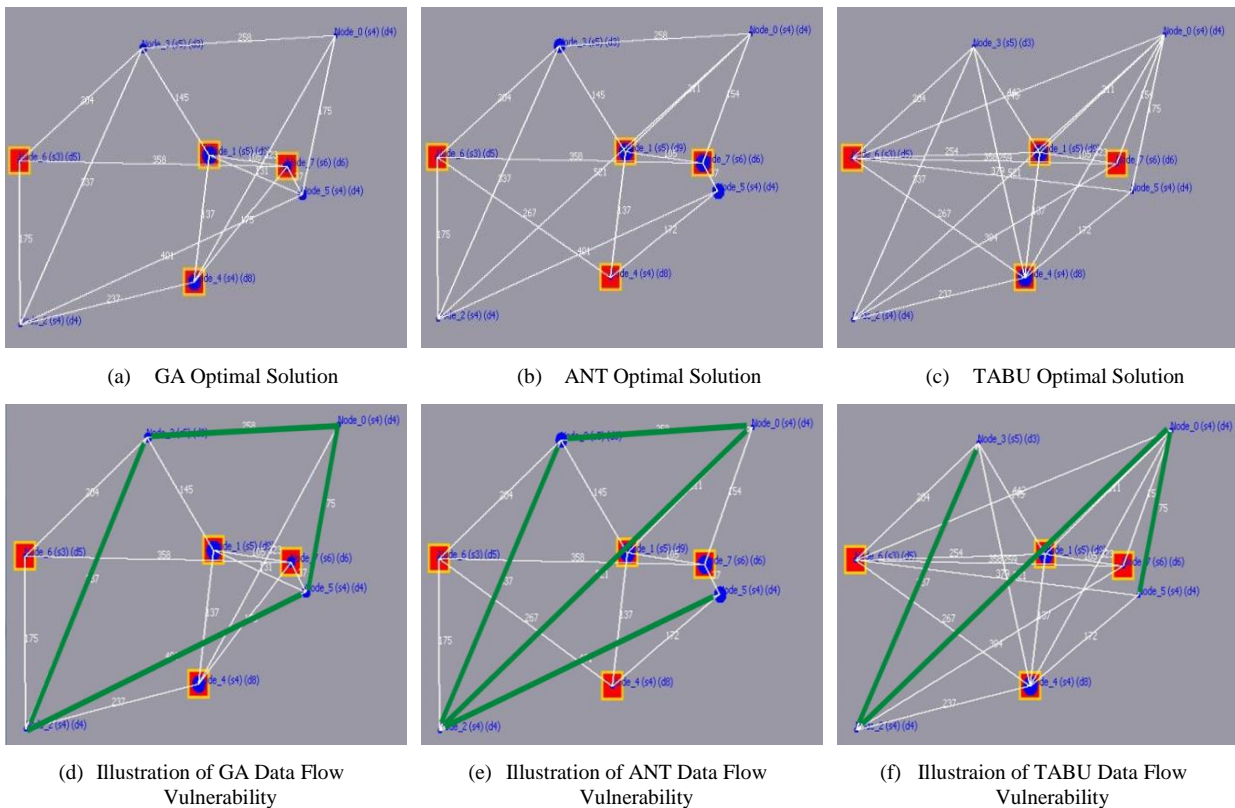


Figure 5.16. Evaluating Modified Network B Failure and Reconfiguration

After conducting this series of simulations, once again we immediately notice the lack of generated alternative solutions via the use of TABU. While TABU on average performed 41.77% faster than the other two algorithms, it failed to consider many alternative solutions that potentially could have increased security, mitigated risk, and improved robustness levels, regardless of altering centrality factors. This quick and low cost solution is failing to provide us with alternative candidates to not only analyse, but if applied to a real network would fail in ensuring that alternative solutions were reported and considered.

For example, the TABU simulations for both networks show a significant increase in communication cost for the optimum candidate, decision makers could fail to implement the new alternative solution based on this report due to no alternatives being presented and concerns over the increasing costs. Should the network remain unaltered, then security would remain at 69% secure for network A (Figure 5.14-a) and 47% secure for network B (Figure 5.14-b). Had a cheaper alternative been reported, then while not being considered the optimum, it could have provided a more secure solution, mitigated risk, and reduced robustness scores while marginally modifying centrality factors further.

The GA and ANT processes generated and reported the optimum solution and detailed multiple improved candidates. Assisting in the identification of alternative solutions, while not considered optimum, indicates improvements to SoS security and robustness.

As illustrated in Figure 5.15 and Figure 5.16, during the security risk mitigation process links were removed and replaced throughout the evolutionary stages. Again there is a notable increase in established links when we compare the original network topology to the final candidate. While this high connectivity among nodes is considered secure, the cost increase of both networks (Figure 5.17-a and Figure 5.18-a) reflects this increase of new links to assure security. In both the GA and ANT simulations we note while there is a significant increase in network cost for all evolved solutions, the optimum candidate for both solutions is not the most expensive to implement.

While security (Figure 5.17-e and Figure 5.18-e) has increased by 44.93% for network A and 112.77% for network B, the graphs highlight some problematic issues for both networks.

Network A - For network A candidates Figure 5.15-d and Figure 5.15-e, secure data flow is highlighted with thick green lines, identifying every possible secure connection in which data with a security level of 4 can traverse between secure nodes. Thick yellow lines on the graph identify those links which have potential vulnerabilities and have the potential to cause risks to the network and data during transfer (i.e. communications links joining nodes identified as insecure and vulnerable). While unaltered white lines indicate potential data access violations (i.e. communication links joining blocked nodes). Figure 5.15-d shows there is a single link between secure nodes 2 and 3, whereas Figure 5.15-e shows a single link between secure nodes 1 and 2. In either network should the single link between these identified nodes fail or be removed, then communication would fail between node 2 and the rest of the network cutting it off from the SoS. However, unlike the topology of network B, network A has alternative paths that would ensure that node 2 remained connected to the remainder of the network via the identified insecure nodes. While for security these links are not ideal, in an emergency it would allow for insecure nodes to be utilised to prevent network failure or interruptions to data communications.

As the framework has identified numerous nodes as insecure, communications would be routed via the secure links, in an endeavour to assure data communication security. Then again, early identification of insecure nodes via the SeCurity Risk Analysis and Mitigation Framework outlining the vulnerabilities of nodes, would allow for early intervention to manage risks and rectify node insecurity. Subsequently, depending on time frames and vulnerability factors, identified insecure nodes potentially could be made safe and re-categorised as secure, allowing for these essential links to then support secure data access.

Viewing the topology as a whole, we can identify that while there is limited secure paths, alternative communication links via non-blocked nodes are available despite being assigned as insecure. As while the nodes have been deemed insecure they do have a data access level of 4 or above. These alternative links exist as the network is formed with only a minority of nodes not having a sufficient data access grade (i.e. the network only consists of two blocked nodes).

Network B - In network B optimum candidates Figure 5.16-d and Figure 5.16-e demonstrates that there is an established secure data flow, with alternative routes between secure and ready nodes. Green lines indicate the secure data paths between nodes, which data with a security level of 4 or above can traverse. With unaltered white lines indicating potential data access violations (i.e. communication links joining blocked nodes). In Figure 5.16-e we do note there is a significant SPoF, as, if node 2 was to fail or be removed, node 5 would be cut off from the secure data route, or could be forced to send data via nodes in breach of data access requirements.

Figure 5.16-f indicates there is a single path between nodes 0, 2, 3, and 5, should any individual link or node be removed or fail, then communication between these nodes will cease as there is no alternative communication paths. In addition, nodes 0, 2, 3, and 5 could be categorised as SPoF.

When we view the topology as a whole, we quickly can identify that this issue is partly due to the fact that the network is formed via eight nodes and half of these are identified as insecure, due to their data access level being lower than the network data access level of 4. Also, the nodes' positions and the topology of the network play an influential role during evolution. Therefore, the tool is restricted evolving the network via the remaining four secure nodes, and attempts to balance cost with improvements to the network's robustness and security, while not unduly impacting centrality factors.

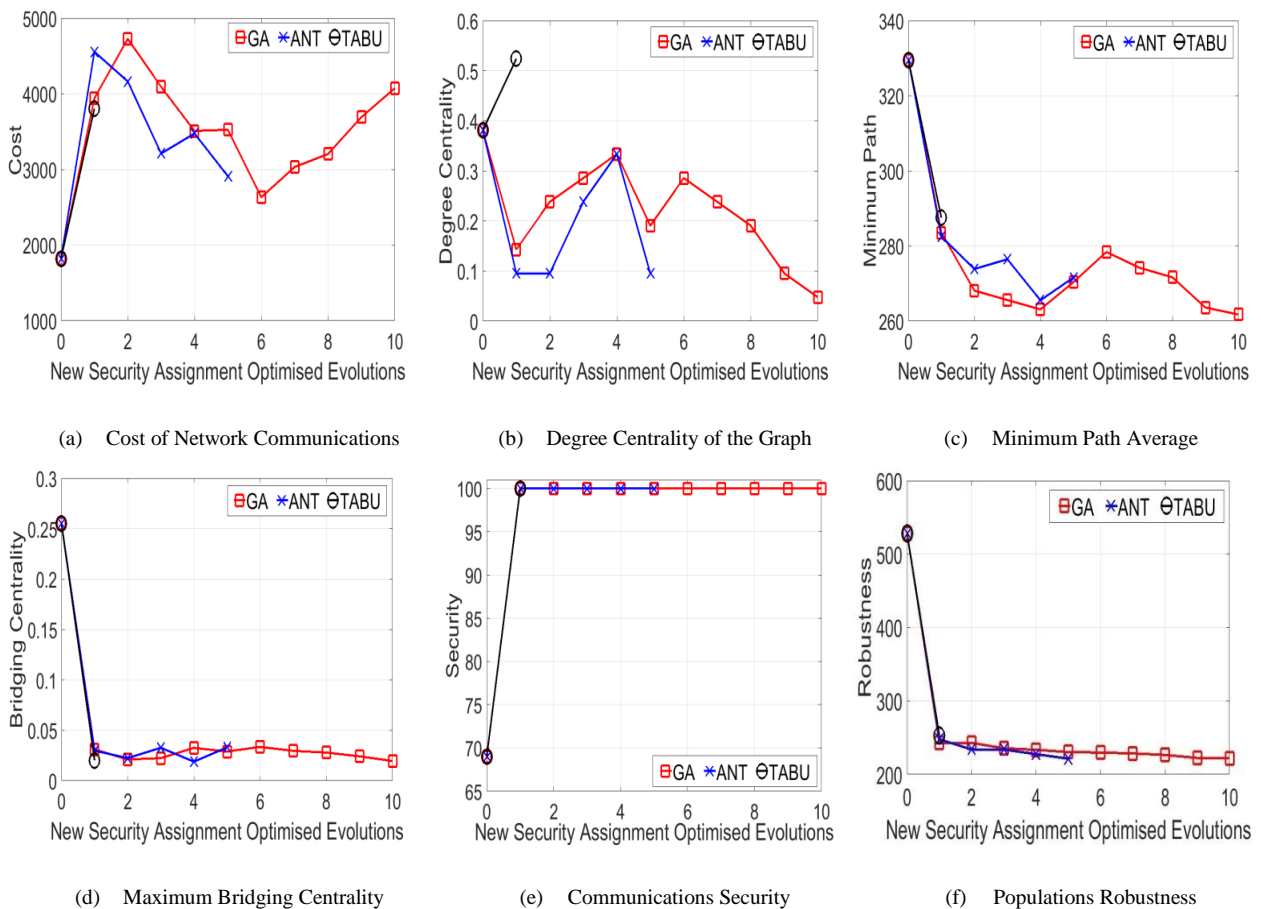


Figure 5.17. Evolution Analysis of Modified Primary Network Reconfiguration

The combined process of the robustness and security risk mitigation algorithms, ensure that node status (i.e. ready, insecure, and blocked) and evolvments of the deleted and replaced links are considered when quantifying the robustness and security of the network. SCRAM mitigates security risks and evolves the network in the belief that new evolved links to secure nodes with the same data level access or higher will be responsible for secure communications, while insecure or blocked nodes will not be traversed.

Both evolved networks A and B establish a higher security level via the development of new alternative links, and it must be noted other configurations potentially could be more affordable and sustain a similar security level compared to the optimum reported solution. In these scenarios a reduction in links is not viable as primary network A only has 69% communication security and network B has 47%. Furthermore, the original network topologies indicate nodes are isolated and cut off from other nodes in the network as they are currently connected via paths routed through insecure and blocked nodes, meaning further link reduction could result in nodes being disconnected from the SoS completely. Additional link reduction would negatively impact security and further impede network communication.

All algorithms show a significant communication security increase from the first mutated evolvment (Figure 5.17-e and Figure 5.18-e), and all maintain 100% communication security for each reported solution, increasing communication security by 44.93% for network A and 112.77% for network B.

Due to the random mutations during evolvment, we see different evolved security enhanced solutions; however, we do notice similarities during analysis. Regarding degree centrality (Figure 5.17-b and Figure 5.18-b) for both networks A and B, analysis shows a significant reduction. GA reduces degree centrality by 87.5% from 0.38095 to 0.04762 for both networks, and ANT reduces degree centrality by 75% from 0.38095 to 0.09524 for network A and 87.5% from 0.38095 to 0.04762 for network B.

TABU was the only algorithm that increased degree centrality by 37.5% for network A, and due to its low yield of improved candidates it is difficult to determine if the algorithm's process is not as adequate compared to its counter solutions or if this anomalous result is due to the randomness of the network's evolvment. TABU did reduce degree centrality by 37.5% from 0.38095 to 0.238095 for Network B.

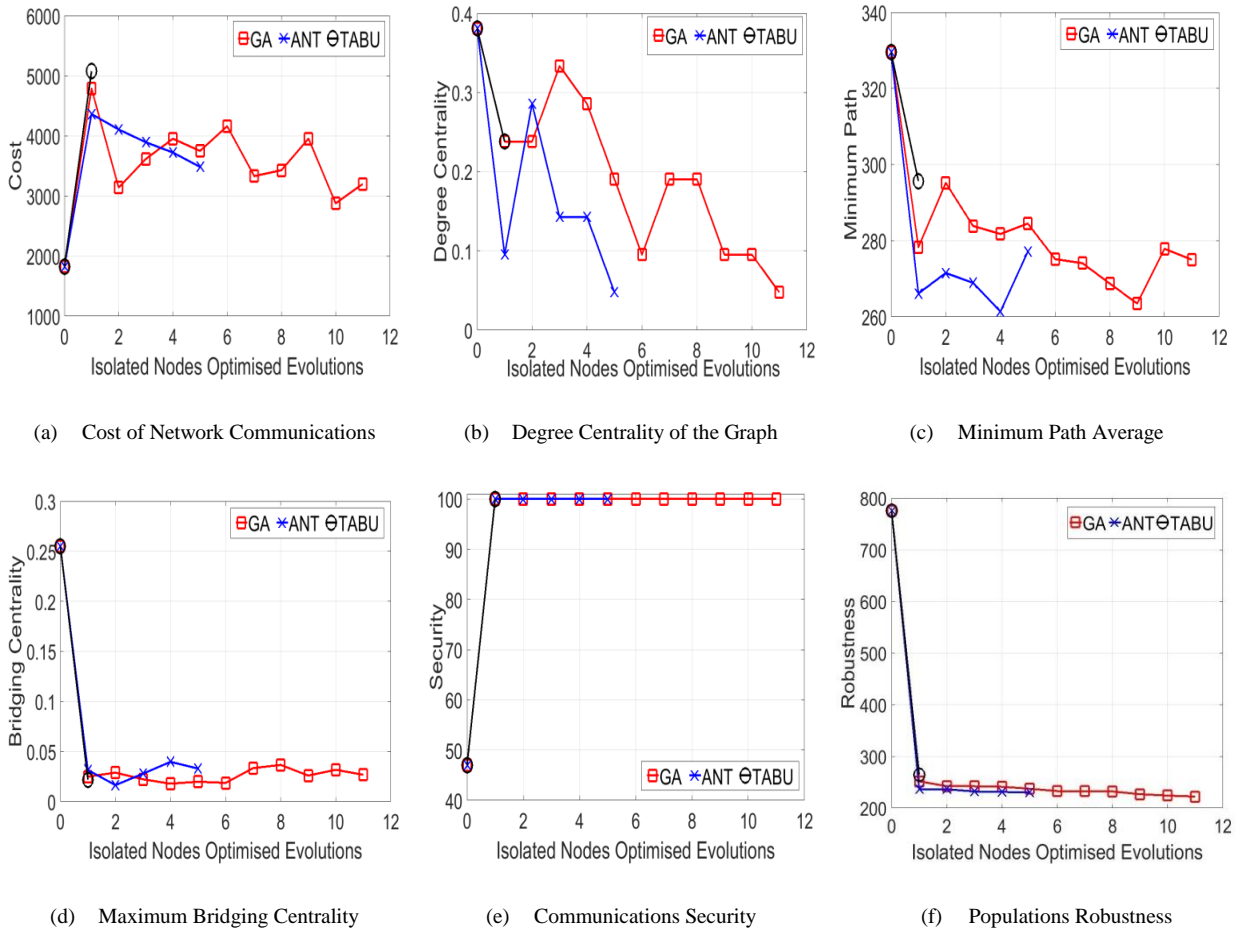


Figure 5.18. Evolution Analysis of Modified Primary Network Failure and Reconfiguration

When we analyse minimum path average (Figure 5.17-c and Figure 5.18-c) and maximum bridging centrality (Figure 5.17-d and Figure 5.18-d) we see a vast reduction in both areas when we applied all three algorithms to both networks. Minimum path average is reduced by 20.56% for network A and 16.54% for network B using GA, 17.58% for network A and 15.9% for network B using ANT, and 12.75% for network A and 10.28% for network B using TABU. Whereas, bridging centrality decreased by 92.36% for network A and 89.5% for network B via GA, by 87.08% for both networks A and B using ANT, and by 92.13% for network A and 91.46% for network B via TABU respectively.

These reductions are reflected in the decreased robustness scores (Figure 5.17-f and Figure 5.18-f) for the optimum candidates, which show GA improving robustness by 57.93% and 71.34% for networks A and B respectively, ANT improves robustness by 58.08% and 70.29% for networks A and B respectively, and TABU reduces robustness by 52.01% and 65.89% for networks A and B respectively. Ensuring that optimum candidates presented by each algorithm, are more appropriate and improved due to the significant robustness score decreases in comparison to the original networks.

In comparison, all algorithms generated optimum candidates with vastly increased costs (Figure 5.17-a and Figure 5.18-a) to communications. Which can be directly attributed to the increase in communication links between nodes which decreased minimum path average, reduced maximum bridging centrality, degree centrality (with the exception of Network A when TABU was applied), and reduced the population robustness of the networks. The results for these simulations are similar to the previously reported simulation results, supporting the evolutionary method's appropriateness for network security risk mitigation focusing on not only the security of the network but also its overall robustness, whilst considering factors such as access control violation, high centrality node risks, and cost.

The principles of SCRAM that report alternative candidates when applying GA and ANT, demonstrate the framework's effectiveness to report cheaper alternative network topologies that will significantly ensure higher security levels with a reduced robustness score, that while not identified as optimum could be suitable alternatives to implement and mitigate any related negative risks. An illustration of this is candidate 6 reported by the GA algorithm for network A. The associated costs with this candidate only increase the network cost by an additional 44.54% which is an increase of 811. This improved solution enhances the network's robustness and does not unduly impede centralities when compared to the original network's parameters, while assuring security at 100%, meaning it would be an excellent alternative to the optimum candidate.

5.1.6 Effectiveness of Simulated SCRAM Framework

Similar to the work of Rullo et al. [232], Yan et al. [226], and Yao et al. [231], we have chosen to use simulation to not only generate our framework, but also our simulated environments and experiments. The use of simulation is an acceptable and realistic approach within the field of SoS research, to ensure that methods are vigorously tested and evaluated prior to their application within these complex and dynamic environments. Simulation can adequately imitate the behaviour and representation of networked infrastructures, can assist to understand the interactions and links between components and systems, and can support the understanding of component and system operations, the complexity of SoS, and interdependencies which form due to collaborative relations.

Initial experiments validate that the SCRAM framework is an effective simulated solution which provides us with the capability to emulate realistic SoS, and provides us with a platform to evaluate the theoretical principles presented in this thesis. The use of simulation ensures that our framework and proposed principles will not impact or introduce additional risks into a physical real world infrastructure. Our review of existing methods and applied solutions for example, corroborates that methodologies such as risk assessment when applied directly to networks, can impact the physical systems and their functionality. Within SoS environments any negative effect or failing could quickly

escalate and cause partial or full cascading failures, which potentially could result in critical consequences.

The developed framework delivers us the means to simulate and test multiple different environments, allowing us to generate a range of diverse and configurable distinct environments which we could not physically develop, and would most likely be unachievable due to access and project limitations. That is, the simulated framework allows us to adapt or scale a variety of distinct simulated infrastructures more effectively than had we established the same physical infrastructures.

The SCRAM framework has given us the flexibility to generate and replicate SoS and multi-level SoS, and ensures that we produce useful and accurate statistics and characteristic configurations to represent diverse infrastructures, key characteristics, and system and network behaviours. SCRAM allows us to tailor the algorithms and proposed principles to our test requirements, ensuring their suitability when applied to collaborative infrastructures. Additionally, simulation provided us with the ability to explore and evaluate the advantages of alternative algorithms, and examine the effects of our design choices. Consequently, the framework's practical ability to generate collaborative infrastructures means that we have the capability to establish the appropriateness of an infrastructure prior to its physical construction, and the means to quantify its design and efficiency via the proposed methods.

When applying the proposed principles and algorithms to the simulated environments, the SCRAM framework allows us to vigorously test and investigate the ramifications of the algorithms and processes without impacting physical topologies, and can provide the means to examine identified issues and limitations at different levels of abstraction. Importantly, the use of simulation has given us a platform to effectively demonstrate and validate our concepts, and has assisted us to perform "what if" analysis cheaply and efficiently.

In general, simulated results are accurate in comparison to analytical models for example. Having generated multiple environments and implemented multiple experiments, to confirm the accuracy of the SCRAM framework and principles, we manually checked and quantified all factors. For example, using node property aspects, identified vulnerabilities, topology data, and other factors, we can manually quantify node security scores, data access violations, communication security, robustness scores, and topological vulnerabilities, etc. These scores are then compared against the simulated infrastructure to determine the accuracy of the SCRAM framework's ability to generate and represent SoS and multi-level SoS environments.

When we apply the different algorithms and processes against the topologies in order to mitigate risk and increase communication security utilising only the existing infrastructure, we manually confirm the appropriateness of the reported optimised solutions and proposed principles' results. Firstly, by examining the alterations made to the reconfigured communication paths, specifically analysing paths

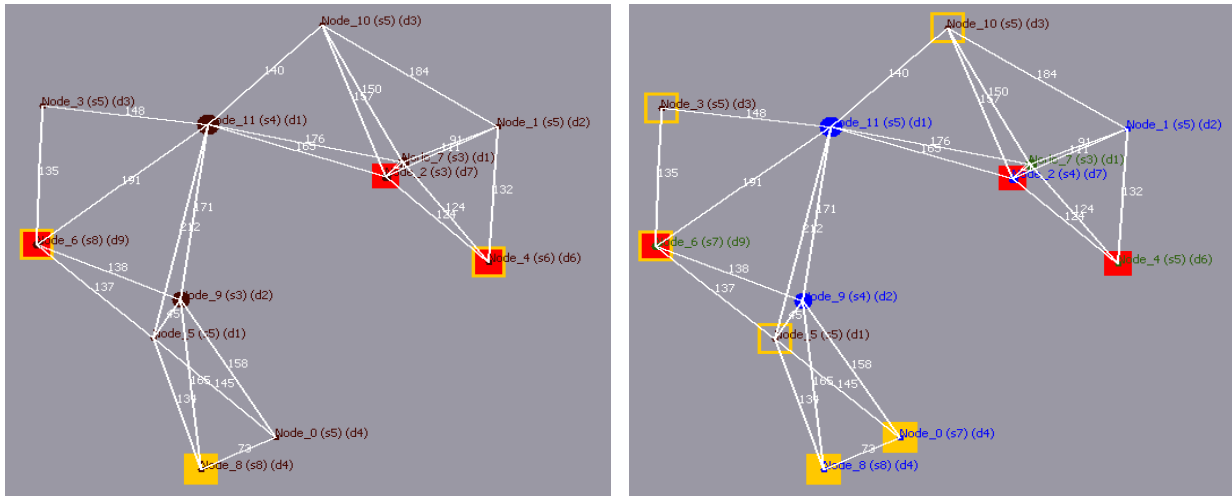
for data access violations and ensuring nodes have not been disconnected from the network. Then we manually quantify topological vulnerabilities, robustness scores, communication security, cost, and minimum path average, etc. This ensures that as we analyse the appropriateness of the risk mitigation processes and algorithms, we not only evaluate the enhanced reported solutions but confirm their accuracy against the manually quantified results, validating the use of simulation, the precision of the simulation processes, and the generated and reported results.

This was essential to prove that SCRAM was not just programmed to imitate results and will only work on a predefined and preprogramed infrastructure; instead that the proposed algorithms and methods are encoded to efficiently mitigate risk and secure infrastructures based upon the designs of the principles, and can be applied to differing distinct SoS and multi-level SoS infrastructures. Manual quantification and comparison of simulated results was also essential, as it allowed us to ensure the method's application and processing design is sufficient, prior to its implementation and application to large and complex infrastructures. In these instances analysis will become less intuitive and more reliant on automated processing, as manual corroboration will become time consuming, impractical, and very likely impossible when we begin to experiment on significantly large multi-level SoS.

5.2 Effectiveness of Integrating Vulnerability Identification

For proof of concept and in order to evaluate our solution, we initially generated a random SoS consisting of eight static nodes and simulated security grades for each node. For node security grades to be accurate it is important that we identify any vulnerability that has the potential to expose the node to risks, which in turn can negatively impede the network's topology which it forms part of. Vulnerabilities can be identified using a vulnerability scanner, allowing for vulnerability scoring and exploit databases to be incorporated into the network's risk assessment methodology. Allowing for risks to not only be documented, but also for these vulnerabilities to be quantified and incorporated as part of the tools security grade assignment, thus improving the accuracy of node security grade scoring.

Building upon the SCRAM framework, we have incorporated this functionality to simulate vulnerability identification and assign reported NVD vulnerabilities to nodes, in a random method based on the device's software, firmware, and hardware. To begin with we generated a random network consisting of 12 static nodes which have one of three operating systems (Linux, Android or Windows), with a connectivity level of 30% (Figure 5.19-a). The primary simulation does not conduct any vulnerability scans upon the nodes within the network, as evident by nodes being visualised in dark red.



(a) SoS Assigned with Security Values (b) SoS Assigned with NVD Vulnerabilities and CVSS v3 Scores

Figure 5.19. Network Comparison of Assigned Security and Simulated Vulnerabilities

The primary network (Figure 5.19-a) was then re-imported into the SCRAM framework (see Figure 5.19-b), each node was then simulated with a randomly assigned vulnerability node status, and provided they were designated as scanned, a random number of associated vulnerabilities from the NVD database were assigned. Table 5.5 demonstrates the visualised graph parameters, while Table 5.6, Table 5.7, and Table 5.8 are example excerpts of vulnerabilities reported via NVD which have been incorporated into the SCRAM framework.

Table 5.5. Visualised Security Graph Vulnerabilities and Parameters

Graph	Parameter	Symbol	Description
All graphs	Scanned node no vulnerabilities.		Dark green node/tag.
	Scanned node unresolved identified vulnerabilities.		Blue node/tag.
	Unscanned node.		Dark red node/tag.
	Node size represents quantified bridging centrality score, i.e. the width of the node is proportion to its bridging centrality value.		
Security	Insecure node.		Node encased with a solid orange box.
	Blocked node.		Node encased with a solid red box.
	Blocked and insecure node.		Node encased with a solid red box with orange border.
	Node quantified secure and unscanned.		Node encased with a non-solid orange box.

Table 5.5 depicts the visualised parameters used to generate the undirected graphs, and key parameters are visualised as follows:

- Nodes that have been scanned with no found vulnerabilities will be visualised with a dark green node and name tag. An example of this is node 4 Figure 5.19-b.
- Nodes that have been scanned with identified unresolved vulnerabilities will be visualised with a blue node and name tag. An example of this is node 1 Figure 5.19-b.
- Nodes that have not been scanned will be visualised using a dark red node and name tag. An example of this is node 10 Figure 5.19-b.
- Nodes assigned a security score of 5 or below that have had their security quantified as secure but have failed to run a vulnerability scan, will be visualised with a non-solid orange box surrounding the node. An example of this is node 10 Figure 5.19-b.
- Insecure nodes are visualised with a solid orange box surrounding the node. An example of this is node 8 Figure 5.19-a.
- Blocked Nodes are visualised with a solid red box surrounding the node. An example of this is node 4 Figure 5.19-b.
- Blocked and insecure nodes are visualised with a solid red box with an orange border surrounding the node. An example of this is node 4 Figure 5.19-a.
- Nodes quantified with higher bridging centralities are represented with wider nodes. An example of this is node 11 in the centre Figure 5.19-a.

Each month hundreds of vulnerabilities are reported and processed via NVD, for these early experiments we did not feel it necessary to program every reported vulnerability that is associated with Windows, Linux, or Android devices into the framework, in order to prove the effectiveness of the principles and algorithms. Instead we captured reported and revised vulnerabilities associated with each of the devices for a specific time frame (3rd June 2016 and 18th June 2016). These reported vulnerabilities taken directly from the NVD website were then programmed into the SCRAM framework, utilising each of the vulnerabilities' unique CVE reference IDs and their CVSS v3 base scores. Tables 5.6, 5.7, and 5.8 provide a small selection of these captured risks to demonstrate the different types of vulnerabilities that are being simulated within each of the generated SoS and multi-level SoS, and the tables provide a description of the vulnerabilities, the threat level of each vulnerability, and their quantified impact and exploitability score as reported by NVD. These vulnerabilities demonstrate some of the risk vectors that we are endeavouring to mitigate and secure within SoS environments.

Table 5.6. Identified NVD Vulnerabilities with CVSS v3 Scores for Android OS Devices

CVE-ID	Original release date	Revised date	Overview	CVSS v3.0 Base Score	Threat level	Impact Score	Exploitability Score
CVE-2015-8950	10/10/16	10/12/16	arch/arm64/mm/dma-mapping.c in the Linux kernel before 4.0.3, as used in the ION subsystem in Android and other products, does not initialize certain data structures, which allows local users to obtain sensitive information from kernel memory by triggering a dma_mmap call.	5.5	Med	3.6	1.8
CVE-2016-3933	10/10/16	10/12/16	mediaserver in Android before 2016-10-05 on Nexus 9 and Pixel C devices allows attackers to gain privileges via a crafted application, aka internal bug 29421408.	7.8	High	5.9	1.8
CVE-2016-5348	10/10/16	10/12/16	The GPS component in Android 4.x before 4.4.4, 5.0.x before 5.0.2, 5.1.x before 5.1.1, 6.x before 2016-10-01, and 7.0 before 2016-10-01 allows man-in-the-middle attackers to cause a denial of service (memory consumption, and device hang or reboot) via a large xtra.bin or xtra2.bin file on a spoofed Qualcomm gpsonextra.net or izatcloud.net host, aka internal bug 29555864.	5.9	Med	3.6	2.2
CVE-2016-6674	10/10/16	10/11/16	system_server in Android before 2016-10-05 on Nexus devices allows attackers to gain privileges via a crafted application, aka internal bug 30445380.	7.8	High	5.9	1.8
CVE-2016-6677	10/10/16	10/11/16	The NVIDIA GPU driver in Android before 2016-10-05 on Nexus 9 devices allows attackers to obtain sensitive information via a crafted application, aka internal bug 30259955.	5.5	Med	3.6	1.8
CVE-2016-6684	10/10/16	10/11/16	The kernel in Android before 2016-10-05 on Nexus 5, Nexus 5X, Nexus 6, Nexus 6P, Nexus 9, Nexus Player, and Android One devices allows attackers to obtain sensitive information via a crafted application, aka internal bug 30148243.	5.5	Med	3.6	1.8
CVE-2016-6687	10/10/16	10/11/16	The NVIDIA profiler in Android before 2016-10-05 on Nexus 9 devices allows attackers to obtain sensitive information via a crafted application, aka internal bug 30162222.	5.5	Med	3.6	1.8

Source: Nvd.nist.gov [92].

Table 5.7. Identified NVD Vulnerabilities with CVSS v3 Scores for Windows OS Devices

CVE-ID	Original release date	Revised date	Overview	CVSS v3.0 Base Score	Threat level	Impact Score	Exploitability Score
CVE-2016-3267	10/13/16	10/14/16	Microsoft Internet Explorer 9 through 11 and Microsoft Edge allow remote attackers to determine the existence of unspecified files via a crafted web site, aka "Microsoft Browser Information Disclosure Vulnerability."	5.3	Med	3.6	1.6
CVE-2016-3331	10/13/16	10/14/16	Microsoft Internet Explorer 11 and Microsoft Edge allow remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted web site, aka "Microsoft Browser Memory Corruption Vulnerability."	7.5	High	5.9	1.6
CVE-2016-3382	10/13/16	10/14/16	The scripting engines in Microsoft Internet Explorer 9 through 11 and Microsoft Edge allow remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted web site, as demonstrated by the Chakra JavaScript engine, aka "Scripting Engine Memory Corruption Vulnerability."	7.5	High	5.9	1.6
CVE-2016-3391	10/13/16	10/17/16	Microsoft Internet Explorer 10 and 11 and Microsoft Edge allow context-dependent attackers to discover credentials by leveraging access to a memory dump, aka "Microsoft Browser Information Disclosure Vulnerability."	5.3	Med	3.6	1.6
CVE-2016-6992	10/13/16	10/17/16	Adobe Flash Player before 18.0.0.382 and 19.x through 23.x before 23.0.0.185 on Windows and OS X and before 11.2.202.637 on Linux allows attackers to execute arbitrary code by leveraging an unspecified "type confusion."	9.8	Critical	5.9	3.9
CVE-2016-6999	10/13/16	10/14/16	Integer overflow in Adobe Reader and Acrobat before 11.0.18, Acrobat and Acrobat Reader DC Classic before 15.006.30243, and Acrobat and Acrobat Reader DC Continuous before 15.020.20039 on Windows and OS X allows attackers to execute arbitrary code via unspecified vectors.	9.8	Critical	5.9	3.9

Source: Nvd.nist.gov [92].

Table 5.8. Identified NVD Vulnerabilities with CVSS v3 Scores for Linux OS Devices

CVE-ID	Original release date	Revised date	Overview	CVSS v3.0 Base Score	Threat level	Impact Score	Exploitability Score
CVE-2016-5995	09/30/16	10/03/16	Untrusted search path vulnerability in IBM DB2 9.7 through FP11, 10.1 through FP5, 10.5 before FP8, and 11.1 GA on Linux, AIX, and HP-UX allows local users to gain privileges via a Trojan horse library that is accessed by a setuid or setgid program.	7.3	High	5.9	1.3
CVE-2016-6992	10/13/16	10/17/16	Adobe Flash Player before 18.0.0.382 and 19.x through 23.x before 23.0.0.185 on Windows and OS X and before 11.2.202.637 on Linux allows attackers to execute arbitrary code by leveraging an unspecified "type confusion."	9.8	Critical	5.9	3.9
CVE-2016-7039	10/16/16	10/18/16	The IP stack in the Linux kernel through 4.8.2 allows remote attackers to cause a denial of service (stack consumption and panic) or possibly have unspecified other impact by triggering use of the GRO path for large crafted packets, as demonstrated by packets that contain only VLAN headers, a related issue to CVE-2016-8666.	7.5	High	3.6	3.9
CVE-2016-8658	10/16/16	10/18/16	Stack-based buffer overflow in the brcmf_cfg80211_start_ap function in drivers/net/wireless/broadcom/brcm80211/brcmfmac/cfg80211.c in the Linux kernel before 4.7.5 allows local users to cause a denial of service (system crash) or possibly have unspecified other impact via a long SSID Information Element in a command to a Netlink socket.	6.1	Med	4.2	1.8
CVE-2016-8666	10/16/16	10/18/16	The IP stack in the Linux kernel before 4.6 allows remote attackers to cause a denial of service (stack consumption and panic) or possibly have unspecified other impact by triggering use of the GRO path for packets with tunnel stacking, as demonstrated by interleaved IPv4 headers and GRE headers, a related issue to CVE-2016-7039.	7.5	High	3.6	3.9

Source: Nvd.nist.gov [92].

When we compare the networks in Figure 5.19, we notice significant differences in node statuses related to their quantified security grades. For example, without the simulated node scan, node 0 had a security level of 5. Once the vulnerability scan was simulated, the node was identified as having vulnerabilities that significantly altered its security grade from 5 to 7. This node is now considered insecure and data transfer via links connected to the node would now be restricted. Node 4 has also been reassigned as blocked instead of its previous assignment of blocked and insecure, after being identified as scanned with no vulnerabilities, and its quantified security grade decreased from 6 to 5.

In addition, nodes 3, 5 and 10 have been identified by the tool as having a security score of 5 which is quantified as secure, yet ascertains these nodes have failed to run a vulnerability scan. Consequently these nodes have been visualised with a yellow non-solid box surrounding each node, as we can't truly ascertain if these nodes contain vulnerabilities which could expose the network, nor rely on the accuracy of the security grade. The graph generated by SCRAM determines node 2, 9 and 11 have all had their security grades increased by one, and node 6 has dropped by a single grade, these changes to their grades however in this simulation did not alter their node standings. Also, the graph visualises the changes to nodes 1, 9 and 11, these nodes have been quantified as secure and have been represented via blue nodes and name tags. This is due to them having vulnerabilities found during the simulated scan that potentially could expose them and the network to risks.

The added functionality within the tool to identify vulnerabilities which have the potential to expose nodes and the network to risks ensures that we have the capability to generate more accurate security

grades and produce more detailed and accurate graphs. Evident via the re-categorisation of node security grades when we compare network Figure 5.19-a against network Figure 5.19-b. These accurate grades are visualised as part of the tool's framework, allowing us to intuitively analyse the networks for threats, as demonstrated via the conducted simulations in Figure 5.20.

Table 5.9. Excerpt from SCRAM Security Report for Network Figure 5.19-b

Node ID	Firewall	IDS	Encryption Type	Staff Level	OS Type	Anti-Virus	Internet Access	Vulnerabilities identified from NVD
0	False 10	True 1	RC2-128 5	Medium 5	Android 3	True 1	True 10	CVE-2016-6677 - 6 CVE-2016-6674 - 8 CVE-2016-3933 - 8
1	True 1	False 10	WEP-114 7	Low 10	Android 3	True 1	True 10	CVE-2015-8950 - 6
2	True 1	True 1	RC2-128 5	Medium 5	Linux 1	True 1	False 1	CVE-2016-5995 - 7 CVE-2016-7039 - 8
3	True 1	False 10	RC2-128 5	Low 10	Linux 1	True 1	True 10	Not scanned 10
4	False 10	True 1	RC2-128 5	Medium 5	Windows 5	True 1	True 10	Scanned vulnerabilities 0
5	False 10	True 1	RC2-128 5	Low 10	Windows 5	True 1	True 10	Not scanned 10
6	True 1	True 1	None 10	Medium 5	Windows 5	False 10	True 10	Scanned vulnerabilities 0
7	False 10	True 1	AES-256 1	Medium 5	Windows 5	True 1	False 1	Scanned vulnerabilities 0
8	True 1	False 10	None 10	Medium 5	Android 3	False 10	True 10	CVE-2016-3933 - 8 CVE-2015-8950 - 6
9	True 1	True 1	AES-256 1	Low 10	Linux 1	True 1	True 10	CVE-2016-5995 - 7
10	True 1	False 10	TDES-168 2	Low 10	Android 3	True 1	True 10	Not scanned 10
11	False 10	True 1	TDES-168 2	High 1	Linux 1	False 10	False 1	CVE-2016-6992 - 10

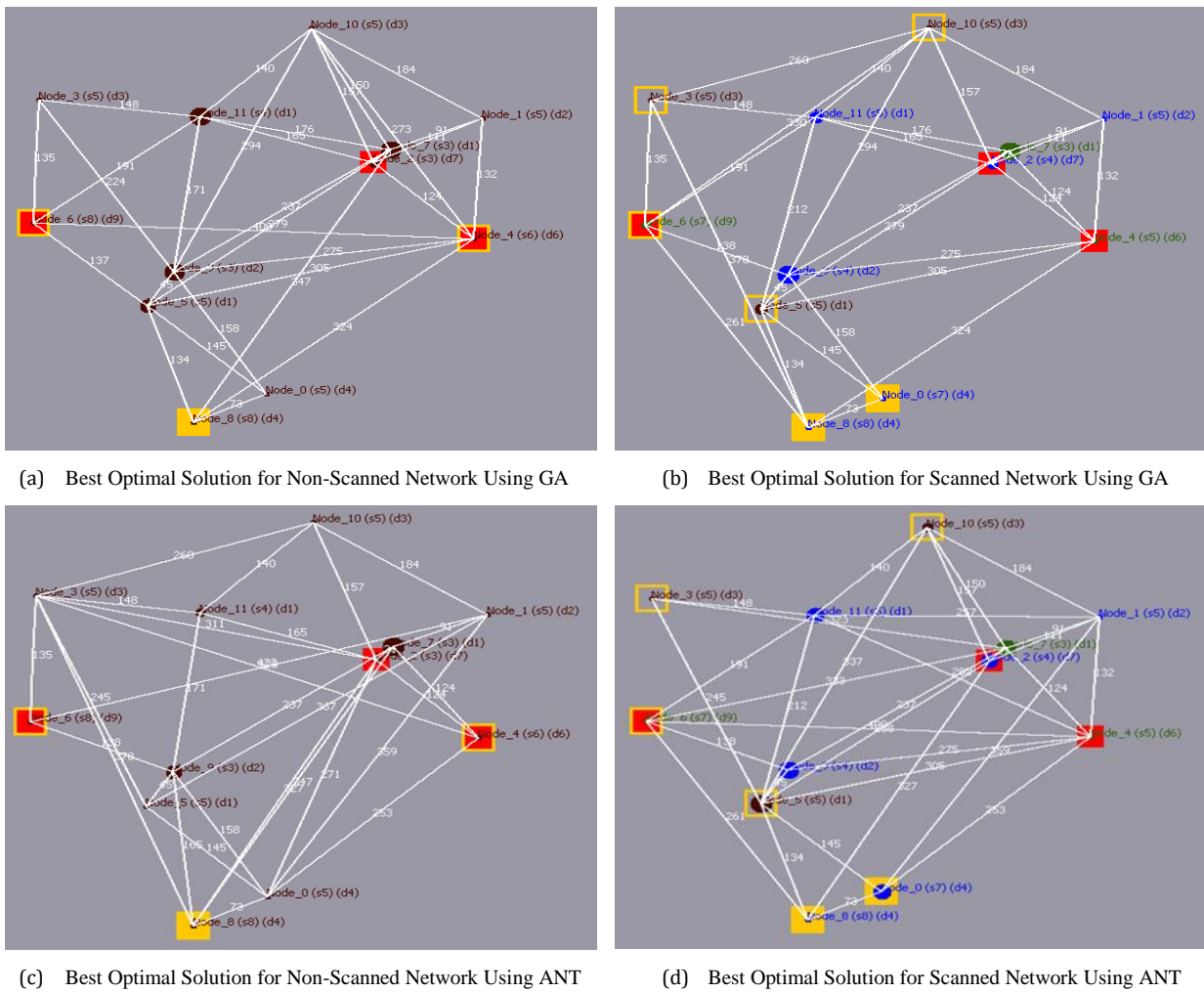


Figure 5.20. Evaluating Network Vulnerability Identification

In addition, SCRAM not only simulates the scan and provides new quantified security scores and visualisation; it also generates a detailed report on all security parameters including security and vulnerability scores, and identified vulnerabilities with their associated CVSS v3 base scores (see excerpt in Table 5.9).

Focusing upon the application of GA and ANT within our framework due to their encouraging yield of alternative reconfigured and security enhanced solutions, Figure 5.20 visualises the final optimum solutions for each algorithm applied to both the non-scanned network with assigned vulnerabilities and scanned network with NVD security vulnerabilities and their associated CVSS v3 scores. Both algorithms reported a series of optimised solutions detailing alternative improved candidates, and while these evolved networks are not considered optimum they demonstrate improvements to both the security and robustness of the SoS.

In comparison to the original network's topology (Figure 5.19) there is a noticeable increase in established links generated for all four final optimal candidates (Figure 5.20). For both the scanned and unscanned networks, when GA was applied to mitigate risk and evolve the network there was a 25% increase in the number of new communication links, and when ANT was applied there was a

20.83% increase in the number of additional communication links. The cost increase for all networks (Figure 5.21-a) reflects this growth of communication paths.

When we analyse the GA optimal solutions for both networks, it is noted that this 25% increase in links increases communication costs for both networks by 67.7% and security by an average 18.56%, while reducing robustness scores by an average of 20.48%. The ANT optimal solutions for both networks increases communication costs by 86.29% for the unscanned network and 93.99% for the scanned network supporting the 20.83% link increase for both optimal candidates, while increasing security by an average of 19.77% and decreasing robustness scores by an average of 17.17%. Comparable to previous simulations, these four reported optimal solutions are not the most expensive or cheapest solution to implement. It must also be noted that while these new links will support node connectivity and increase security, they also introduce additional risk factors.

Viewing the topology of all four optimal candidates (Figure 5.20) we can intuitively identify that there is a number of prime links between secure (ready) nodes which assure secure data flow across the network, and recognise the paths that are linked to nodes that have the potential to cause data access violations or expose data, nodes, and the network to risks. It is also evident that there are multiple alternative links between nodes, meaning should a single node or secure link within the topology fail or be removed, then there is no single point of failure and communication(s) can be routed via alternative secure paths across the topology. In the event of link failure it is unlikely that a node will become isolated or cut off from the remainder of the network.

It is essential that the security risk mitigation process when adding and removing links between nodes, balances connectivity with improvements to the network's robustness and security, while unduly impacting centrality factors. The framework is not attempting to revise cost, simply it is attempting to associate the network's cost with recommended network modifications.

Analysing all of the enhanced reported candidates, it is evident that link reduction for this network's topology would negatively impact security and impede network communication, and all reported candidates guarantee improved security from the first mutated evolution. Based on the increased number of nodes in our current topology, the increased number of communication links, along with functionality changes to the framework when quantifying node security grades and vulnerability identification, we observe a larger volume of communication security fluctuations between reconfigured candidates.

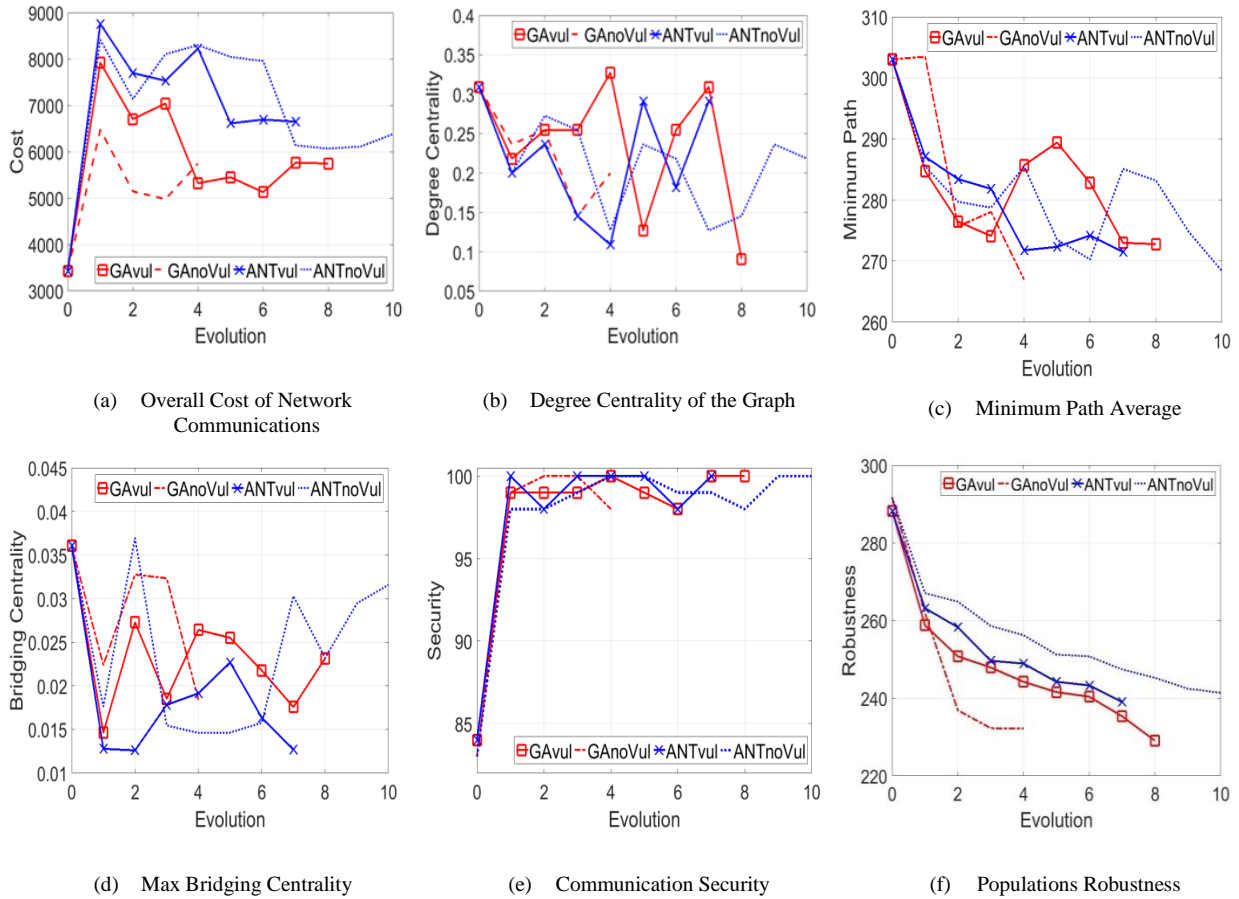


Figure 5.21. Evolution Analysis of Network Vulnerability Technique Comparison

In previous simulations, we typically saw communication security increase to 100% from the first evolved candidate, where it remained throughout all evolutions, seldom did we see a minor reduction in security. Throughout these simulations the security fluctuates from 98% to 100% (Figure 5.21-e), with the optimal solution for the unscanned network evolving with a 98% secure solution when GA was applied, while ANTs optimal solutions present a 100% secure candidate. These fluctuations are directly attributed to the increased number of nodes, communication links, and additional security grade parameters we have introduced into the framework.

Comparing the analysis for degree centrality (Figure 5.21-b), minimum path average (Figure 5.21-c), and maximum bridging centrality (Figure 5.21-d), comparable to the previous simulations we note significant fluctuations between candidate scores. An example of this is the results documented for degree centrality (Figure 5.21-b) when we specifically look at the GA when applied to the network with a conducted vulnerability scan. Results show in comparison to previous simulations that degree centrality for the mutated candidates prominently fluctuated throughout the process, whereas previous simulations in general had minor fluctuations in degree centrality and tended to steadily increase or decrease throughout evolution. In this instance however, we note that degree centrality for the scanned network with vulnerabilities in evolution 4 is increased by 5.88% when compared to the original topology, and in the final evolution it has decreased by 70.59%. While minor increases to

centralities would not necessarily impede an overall network's robustness, when we see significant fluctuations these will influence the suitability of those candidates which appear to have been negatively impacted.

Due to the increased number of nodes and links which form the network topology there are considerably more alternative network configurations which the network could evolve into, and as security grades are more accurate these positively influence the security risk mitigation process. All these factors will directly impact how large SoS will evolve, which candidates are to be considered, while modifying and impeding centrality factors and security. And as previously stated, the framework generates and reports not only the optimum candidate but alternative solutions allowing for us to analyse results and make informed decisions.

5.3 Effectiveness of Risk Mitigation Within Secure Networks

In an effort to determine and evaluate the proposed framework's complete functionality, we decided to simulate and analyse networks that were considered as 100% secure, to establish if the risk mitigation process could enhance data security and improve data flow, reporting additional improvements, while maintaining the security of the network. Firstly, we simulated a network consisting of 8 static nodes with 50% connectivity and 100% communication security, using the simplistic version of the SCRAM framework running the GA and ANT algorithms against the test network visualised in Figure 5.22.

When the ANT based risk mitigation process was applied to the network, it failed to find any improved solutions, which is caused by the use of comparison trails within the algorithm. As the network already has 100% secure communication, it will not consider any solution which does not match or improve this parameter. Bridging centrality and the robustness of the network are example constraints which are also compared, should any one of these parameters fail to be improved then the evolved network is simply ignored, resulting in the risk mitigation process failing to find any improved solutions.

The algorithm's trails restrict the framework from considering alternative solutions that while they might increase cost or impede other centralities have the overall potential to provide a more ideal network configuration in comparison to the original network.

Applying the GA based risk mitigation processes to the network, the tool reports eight mutated networks that all have improved robustness levels and have maintained 100% security. Figure 5.22 visualises the original network and every subsequent enhanced solution found, in a series of undirected graphs.

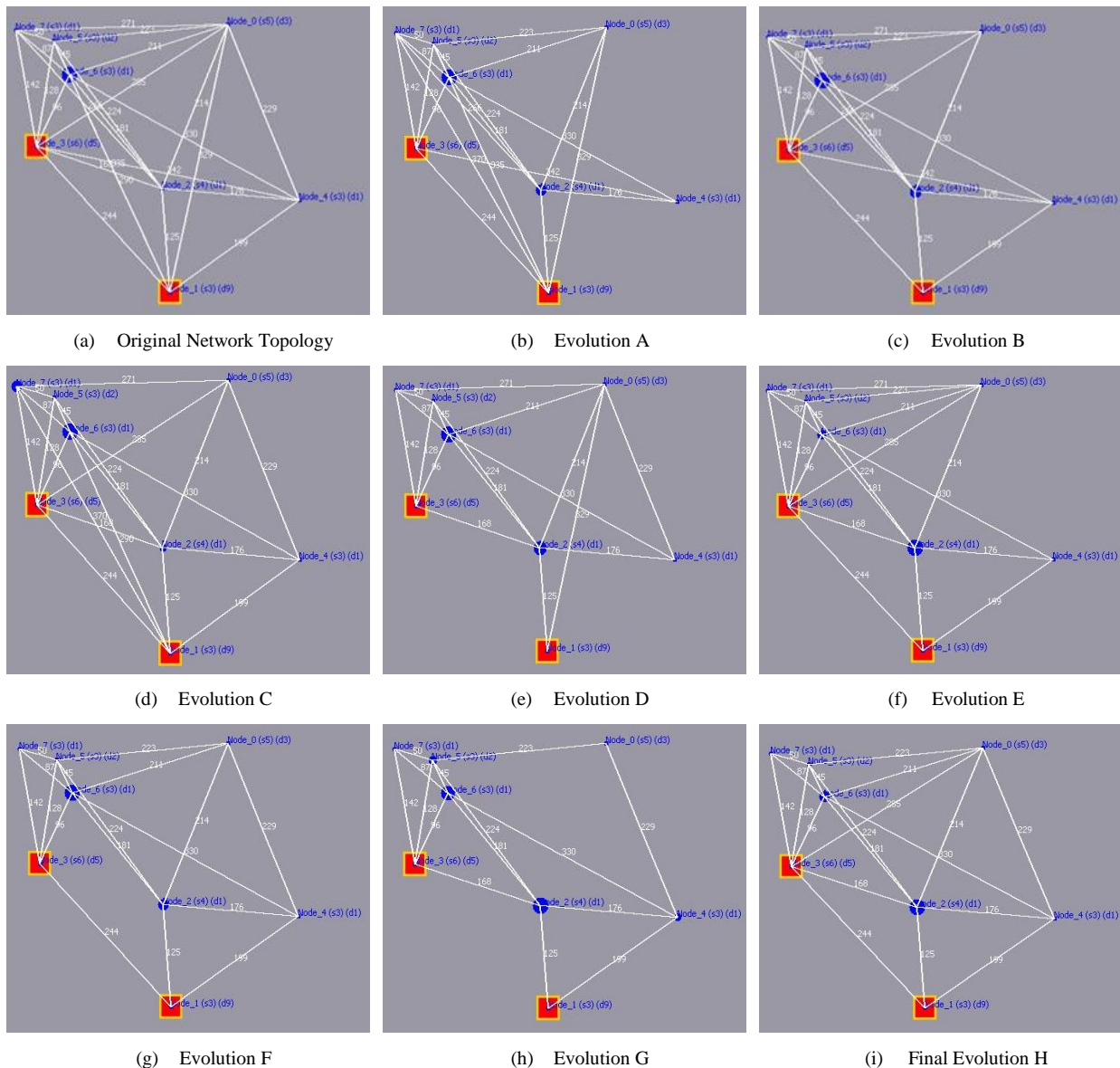


Figure 5.22. Evaluating a Secure System-of-Systems

When analysing the GA improved solutions we quickly ascertain that while evolution H (Figure 5.22-i) is quantified the optimum solution and costs 3357 which is 35.32% less than the original network, evolution G (Figure 5.22-h) is the cheapest improved solution reducing the cost by 53.7% from 5190 to 2403. This reduction in cost is also reflected in the number of links each evolved solution is constructed with.

The test network is formed using 8 static nodes connected via 23 communication links. As visualised in Figure 5.22 we can intuitively see that there has been a reduction in the number of links in each of the evolved reported solutions, with the optimal solution (evolution H) connecting all nodes via 19 links which is a 17.39% link reduction, while evolution G is formed with only 15 communication links which is a 34.78% link reduction. We also note that in both these instances we can clearly identify that there are alternative communication paths between nodes, meaning should a node or link be removed or disabled, data flow will be maintained between all remaining nodes.

As the final optimum solution has additional links compared to other reported alternative solutions, it means we have to consider that while additional links can increase network risks there is more redundancy within its configuration, thus there are many alternative paths between nodes, i.e. in the event of multiple path failures the risk of network failure or node isolation is minimal in comparison, for example, to evolution G.

As stated the framework has maintained 100% security (Figure 5.23-e) and improved the robustness of the network (Figure 5.23-f) by reducing its robustness value marginally. While the optimum solution has only reduced robustness by 7.66% from 216.6052 to 200.0033, and the cheapest evolution reduced robustness by 7.18% to 201.0575, when we consider the financial savings that could be accrued by implementing either improved solution it potentially outweighs the consequence of only marginally improving the network's robustness. However, we do have to consider the negative impacts such as the increase to other centralities.

Evolution g and h both increase the degree centrality of the graph (Figure 5.23-b) and the maximum bridging centrality (Figure 5.23-d). Evolution g is the cheapest evolution, and increases degree centrality by 66.67% which is 0.238095 compared to the original network which was only 0.142857, and increases bridging centrality by 93.65% from 0.013173 to 0.02551. The optimal solution (evolution h) also increases degree centrality by 66.67%, and increases bridging centrality by only 32.27% to a value of 0.18083. At first glance the percentages appear to be high, but the values are still very low and in an acceptable range, and this tolerable increment is acceptable given the overall cost reductions.

Minimum path average (Figure 5.23-c) is marginally increased by the optimal solution, with the optimal value of 229.0714 only increasing the original network path average value of 227.1071 by 0.86%, while the cheapest evolution increases minimum path average by 8.08% to 245.4643. The average minimum path is slightly increased as a direct result of the loss of links, but again this would be considered acceptable given the reduction in network cost and risk.

Figure 5.24 shows the centrality values degree, betweenness, closeness, eigenvector, and bridging for each node in the original network, optimal network (evolution H), and the lowest costing network (evolution G). In each line graph the optimal solution node values are characteristically centralised, in between the original network and cheapest alternative enhanced solution. This implies the optimal solution average node centralities never fluctuate above or below the original network and evolution G.

For example, as an alternative to analysing the entire network, using the framework's reports we can also evaluate each node within the original network and each of the reported candidates. When we view degree centrality (Figure 5.24-a), in the original network the average node value is 0.892857. After the security risk mitigation process, reports indicate that the optimal solution has reduced

average node degree centrality by 24% reducing the average value to 0.678571, while evolution G has reduced average node degree centrality by 40% to 0.535714.

Average node centrality reductions are also evident in closeness and eigenvector, with the optimal solution reducing the average node closeness centrality by 0.81% and average node eigenvector centrality by 48.38%. Evolution G had a reduction of average node closeness centrality by 7.03% and average node eigenvector centrality by 70.16%. Reductions with closeness centrality are to be expected due to the loss of communication paths. While many of the values are significantly reduced and others only minimally improved the overall robustness of the network, when we consider the reduction in network cost, both the optimal solution and cheapest alternative would be highly acceptable and suitable configurations to implement.

We do note however, that average node centrality values for betweenness (Figure 5.24-b) and bridging (Figure 5.24-e) centrality do increase. The optimal solution increases the average node bridging centrality by 224.97 % and average node betweenness centrality by 200%, while evolution G increases average node bridging centrality by 475.99% and average node betweenness by 366.67%. At first glance these average node centrality percentages appear to be excessively high, however, when we view the actual values we see that the average node values are in an acceptable range.

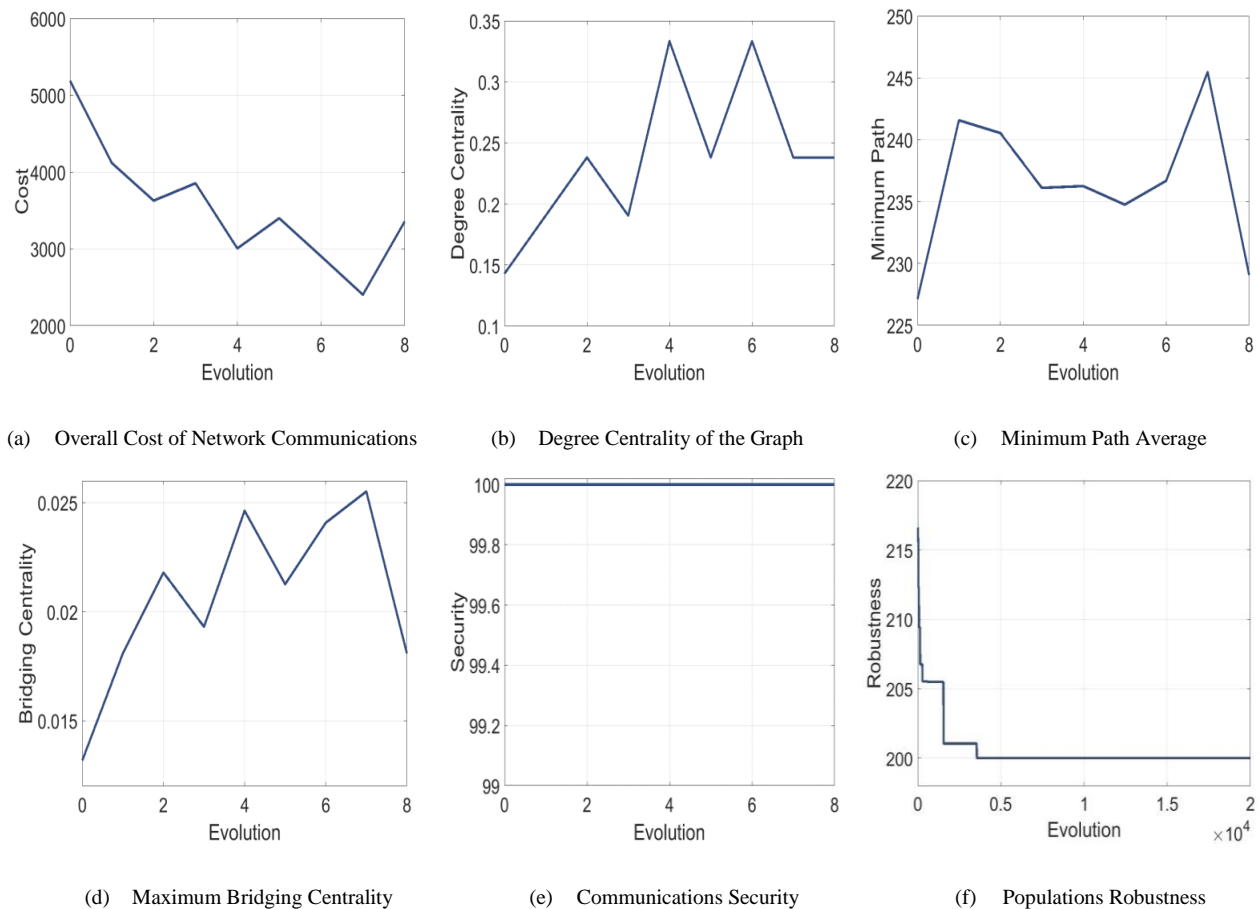


Figure 5.23. Analysis of the Secure Network When GA was Applied

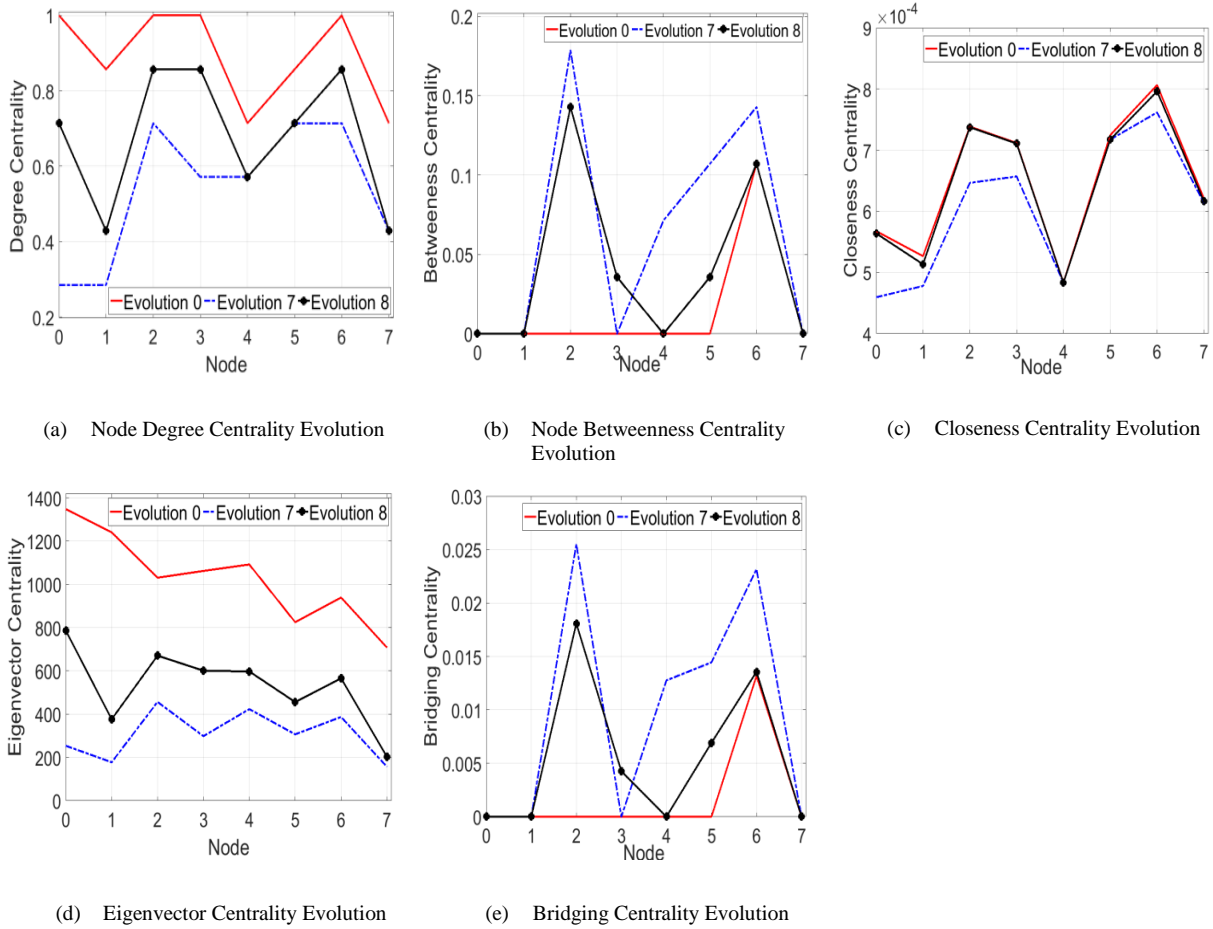


Figure 5.24. Analysis of Node Centralities of Secure Network When GA was Applied

For example, the original average node bridging centrality is 0.001647, with the optimal solution only increasing to 0.005351 and evolution G only increasing to 0.009485. The same can be seen with the original average node betweenness centrality which was 0.013393, with the optimal solution only increasing to 0.040179 and evolution G only increasing to 0.0625. All these values are in a tolerable range, and with the reduction of communication links and cost decrease we expect to see a direct correlation to the increase of these two centralities.

5.3.1 Evaluating Positive Security Risk Mitigation of Secure SoS

Using the full implemented SCRAM framework and applying both GA and ANT security risk mitigation techniques, we simulated four different networks which all had a communication security scores of 100%. Firstly, we simulated two 8 node networks, one with 30% connectivity (Figure 5.25-a) and one with 50% connectivity (Figure 5.25-d). Secondly, we simulated two 12 node networks, one with 30% connectivity (Figure 5.26-a) and one with 50% connectivity (Figure 5.26-d).

Alike to our previous simulation to enhance a 100% secure network, when we combined the risk mitigation algorithm with ANT optimisation algorithm and applied the technique to all four networks,

it failed to find any improved solutions due to its strict comparison criteria trails. When we applied the combined risk mitigation algorithm with GA via the framework to the same four individual networks, the security risk mitigation process reported a series of evolved candidates for each network, detailing alternative improved candidates.

When we review the undirected graphs, we can intuitively see a reduction in links for all four evolutions when we compare the original networks (Figure 5.25-a, Figure 5.25-d, Figure 5.26-a, and Figure 5.26-d) to the first mutated evolution (Figure 5.25-b, Figure 5.25-e, Figure 5.26-b, and Figure 5.26-e) to the final optimum evolution (Figure 5.25-c, Figure 5.25-f, Figure 5.26-c, and Figure 5.26-f). Table 5.10 through to Table 5.13 display each network’s reported improved candidates, including the number of communication links connecting static nodes within the network.

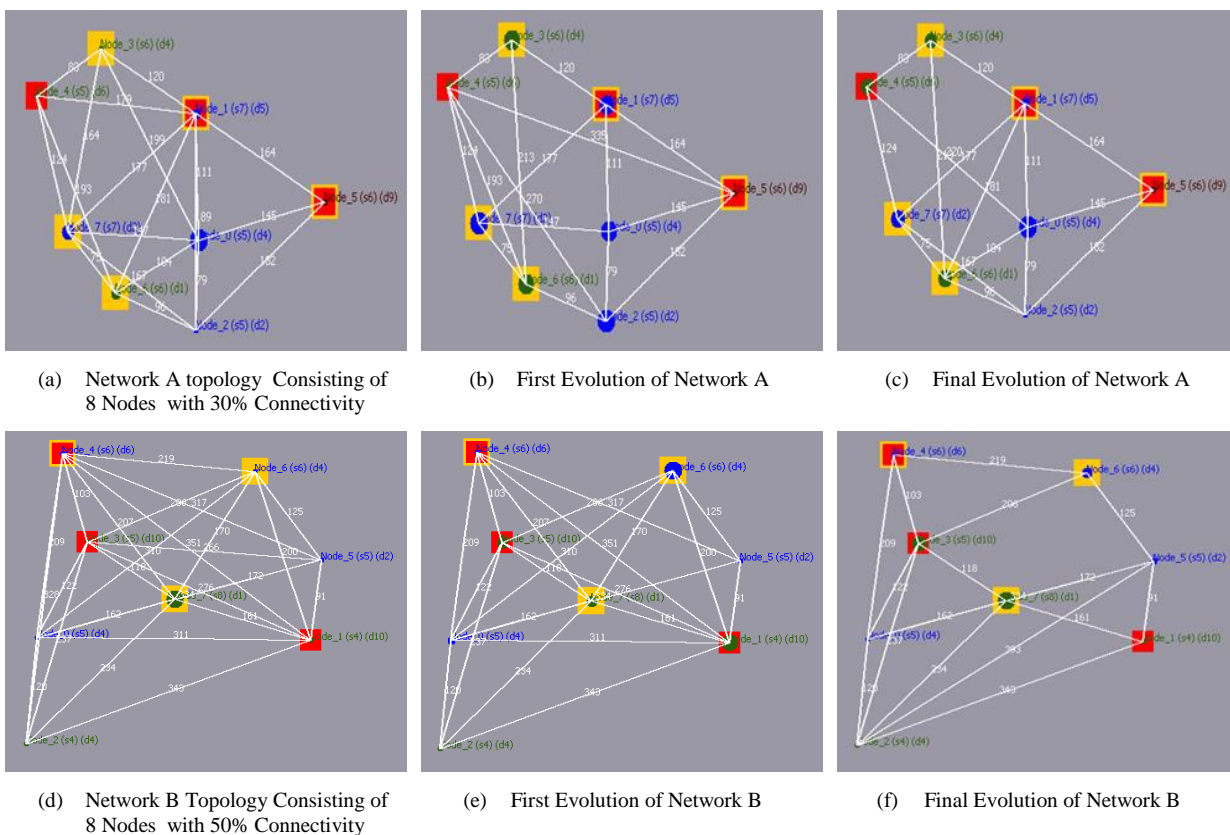


Figure 5.25. Secure 8 Node Network Comparisons with Applied GA Risk Mitigation

Table 5.10 establishes that the original network is comprised of 20 communication links; once the GA process was applied to the network each improved mutated network generated was comprised of a reduced number of data links. For example, Network A (Figure 5.23-a, 8 nodes 30% connectivity) sees a 20% reduction in links for the first evolution reducing communication links from 20 to 16, as the network is further mutated we see the number of links fluctuate down to 15 in its second improved candidate then increase to 17 links until its final optimum solution is reported. The final optimum solution also sees a reduction of links by 20%; this reduction is reflected in the decrease to communication costs which drops by 22.16%.

While network B (Figure 5.25-d, 8 nodes 50% connectivity) reduces links by 34.62% and cost by 41.16% (Table 5.11), Network C (Figure 5.26-a, 12 nodes 30% connectivity) reduces links by 34.69% and costs by 32.16% (Table 5.12), and Network D (Figure 5.26-d, 12 nodes 50% connectivity) has a final optimum solution that reduces links by 39.29% and has a reduced cost of 43.7% (Table 5.13). These significant reductions to the overall cost of the network communications (Figure 5.27-a) and decreases to the number of data links, while they do not greatly improve population robustness (Figure 5.27-f) there is a notable improvement to comparison parameter scores confirming the overall appropriateness of the network's improvements.

Alike to previous simulations, these reductions to robustness, cost, and link reduction, increase each network's minimum path average (Figure 5.27-c) marginally by an average of 5.27% and bridging centrality (Figure 5.27-d) notably by 179.74%. Again, while the percentage increases appear to be significantly high for the bridging centrality, when we examine the values we note they remain low and within a tolerable range. For example Network A only increase from 0.013812 to 0.0217, Network B increase from 0.0093236 to 0.0251, Network C from 0.005731 to 0.021212, and Network D increase its bridging centrality to 0.013492 from 0.004233.

Table 5.10. Secure Network A Security Evolution Results

Evol.	No. of Links	Cost	Min. Path Avg.	Sec.	GA		Node Centrality Averages			
					Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	20	2879	183	100	169.6392	0.714286	0.035714	0.000798	477.25	0.004735
1	16	2514	192.0357	99	166.8778	0.571429	0.053571	0.000755	323.4531	0.013947
2	15	2321	192.25	99	163.6267	0.535714	0.058036	0.000754	277.625	0.015831
3	17	2679	186.6071	100	163.5985	0.607143	0.053571	0.000782	371.1406	0.012117
4	17	2629	187.2857	100	163.5147	0.607143	0.053571	0.000781	375.4531	0.008269
5	17	2508	186.1429	100	162.6874	0.607143	0.053571	0.000785	338.3438	0.009929
6	16	2241	188.2857	100	160.8404	0.571429	0.058036	0.000776	291.7969	0.012439

Table 5.11. Secure Network B Security Evolution Results

Evol.	No. of Links	Cost	Min. Path Avg.	Sec.	GA		Node Centrality Averages			
					Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	26	5692	232.2143	100	222.4113	0.928571	0.008929	0.00063	1166.078	0.001155
1	22	4707	241.6429	100	222.2639	0.785714	0.026786	0.000604	843.5156	0.003768
2	20	4014	244.75	98	218.6872	0.714286	0.035714	0.000598	665.3281	0.005135
3	21	4542	240.3214	100	214.7726	0.75	0.03125	0.000609	771.4063	0.004399
4	19	3897	241.8929	100	213.3133	0.678571	0.044643	0.000606	617.375	0.006534
5	18	3444	239.8571	98	212.0986	0.642857	0.044643	0.000609	519.2813	0.006567
6	16	3194	247.25	100	210.9302	0.571429	0.058036	0.000589	418.125	0.010907
7	16	3163	243.75	100	208.9639	0.571429	0.058036	0.000599	399.3125	0.011515
8	17	3349	242.8929	100	207.945	0.607143	0.053571	0.0006	470.9063	0.010198

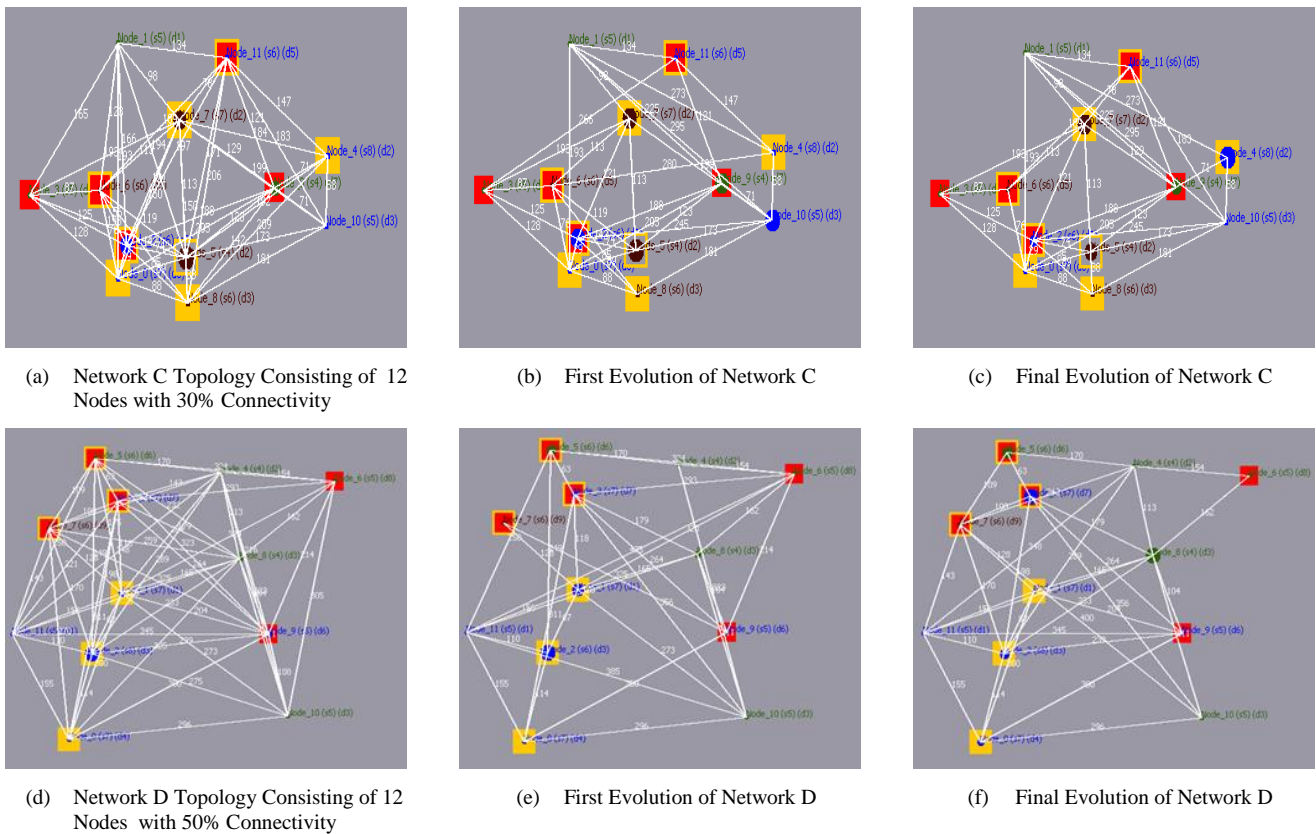


Figure 5.26. Secure 12 Node Network Comparisons with Applied GA Risk Mitigation

The framework’s detailed reports also allow us to look closely at each node to see how their centralities are impacted as the network evolved. Table 5.10 through to Table 5.13, display aggregated node centrality scores for each node within the network, as an overview for each improved reported network candidate. Having an overview of nodes assists greatly when we want to establish how individual nodes are impacted in comparison to the network, and if we look at bridging centrality for Network B for example, we ascertain that while the overall network bridging centrality has increased by only 172.94%, the average node bridging centrality has increased by 782.94% from 0.001155 to 0.010198 (Table 5.11). The report indicates that in this instance the value is still in an acceptable range, but provides a solution for quick analysis to assist with any decision making processes. Reviewing individual node risks is vital, especially if we have to ensure that a specific node conforms to strict risk limitations if it is deemed as a critical node within the infrastructure.

Additionally, these reports help us to identify not only the values for the optimum solution, but we can scan for the candidate that would be the cheapest to implement and its impact on centrality values and security, in an attempt to identify if it would be a suitable alternative solution in comparison to the optimum solution for example. When we review Networks A and C, it is ascertained that there are no cheaper alternatives to consider, the final optimum solution has the lowest network communication cost to impellent and maintains 100% communication security in comparison to the original network’s parameters.

Table 5.12. Network C Security Evolution Results

Evol.	No. of Links	Cost	Min. Path Avg.	Sec.	Node Centrality Averages					
					GA Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	49	6664	171.8788	100	176.6313	0.742424	0.021465	0.000544	785.0486	0.001711
1	34	5057	199.0455	100	176.2492	0.515152	0.044192	0.000466	409.2361	0.006206
2	35	4781	187.6061	100	167.7929	0.515152	0.044192	0.000496	391.0694	0.005718
3	32	4521	183.0152	100	166.4406	0.5	0.051768	0.000512	368.9931	0.00768

Table 5.13. Network D Security Evolution Results

Evol.	No. of Links	Cost	Min. Path Avg.	Sec.	Node Centrality Averages					
					GA Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	56	12127	245.9545	100	263.0117	0.848485	0.012626	0.00038	1591.632	0.001032
1	32	7354	285.5454	98	261.9081	0.484848	0.05303	0.000327	601.1111	0.008356
2	42	9861	264.894	100	259.1658	0.636364	0.030303	0.000352	1010.028	0.003778
3	37	8325	276.4243	100	254.8373	0.560606	0.037879	0.000337	729.3194	0.005114
4	35	7217	271.6364	100	246.8471	0.530303	0.041667	0.000343	626.8819	0.005559
5	32	6556	278.7727	99	243.5951	0.5	0.049242	0.000334	540.0694	0.006642
6	35	7365	263.8788	100	243.2883	0.530303	0.040404	0.000354	669.8056	0.004138
7	32	6447	268.394	98	242.0958	0.484848	0.046717	0.000347	519.125	0.005141
8	31	6505	280.4546	100	240.349	0.469697	0.046717	0.000332	508.5556	0.006897
9	33	6876	266.6667	100	235.155	0.484848	0.046717	0.000349	543.0278	0.00561
10	33	6538	265.1364	100	234.749	0.5	0.045455	0.000352	545.4306	0.005891
11	34	6827	263.4091	100	234.6132	0.515152	0.042929	0.000354	588.6528	0.005522

Networks B and D both have cheaper candidates that could be alternative solutions in comparison to the original network and optimal candidate. Reviewing Network B we identify that candidate 7 (Table 5.11 evolution 7) would be marginally cheaper to implement than the optimum candidate (Table 5.11 evolution 8), with a 5.55% difference between their costs. Both candidates maintain 100% communication security and improve network robustness, with a 0.49% difference between their robustness scores. This marginal cost saving means, while this candidate would be the cheapest to implement, analysis suggests that the optimum solution would be ideal to implement as it still reduces overall network cost by 41.16%, maintains security, enhances robustness, and has a marginally lower minimum path average in comparison to the cheapest alternative candidate.

Analysing Network D, we ascertain that there are four lower costing alternative candidates than the optimum final solution. Candidate 7 (Table 5.13 evolution 7) is the cheapest reported solution in comparison to all candidates, but it does not maintain 100% communication security as it reportedly has a security value of 98%. Immediately we dismiss this solution as our objective is to improve and mitigate risk within the network while maintaining its 100% communication security status. The second cheapest candidate to implement is candidate 8 (Table 5.13 evolution 8) which reduces network costs by 46.36% from 12127 to 6505, in comparison to the optimum final candidate (Table 5.13 evolution 11) which reduces cost by 43.7% to 6827. There is only a 4.16% difference between

the candidates' costs meaning evolution 8 would save an additional 282. Furthermore, candidate 8 has lower node averages for degree and eigenvector centralities in comparison to the optimum solution.

We observe that the optimum solution in comparison to candidate 8 has a marginal lower minimum path average and robustness, which is a minor difference of 2.44% for robustness and 6.47% for minimum path average. It is ascertained though that both candidates have reduced cost, robustness, and average node degree centrality, while maintaining 100% communication security. And the marginal differences between parameters is not significant enough to ascertain which would be the most viable to implement, as while one is considered the optimum, candidate 8 does have a cost saving. If an organisation had budget restrictions because the framework did not only just present the optimum solution but alternative candidates, these alternative evolutions can be considered for application in the awareness that the framework has mitigated risk and improved the overall robustness of the network. These evolutions and recommended improvements assure network security and reduce potential risks to data communications.

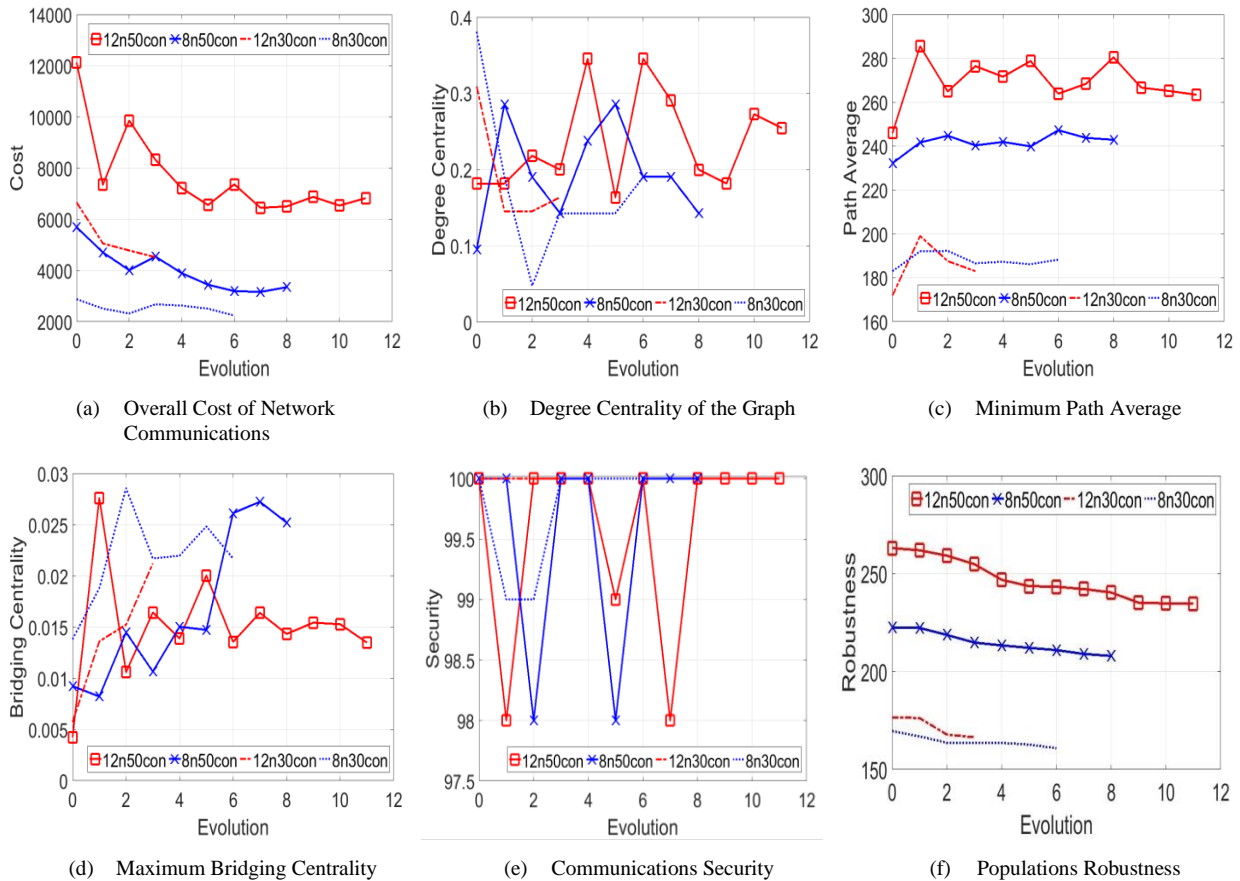


Figure 5.27. Security Analysis of Secure Networks When GA is Applied to Mitigate Risks

Comparing the analysis for communication security for all four networks, again we observe a larger volume of communication security fluctuations between evolved candidates when utilising the full framework to run simulations, as evident in Figure 5.27-b. Due to the improved functionality of the framework, when quantifying security grades and vulnerability identification, owing to the inclusion

of NVD vulnerabilities and CVSS v3 scores, we see similar characteristics to previously discussed simulations, where the network has been evolved and the overall robustness of the networks is improved, while negatively impacting the network communication security. The lowest reported security value for these simulations is 98%. When evolving networks, due to the accuracy of vulnerability scoring and reduction in communication links, we recognise that this will negatively impact security, closeness centrality, and impede network communication. Even so, all four simulations have reported their final optimal solutions have evolved and maintained 100% security.

5.3.2 Evaluating Negative Security Risk Mitigation of Secure SoS

It must be noted that not all 100% secure networks can be evolved and enhanced with a positive outcome. The previous four simulated networks (Section 5.3.1, Figure 5.25 and Figure 5.26) clearly demonstrate how 100% secure networks can be enhanced further and have their risks mitigated, robustness and related centralities improved, along with reducing their associated costs. During our simulations and analysis we modelled several 100% secure networks that could not be improved and the SCRAM framework reported no improved candidates, indicating that the network was optimally configured to meet security requirements and could not be enhanced further.

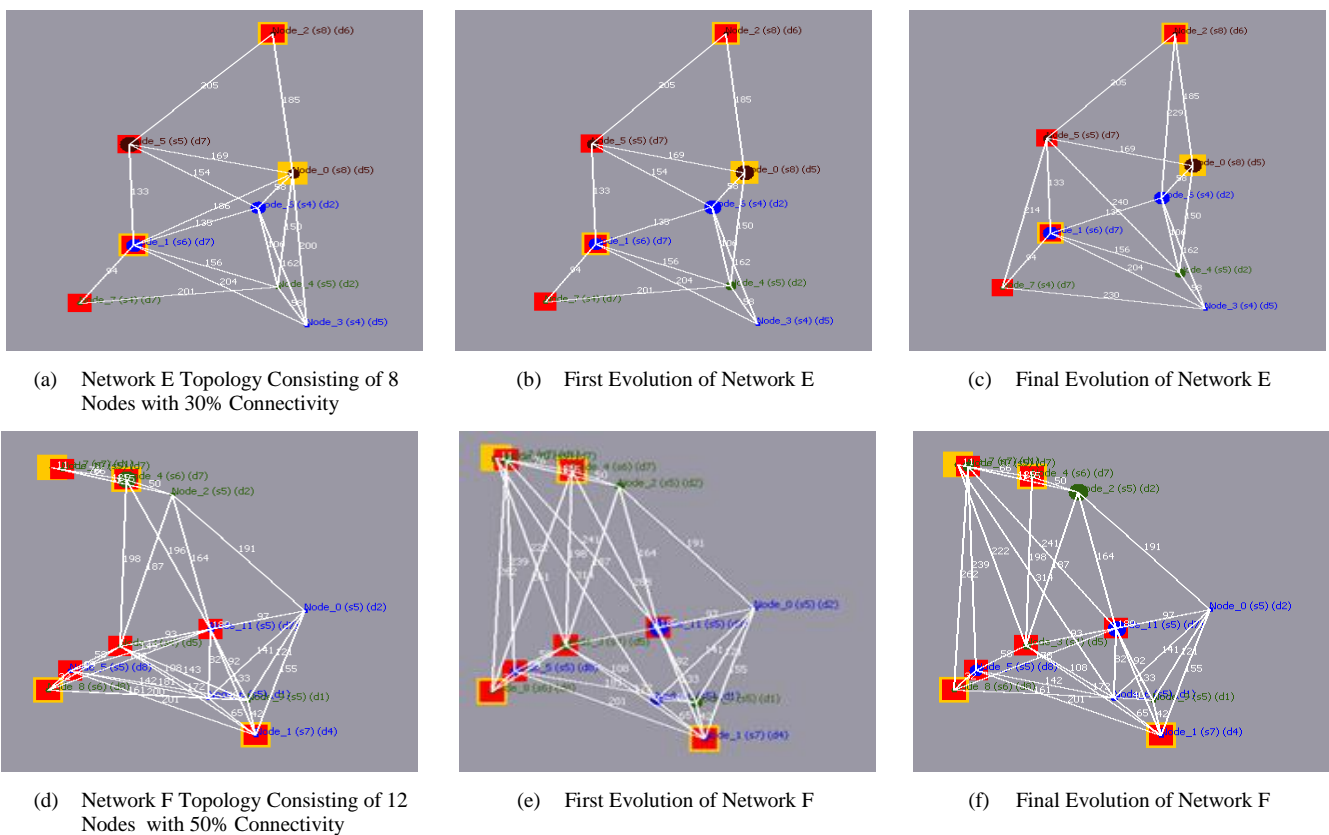


Figure 5.28. Comparison of Secure Networks Evolved that Negatively Impact Network Cost

In this section, two simulated networks (Figure 5.28) are presented that model 100% secure networks that when evolved negatively impact the overall cost of network communications. Network E (Figure 5.28-a) consist of 8 static nodes with 30% connectivity and 100% security, and Network F (Figure 5.28-d) consists of 12 static nodes with 30% connectivity and 100% security. When we view the candidate graphs in Figure 5.28, we cannot intuitively see a reduction in communication links between the original graphs, the first evolution candidates, and the final optimum solutions, instead we have to analyse the network reports to ascertain if there has been any link reduction. Table 5.14 indicates that the optimum solution for Network E has no link reduction despite other reported candidates evolving with marginally fewer links, while Table 5.15 indicates that the optimum solution is reduced by 5.56% from 36 links to 34, and its other reported candidates have evolved with fewer links.

Despite both networks maintaining or reducing the number of communication links, these networks, when processed, report an increase to the overall cost of the network (Figure 56-a). Network E has increased its cost by 6.73% and network F is increased by 5.49%, therefore reconfiguring the network when on a tight budget may not be advisable. Although it must be noted that this small increase to cost would assure that the network was reconfigured mitigating risk and enhancing security, as evidenced by the minimal improvement to the robustness score (Figure 56-f). The robustness for Network E only improves by 1.91%, but Network F is improved by 9.35%. If the minimal cost increase was within budget then these improvements would further reduce the associated risks to the network and strengthen the SoS security.

Table 5.14. Network E Security Evolution Results

Evol.	Number of Links	Cost	Minimum Path Avg.	Security	GA		Node Centrality Averages			
					Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	17	2556	209.2857	100	181.0403	0.607143	0.053571	0.000712	376.1875	0.007908
1	15	2170	210.3571	98	179.4276	0.535714	0.071429	0.000708	274.5938	0.011962
2	16	2498	213.25	100	178.2958	0.571429	0.058036	0.000696	334.5	0.009631
3	17	2728	210.8214	100	177.5785	0.607143	0.053571	0.000702	371.0938	0.009807

Table 5.15. Network F Security Evolution Results

Evol.	Number of Links	Cost	Minimum Path Avg.	Security	GA		Node Centrality Averages			
					Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0	36	4443	204.0758	100	194.4821	0.545455	0.050505	0.000456	414.4375	0.00613
1	33	4860	204.2424	100	184.7208	0.5	0.045455	0.000449	381.1319	0.007668
2	32	4764	206.2121	100	183.6317	0.484848	0.045455	0.000448	398.1319	0.005059
3	34	4687	196	100	176.3076	0.515152	0.04798	0.000472	403.8194	0.005865

These two networks are only similar in the fact they are both fully secure networks, and when we review the enhanced results we note that Network E has a 0.73% increase to its minimum path average and 10.71% increase to its bridging centrality, whereas Network F shows a decrease to its minimum path average by 3.96% and has a network bridging centrality decrease of 62.3%. This is

expected due to the elimination and establishment of links between nodes within each network, as minimum path lengths alter as routes are reconfigured and security maintained.

We also see a reduction to the network's degree centrality (Figure 5.29-b), with Network E reducing by 57.14%, from 0.333333 to 0.142857, and Network F reducing by 22.22%, from 0.327273 to 0.254545. In addition, to evaluating the network as a whole, we also analyse the node centralities for these networks. Reviewing Network E (Table 5.14), we note that the final enhanced solution (Table 5.14 evolution 3) maintains the average node degree and betweenness centralities, while reducing the eigenvector centralities and slightly impacting closeness centralities compared to the original network (Table 5.14 evolution 0).

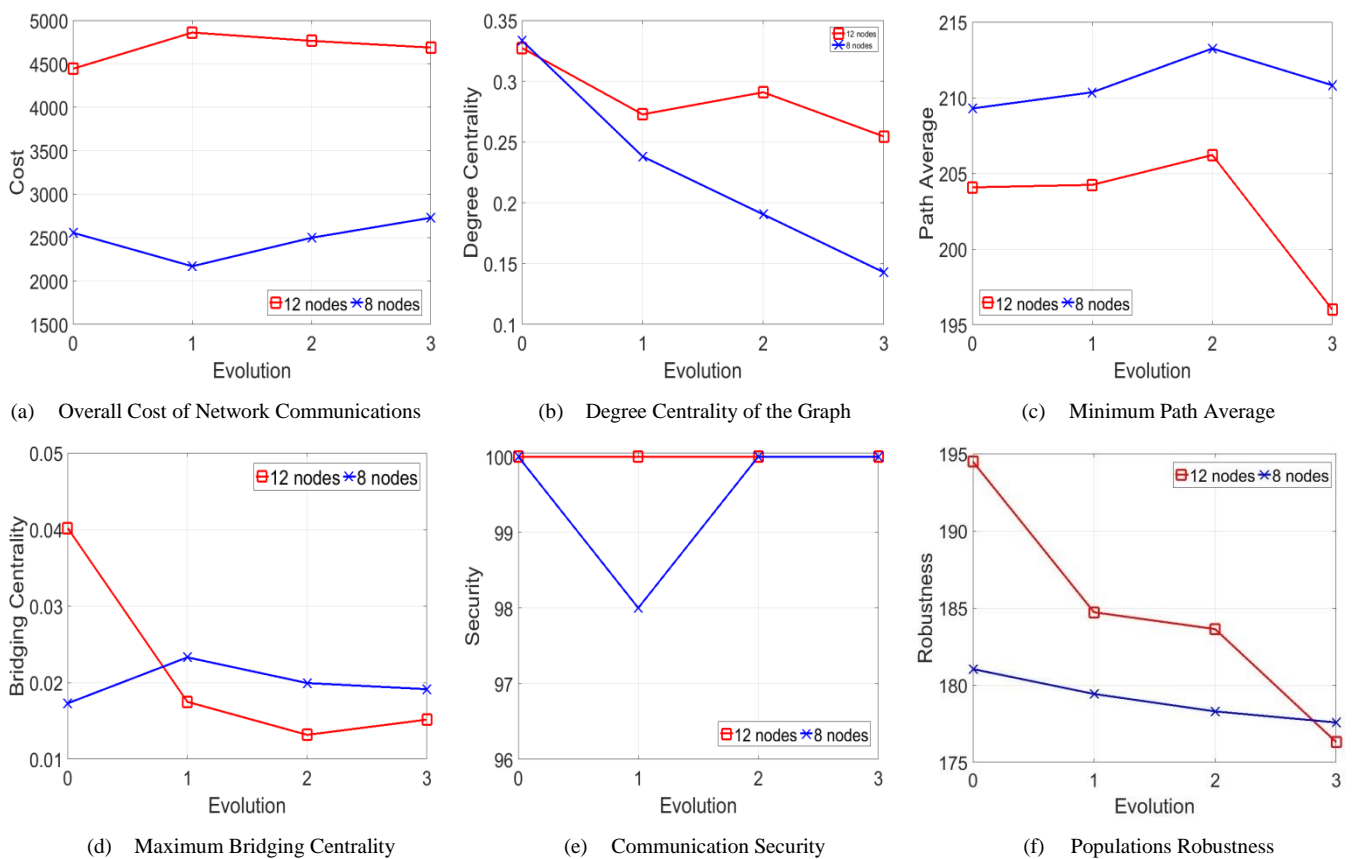


Figure 5.29. Security Analysis of Secure Networks When GA is Applied with Negative Outcomes

Table 5.14 also shows that evolution 2, while not the optimum solution should be considered as a suitable alternative. Evolution 2 has lower cost, robustness score, and network bridging centrality in comparison to the original network, and maintains 100% communication security. This candidate's average node degree, closeness, and eigenvector centralities are also lower than the original networks, and it minimally impacts the networks minimum path average, closeness centrality, and aggregated node betweenness and bridging centralities. Again, having an optimum solution and second candidate to consider provides decision makers with alternative solutions to contemplate for implementation, and in this instance provides solutions that are capable of assuring security and mitigating risk, as well

as reporting cheaper alternatives to the optimum candidate that maintain security should financial restrictions be a major factor for consideration.

Analysing Network F (Table 5.15), we ascertain that the optimum candidate (Table 5.15 evolution 3) is the lowest cost evolved solution, and despite this small increase we see a reduction to not only its minimum path average, number of links and robustness score, but to all node average centralities when we compare them to the original network (Table 5.15 evolution 0). The analysis for this reported candidate shows that its closeness centrality has only been marginally impacted.

As previously stated, while Network F has an optimum candidate with a lower number of communication links, compared to its original configuration it does not mean this is the most secure network. In the event of communication link failure there must be adequate data links to assure that data communication does not fail between nodes, therefore communication must have a safe route via alternative paths. In this instance while fewer links has actually increased the overall cost of the network, this is a direct result of the removal and establishment of new links forming the network, which alter path lengths.

5.4 Case Study

5.4.1 Node Energy Efficiency Problem

We consider Smart Cities to be an ideal representation of an SoS due to their dynamic nature, complexity, diverse composition and architecture, and dissimilar security levels. WSN within Smart Cities are generally formed by means of sensors which are low in cost and power, and capable of sensing, processing, and communicating data. Information is gathered then processed locally, the node then forwards the data to a sink. Characteristically, nodes have short transmission ranges, meaning data is generally transferred via other nodes using multi-hop paths. Sensors are typically powered by batteries, which are difficult or impractical to change and cannot be recharged. Often it can be more cost effective to replace the entire sensor than substitute the original sensor's drained power source [258] [259].

Switching a sensor's radio off when the node is idle can assist with energy conservation, but the node has to remain active in a state of idle listening in case traffic is forwarded on the channel. Remaining in this state can consume 50-100% of a node's energy which is required for receiving, and data exchange is the most prominent function of the node compared to processing and sensing [258].

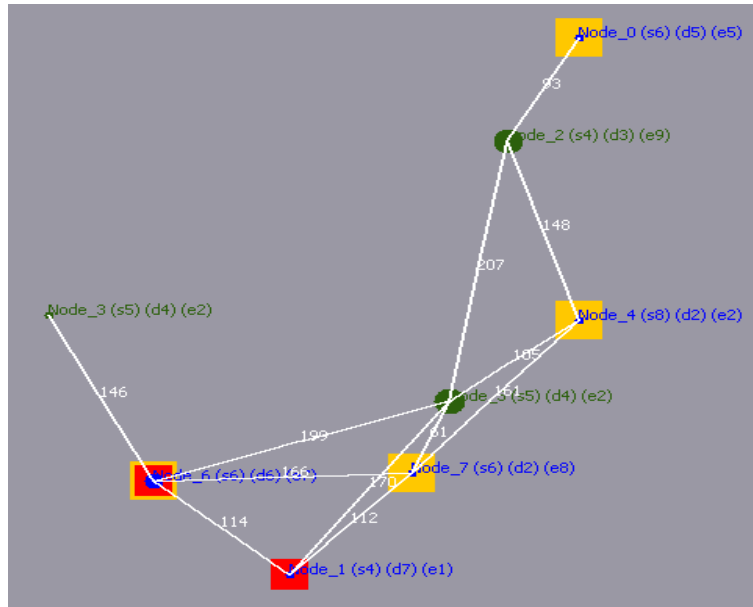
Other issues that impact energy efficiency include control packet overhead, collision, and overhearing. Control packet overhead impacts energy efficiency and bandwidth due to the regular delivery of updated control packets and synchronization. Collision impacts energy efficiency as data

in the network can be transferred by two or more nodes simultaneously; this data can then become corrupted resulting in it being discarded. Consequently, the data requires retransmission, resulting in further energy consumption along with increasing network latency. While overhearing impacts sensor energy as when data is transmitted from one node to another, the data can be accidentally received by all neighbouring nodes despite the fact they were not destined to receive the data. Sensor nodes are also prone to failure due to environmental factors and energy consumption, which causes changes to the resource strained network topology and will impact the energy consumption for the remaining nodes [258] [259]. When considering the integration of WSN into Smart Cities, extending their life time is essential. We monitor energy efficiency for each node and integrate the parameter within the SCRAM frameworks security principles as an identified risk, to assist in enhancing the security and mitigating risks within the SoS, while endeavouring to extend the life of the Smart City.

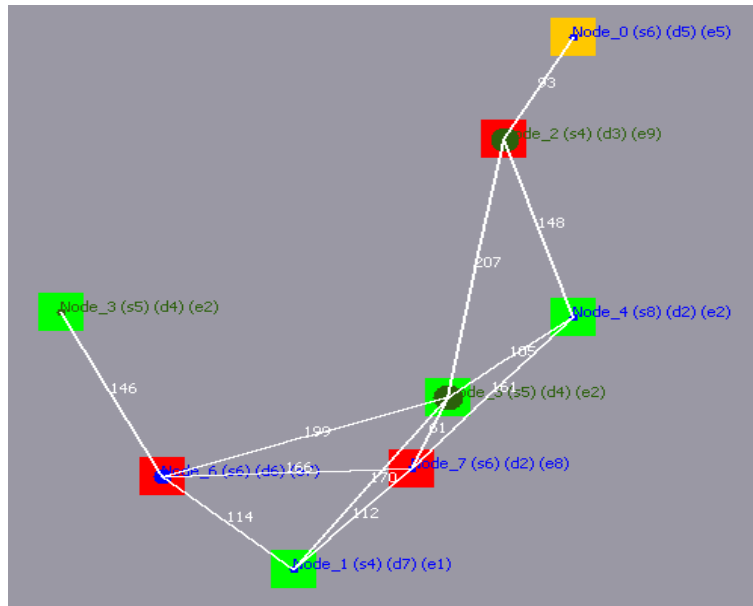
5.4.2 Node Energy Efficiency Comparison

The algorithms, principal concepts, and robustness function presented in Section 4 have been incorporated into the SCRAM framework, and have the ability to reconfigure collaborative networks into optimally secure configurations. This framework additionally provides an inexpensive simulation model to conduct experiments within, allowing us to study the behaviour of the systems and techniques. To evaluate the usefulness of the applied theoretical principles, we extend this novel approach to monitor and incorporate node energy level status, and integrate the quantified energy values within the security risk mitigation process, in an attempt to extend the life of energy restricted devices and networks that form part of an SoS, such as WSN within Smart Cities.

For this case study, we simulate a section of a Smart City, which is a WSN consisting of 8 static nodes with a low connectivity of 30%, formed using a variety of ICT devices. This includes sensor nodes and mobile devices, all of which are assigned the relevant node software, hardware, and firmware parameters such as type of operating system, energy level, data access grade, whether it supports encryption, Internet access, incorporates firewalls, IDS, anti-virus/security, and if the node has been completely updated or has vulnerabilities, depending on device type.



(a) WSN A Security Graph, Including Node Security Grades, Data Levels, Node Status, and Bridging Centralities (Network A)



(b) WSN B Energy Efficiency Graph, Including Energy Level, Security Grades, Data Levels, and Bridging Centralities (Network B)










Figure 5.30. Primary Simulated Smart City WSN Environment

SCRAM randomly assigned all nodes with a security level and connected them via a series of primary communication links, and assigned a random network data level, which nodes will be compared against replicating data access control principles. Then the framework quantified the network's degree, betweenness, closeness, eigenvector, and bridging centralities, the communications security, minimum path average, and the network's associated cost.

Figure 5.30-a depicts the WSN, displaying key parameters so we can examine the graph intuitively, and visualises the security of the network. Figure 5.30-b visualises the same simulated Smart City section, however, it has been imported back into the SCRAM framework with the energy efficiency

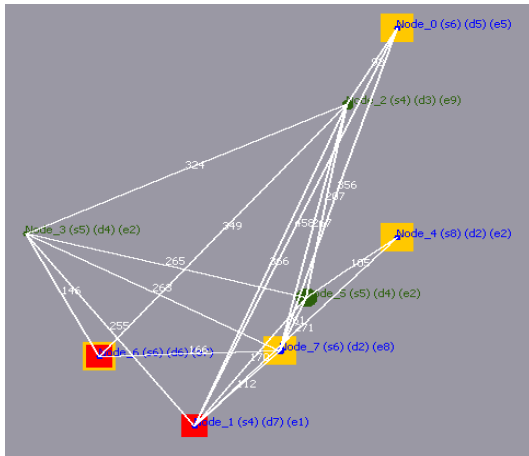
principles applied and exhibits the energy levels of the nodes. Table 5.16 defines the visualised parameters used to generate the undirected graphs for this case study.

Table 5.16. Visualised Security Graph Vulnerabilities, Parameters, and Energy Levels

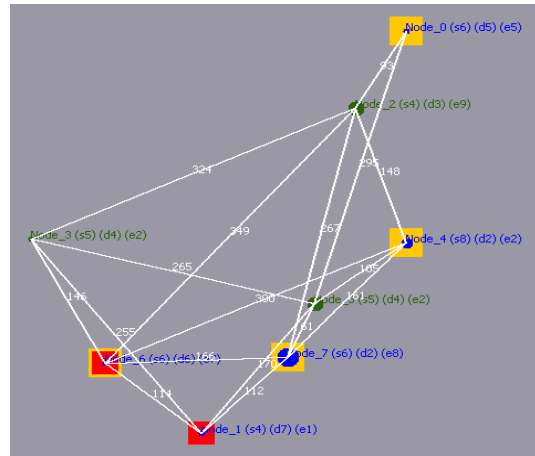
Graph	Parameter	Symbol	Description
All graphs	Scanned node no vulnerabilities		Dark green node/tag
	Scanned nodes identified vulnerabilities		Blue node/tag
	Unscanned node		Dark red node/tag
	Node size represents quantified bridging centrality, i.e. small nodes low and large nodes equal high.		
Security	Insecure node		Node encased with a solid orange box
	Blocked node		Node encased with a solid red box
	Blocked and insecure node		Node encased with a solid red box with orange border
Energy efficiency	High node energy level		Node encased with a solid green box
	Medium node energy level		Node encased with a solid orange box
	Low node energy level		Node encased with a solid red box

Conducting risk assessment on a Smart City is highly problematic, great consideration must be taken when applying methods directly to systems which are deployed or deemed critical, as these methods have the capacity to impact the collaborative components and affect their ability to function and meet objectives. We developed SCRAM as a safe testing environment to prevent our techniques from impeding physical Smart Cities and from introducing new risks to the topology.

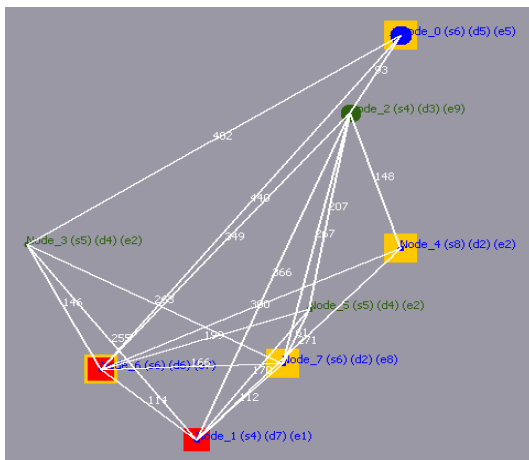
We incorporated risk assessment into the SCRAM framework in order to simulate vulnerability identification within these environments, and assign relevant reported NVD vulnerabilities to nodes, in a random method based on the device's hardware, software, and firmware. Simulating these vulnerability techniques means security scores are quantified with greater accuracy, SCRAM then generates detailed reports on all security parameters, centralities, and identified vulnerabilities, including security vulnerabilities with their associated CVSS v3 base scores. Thus, results and evaluations are more realistic and precise.



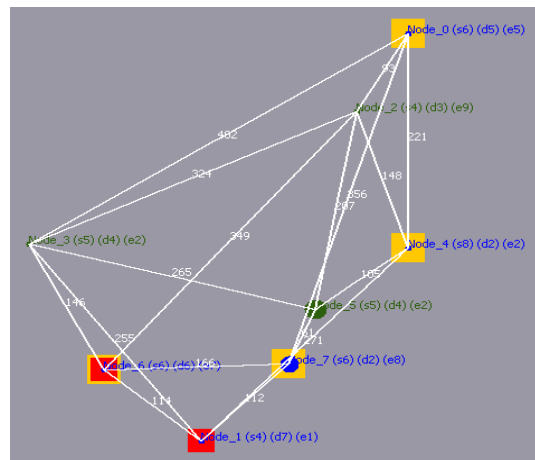
(a) First Mutated Candidate Using GA



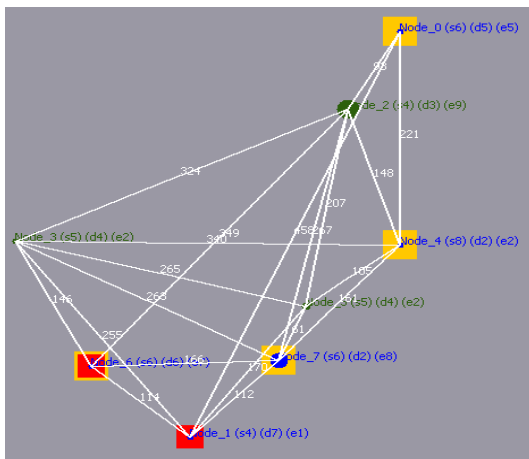
(b) Final Optimum Candidate Using GA



(c) First Mutated Candidate Using ANT

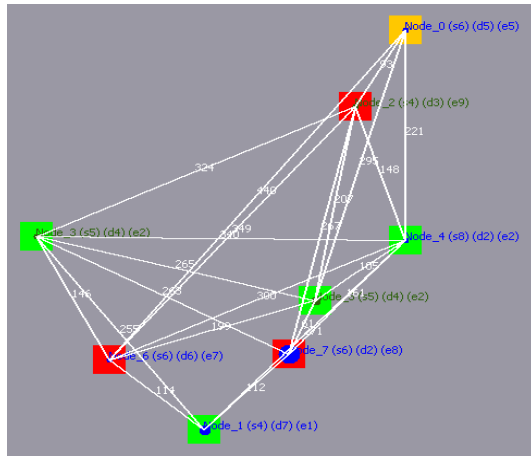


(d) Final Optimum Candidate Using ANT

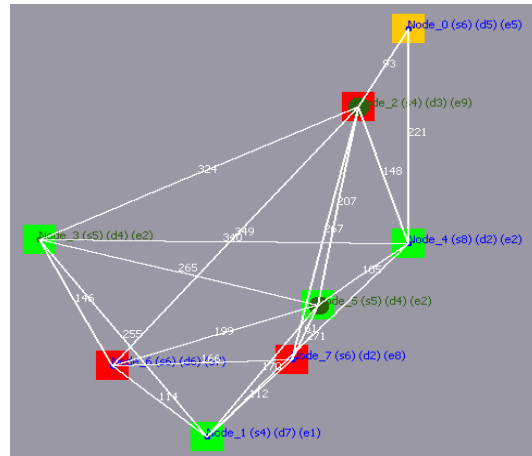


(e) First and Final Optimum Candidate Using TABU

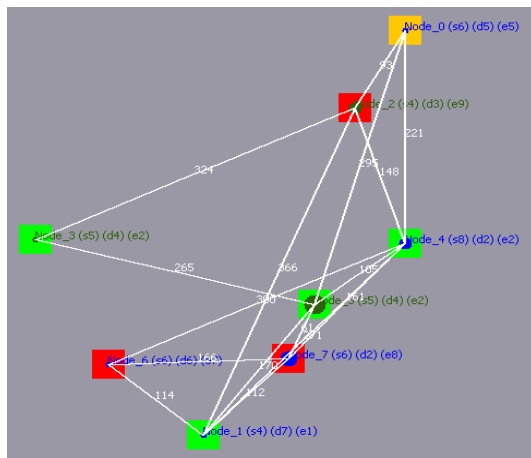
Figure 5.31. Comparison of Smart City WSN Security Risk Mitigation



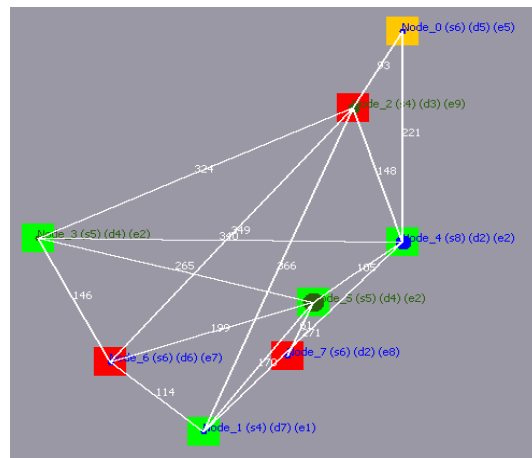
(a) First Mutated Candidate Using GA



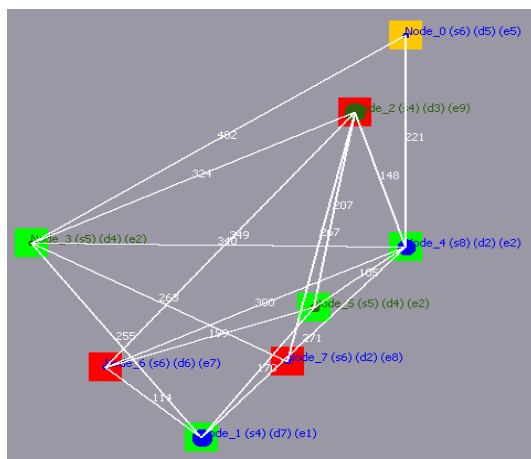
(b) Final Optimum Candidate Using GA



(c) First Mutated Candidate Using ANT



(d) Final Optimum Candidate Using ANT



(e) First and Final Optimum Candidate Using TABU

Figure 5.32. Comparison of Smart City WSN Energy Level and Security Risk Mitigation

As stated, the Smart City WSN section was generated within our SCRAM framework, and the outlined principles were applied to the network consecutively. When each algorithm is applied, it is integrated with the method's security risk mitigation process. The network is evolved by each algorithm into a set of best solutions as described in Section 4.6.4, Figure 5.31 and Figure 5.32 visualise the first evolved improved candidate and the final optimum solution for each of the applied

algorithms, as examples of how the network's security is reconfigured and network evolved in comparison to the original network topology.

5.4.3 Smart City WSN Robustness

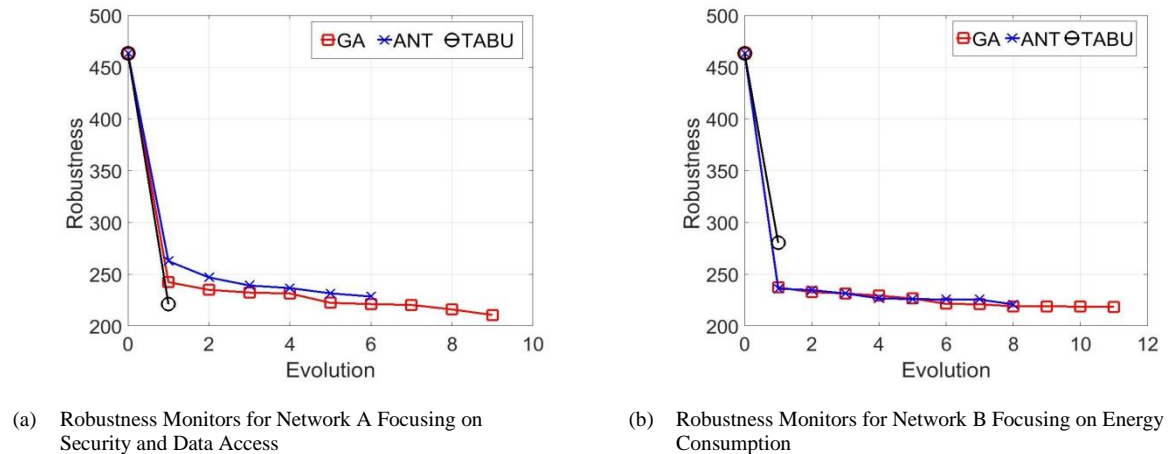


Figure 5.33. Comparison of Smart City WSN Robustness Graphs

Throughout the security risk mitigation process each reported candidate is measured by means of the robustness function (Section 4.6.3), and quantified using the five key parameters (security grade, highest bridging centrality, degree centrality, minimum path average, and cost). Emphasis is placed on the robustness level of the network as it assists the algorithms to produce the next generation of improved solutions, as it utilises the key parameters of individuals being selected. Other factors are also reported and considered such as the degree centrality of the graph and energy efficiency, and key parameters are also reported and analysed as standalone risks despite assisting in the quantification of the robustness score.

The robustness graphs in Figure 5.33 visualise network robustness when each of the algorithms was applied to both WSN states throughout all evolutionary risk mitigation processes. These graphs record a notable reduction in network robustness levels, for both WSNs A and B. When the algorithms were applied they randomly evolved new candidates in a positive method, meaning the reported improved solutions are more appropriate. The robustness monitor for WSN A (Figure 5.33-a) shows the original network had a robustness level quantified as 463.3917. When the GA based risk mitigation process was applied the final optimal solutions quantified robustness score was 201.5488, achieving a 56.51% improvement. Utilising the ANT based risk mitigation process; the final optimal solution's quantified robustness score was 228.368, achieving a 50.72% improvement, compared to the TABU based risk mitigation process which had a final robustness score of 220.9864 improving robustness by 52.31%.

The robustness for WSN B (Figure 5.33-b) was also quantified as 463.3917, in some simulations we see marginal fluctuations of difference between the original robustness scores because the framework

is quantifying the robustness focusing on different key parameters and incorporating energy as an additional risk. When the GA was applied to the network its final optimal candidate improved the robustness by 52.85% scoring 218.4797, ANTs optimal candidate scored 220.7172 improving robustness by 52.37%, while TABU reduced the robustness score of the network by only 39.52% scoring 280.2389. In both scenarios, robustness is improved from the first reported evolution for GA and ANT, ranging between 39.52% and 52.31%. This positive development continues to advance throughout the evolutionary risk mitigation process.

5.4.4 Smart City WSN Data Analysis

During evolution stages the applied principles search for an optimal combination, using processes that remove and replace links within the Smart City WSN, to mitigate risks and strengthen the security, in terms of securing links between nodes and enhancing security for the entire SoS. Figures 5.31-a, 5.31-c, 5.31-e, 5.32-a, 5.32-c and 5.32-e, visualise the first improved generations which assure communication security, each showing an increase of communication links. The cost increase (Figures 5.34-a and 5.35-a) for both scenarios reflects this growth of communication paths, with the applied algorithms increasing the cost of WSN A on average by 104.8% and WSN B by an average of 104.2%. It is essential that the security risk mitigation process when adding and removing links, balances connectivity with improvements to the WSN robustness and overall network security, while not unduly impacting centrality factors. The framework is not attempting to revise cost, simply associate network cost in terms of distance between nodes with suggested WSN modifications. WSN A which prioritises security and data access, shows that GA based security risk mitigation algorithm has the lowest costing optimal solution (Figure 5.34-a) increasing by only 98.04%. WSN B which prioritises energy levels, shows that ANT based security risk mitigation algorithm has the lowest costing optimal solution (Figure 5.35-a) resulting in an increase of 88.59%.

Through the improved robustness techniques, the algorithms and processes sustain low degree centrality (Figures 5.34-b and 5.35-b) for both scenarios. Comparing the analysis for degree centrality, for Network A (Figure 5.34-b) we note significant fluctuations for candidate scores for both GA and ANT; both optimal candidates decrease degree centrality by 62% to 0.1428. As the WSN is evolved WSN B (Figure 5.35-b) also exhibits fluctuations for degree centrality scores. The optimal candidate for GA and ANT both decrease their optimal solution's degree centrality score by 37.52% to 0.2380, despite both evolving different numbers of candidates with different robustness levels.

In both scenarios, while the network's optimal solutions do not have the lowest degree centrality score, each solution with the exception of TABU, has a reported improved centrality score compared to the original network. While degree centrality is not a key parameter used to quantify network robustness, as the algorithms process network evolutions, they reject evolved candidates that

critically increase degree centrality, i.e. minor negative increases are acceptable and considered in tolerable range.

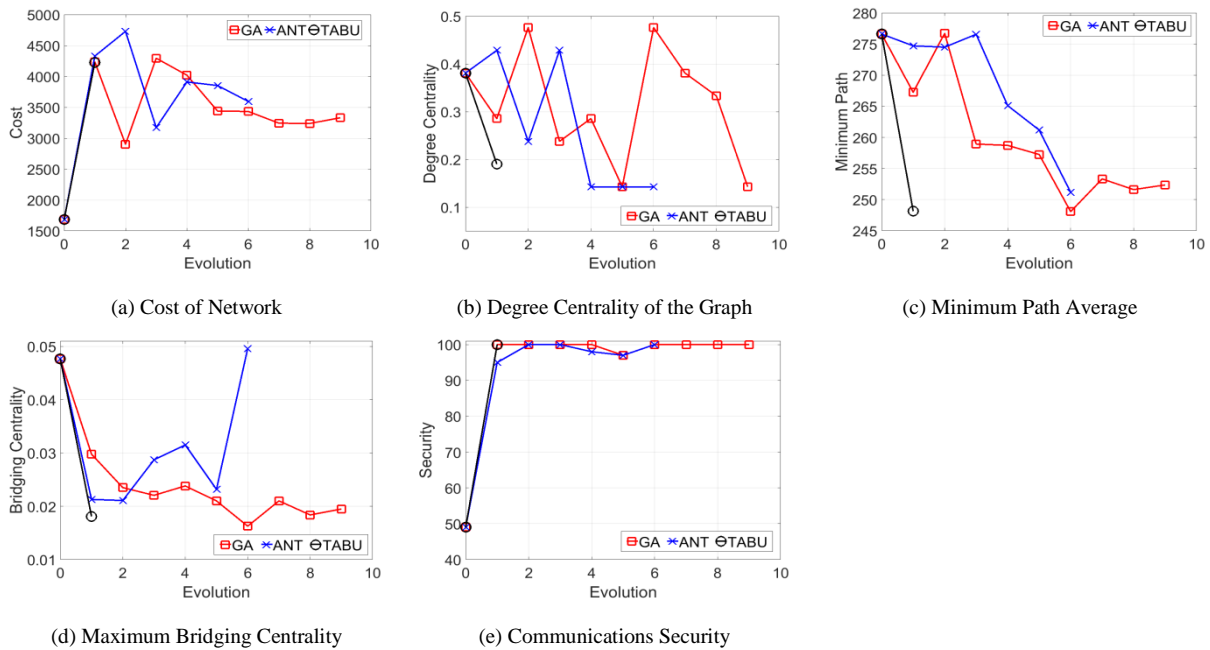


Figure 5.34. Network Evolution Security Results Comparison for WSN A

There are notable fluctuations between reported candidates for minimum path average (Figures 5.34-c and 5.35-c). In both scenarios, the only increase to minimum path average occurred when TABU was applied to WSN B, which focuses on energy levels. This negative increase is reflected in TABU's robustness score (Figure 5.33-b) which is slightly higher compared to the other algorithms' robustness scores. Minimal path average reduced by 24.25% using GA and 25.43% using ANT on WSN A, and by 28% using GA and by 15.29% using ANT on WSN B. These scores directly correlate to the new established links between nodes.

Analysing bridging centrality (Figures 5.34-d and 5.35-d), there are significant fluctuations between candidate scores for both scenarios and all algorithms. WSN A (Figure 5.34-d) indicates that the final optimum solution when ANT was utilised has a minor increase of 4.17% in comparison to the original network. In contrast to GA, which decreased bridging centrality by 59.09% and TABU which decreased by 62.03%. Analysing WSN B (Figure 5.35-d) each of the applied algorithms generated final solutions with decreased bridging centrality scores, GA decreased by 58.79%, TABU decreased by 56.1%, while ANT had the lowest decrease of 46.15%. Despite the single minor increase which is within a tolerable range, the analysis corroborates that as the WSN is evolved and algorithms applied, each of the methods support the security enhancement of the network, ensuring that evolutions that negatively impede security, robustness, and centrality factors are rejected. This is evident from not only sustained low centralities, but also the improvement to the robustness score.

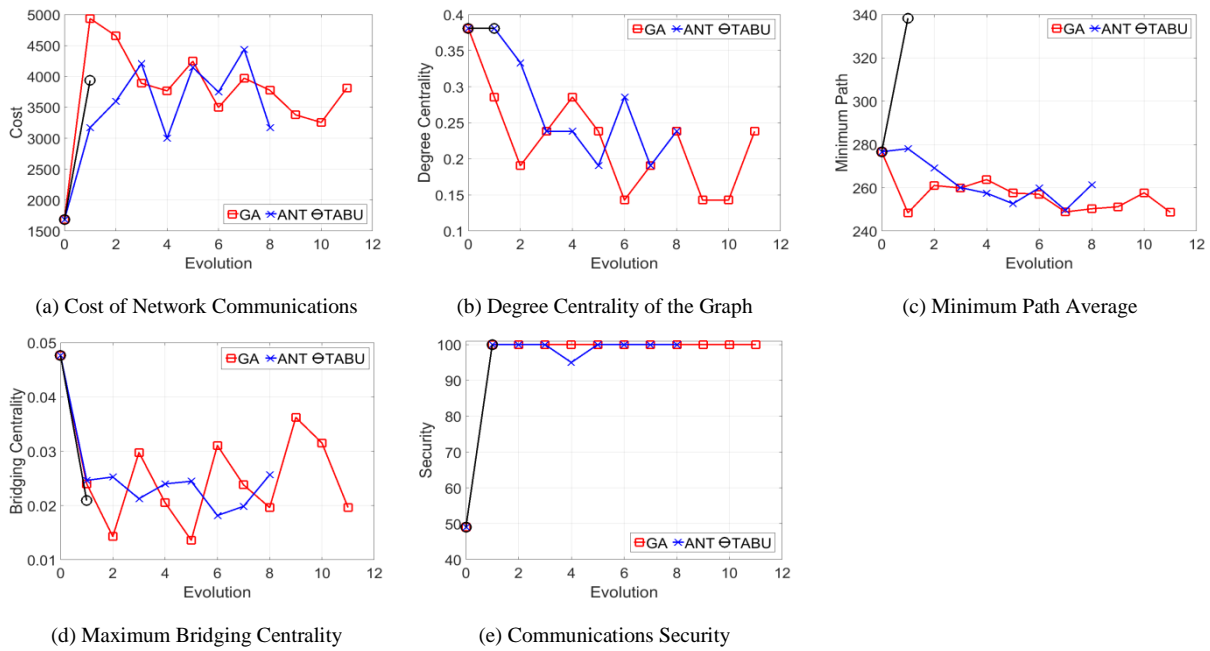


Figure 5.35. Network Evolution Security Results Comparison for WSN B

We observe for both scenarios there are significant increases to communication security from the first evolved candidate (Figures 5.34-e and 5.35-e), with minor fluctuations occurring from 95% to 100% for both networks. Each of the optimum solutions generated report 100% secure network communications, meaning each applied algorithm increased security by 51%.

Alternatively, via the use of the framework’s detailed reports we can evaluate each node and analyse how individual centralities are impacted due to network evolution, Tables 5.17 (WSN A) and 5.18 (WSN B) present aggregated node centrality scores for the primary, lowest costing, and optimum network candidates. Evaluating individual nodes assists in determining how distinct nodes are impacted compared to analysing the SoS as a single entity. An excerpt from a report is shown in Table 5.19, reporting data for Network A when GA was applied, showing in this instance the individual node bridging centrality scores for each enhanced reported candidate. This is vital should there be a requirement to monitor a specific critical node’s centrality values, and evaluate them prior to applying the recommended reconfiguration.

We ascertain that bridging centrality utilising GA in both scenarios decreased by over 58%. However, for WSN A average node bridging centrality only improved by 49.38%, while WSN B improved by 64.87%. In both scenarios utilising ANT, WSN A decreased bridging centrality by 63% yet the average node centrality only improved by 12.23%, while WSN B reported a 56.1% centrality improvement while its average node bridging centrality scored a 46.5% improvement. The report indicates that for all instances the values are in an acceptable range, but these reports provide an alternative means for analysing the SoS components and can be useful during any decision making processes.

Table 5.17. WSN A Security Evolution Results

Evolution	No. Links	Cost	Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0 (Primary Network)	12	1682	463.3917	0.428571	0.102679	0.000545	190.9844	0.01653
GA Evolution 2 low-cost	14	2901	234.8343	0.5	0.066964	0.000539	374.2656	0.006398
GA Evolution 9 optimum	17	3331	210.5488	0.607143	0.049107	0.000581	465.4844	0.008367
ANT Evolution 3 low-cost	16	3172	239.0152	0.535714	0.0625	0.000539	401.4844	0.007346
ANT Evolution 6 optimum	17	3595	228.368	0.607143	0.058036	0.000584	488.6406	0.014509
TABU Evolution 1 optimum	20	4225	220.9864	0.714286	0.035714	0.000592	686.5938	0.005139

Table 5.18. WSN B Security Evolution Results

Evolution	No. Links	Cost	Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0 (Primary Network)	12	1682	463.3917	0.428571	0.102679	0.000545	190.9844	0.01653
GA Evolution 10 low-cost	18	3255	218.6863	0.607143	0.058036	0.000572	462.75	0.012002
GA Evolution 11 optimum	20	3813	218.4797	0.678571	0.044643	0.000592	609.0156	0.005807
ANT Evolution 4 low-cost	16	2998	226.5958	0.535714	0.058036	0.000568	376.9063	0.011908
ANT Evolution 8 optimum	16	3172	220.7172	0.535714	0.066964	0.00056	433.2813	0.008844
TABU Evolution 1 optimum	16	3935	280.2389	0.571429	0.058036	0.000314	544.1094	0.009284

Additionally, these reports help ascertain the values for the optimum solution, and can identify if there are cheaper alternative candidates to implement that don't impact centrality values, robustness, and security, identifying more suitable alternatives than the reported optimum solutions. Reviewing TABU results, there are no cheaper alternatives to consider, due to the algorithm's rigid methods failing to yield alternative enhanced solutions. When GA and ANT are applied to both WSNs, cheaper candidates to implement are reported, which improve both the network's robustness and security, compared to the original Smart City WSN.

Table 5.19. Network A Enhanced Candidates Bridging Centrality Scores

	node 0	node 1	node 2	node 3	node 4	node 5	node 6	node 7
Evolution0	0	0	0.046584	0	0	0.047619	0.03004	0.007996
Evolution1	0	0	0.017007	0	0	0.029762	0	0.012755
Evolution2	0	0	0.008641	0	0	0.023496	0	0.019048
Evolution3	0	0	0.014098	0	0	0.022046	0	0.013825
Evolution4	0	0.011905	0.02381	0	0.008818	0	0.020089	0.010933
Evolution5	0	0.019841	0.015306	0	0.006494	0.010204	0.006494	0.021008
Evolution6	0	0	0.015571	0	0	0.016254	0	0.006342
Evolution7	0	0.009398	0.013255	0	0.021008	0.019133	0	0.019305
Evolution8	0	0	0.010504	0	0	0.014006	0.010302	0.018367
Evolution9	0	0	0.015306	0	0.011161	0.014778	0.006211	0.019481

However, just because cheaper alternatives are established, they should only be considered if they maintain a series of alternative links between secure nodes and 100% security, thus results are compared in conjunction with the undirected graphs.

5.4.5 Smart City WSN Observations

Analysing the undirected graphs that focus on security and data access (Figure 5.31), we intuitively identify in Figure 5.31-a, that the first evolved candidate produced using GA increased the number of links, increasing from 12 to 17, ensuring that a safe route was established between all secure nodes. Figure 5.31-b shows that as the WSN evolved further, a secure route between nodes 2, 3, and 5 was maintained using 17 links. Analysing ANT we identify that the first candidate increased the number of wireless links (Figure 5.31-c) from 12 to 19, and the optimum solution (Figure 5.31-d) maintained a secure route between nodes and is formed using 17 links. Figure 5.31-e visualises the only candidate produced using TABU, this algorithm establishes a secure route between nodes using 20 links, which is greater than solutions generated via GA and ANT. Reviewing the undirected graphs that focus on energy efficiency (Figure 5.32) we see similar characteristics.

For each final optimum solution for WSN A (Figures 5.32-b, 5.32-d, 5.32-e), we intuitively see that all candidates have multiple links between secure nodes, meaning if a secure link was removed, a single secure route will be maintained. This reduces the risk of single point of failures, and ensuring that nodes are unlikely to become isolated and cut off from the remainder of the WSN. Should multiple secure links be removed, there are alternative insecure communication paths between secure nodes. However, data will have to traverse via nodes which have been quantified as insecure placing the data at risk. Fortunately, these links have been identified and reported via the method, and visualised in the undirected graph, providing advanced warning and an opportunity to make changes to improve the security of these nodes prior to vulnerabilities being exploited or risks impacting their operations.

Likewise, final optimum candidates for WSN B (Figures 5.32-b, 5.32-d) identify significant links maintained between high energy nodes. In Figure 5.32-d there is only a single path between secure nodes. Should a single node or link be removed, then there are no secure paths for data to traverse, and data will be transmitted across paths between insecure nodes.

For WSN B the priority of the principles and algorithms was to quantify and enhance the security of the WSN prioritising node energy efficiency, as well as to maintain low centralities, high network security, data access violations, and node vulnerability. While this has been achieved, due to the method's prioritisation of energy efficiency there is a lack of alternative paths between secure nodes that are present within optimum candidates of WSN A. Which is expected as the method's priority is

shifted from network vulnerabilities and data access. Figure 5.32-b is the only exception, the optimum solution utilising GA shows there are multiple links between nodes 2, 3 and 5, therefore if a single node or link was removed nodes can maintain a secure path for data to be routed. We perceive that the applied algorithms and principles adequately support network security enhancement based on energy efficiency and can succeed in extending network life, evident from our initial simulation results.

In WSN, while the data access control problem would be less likely to be a priority over energy efficiency, we aim to improve data flow security. Implementing the new methods to focus on energy efficiency we see unstructured behaviour forming for both GA and ANT. This is due to the security risk mitigation process focusing on the energy efficiency levels and combining security and data access grades into the algorithm's process. As random evolvments occur while the algorithms are prioritising node energy levels ensuring that high energy nodes stay linked in case low level nodes fail, the algorithms still have to ensure that, as alterations occur through the network, security and data access control is maintained.

While TABU ensures a quick and non-costly process, completing its run in 38 seconds compared to GA which completed in 1 minute and 4 seconds and ANT that completed in 45 seconds, it fails to report or consider any alternative solutions that are slightly inferior, and only improved solutions are developed further. This is due to its restricted comparison parameters that must be matched or improved. The tabu list influences cycles preventing reverse evolvment from being considered in order to improve processing time and costs, but again analysis corroborates that other configurations could be appropriate.

Should organisations have financial restrictions in regards to network security, because the framework did not only just present the optimum solution but alternative candidates utilising GA and ANT, these alternative evolvments can be considered for adoption, in the awareness that the framework has mitigated risk, enhanced security, and improved the overall robustness of the network. These evolvments and recommended improvements assure network security and reduce potential risks to data communications and the SoS.

While new communication links help to establish secure routes across the Smart City WSN, as well as supporting node connectivity, they negatively impact network security as they are the basis for additional risk factors. In addition, these new communication links come at a price, as in order to achieve improved network robustness and lower centralities, there is a significant increase to network communication costs.

5.4.6 Smart City Sectors Simulation Evaluation

We generated six different simulations which reflect sectors within Smart Cities, each of which is based on WSN or IoT topology. Figure 5.36 visualises these six sections in a series of undirected graphs, which we have experimentally tested by applying the principles and algorithms discussed in Section 4. These graphs visualise each implemented and tested network's node energy efficiency levels, which not only observes the data access control problem, security levels and identified vulnerabilities, but also focuses on reconfiguring the network during the security risk mitigation process considering each node's energy efficiency level, in an effort to both extend the life of the network and enhance SoS security.

These Smart City sections consist of various devices which include sensors, smart devices, mobile phones, and computers, with differing communication links. Each infrastructure contains 8, 10, or 12 nodes, with a low connectivity level of either 30% or 40%. Individually, the simulated city sections were randomly assigned the relevant node software, hardware, and firmware parameters, comprising of type of device they would represent, operating system installed upon the node, if Internet access, encryption, firewalls, IDS, and anti-virus or security is supported, and if the node is updated or contains vulnerabilities, etc. The SCRAM framework also assigned nodes with a data access level, security grade, energy efficiency level, and then connected them via a series of primary links. Each scenario was then imported back into SCRAM and we applied the GA based risk mitigation algorithm and the ANT based risk mitigation algorithm to each scenario consecutively.

For these investigations we did not utilise TABU as we have ascertained it does not yield adequate results or report alternative security enhanced candidates. In each instance, we prioritised energy efficiency as part of the security risk mitigation process, after initial simulation results showed great capacity for security enhancement, and in an attempt to extend the network life in Smart City scenarios. Figures 5.37-a (GA) and 5.37-c (ANT) visualise each of the network's populations' robustness during the entire evolutionary process. These graphs clearly indicate a notable reduction on the network robustness for all scenarios, corroborating that all final optimum solutions are more appropriate as their robustness levels are quantified lower.

Similar to the above discussed case study, when we analyse the evolution results in Table 5.20 and Figure 5.37 we ascertain that GA produced more evolved candidates for analysis, and for all six scenarios GA generated enhanced evolved optimum candidates with lower robustness scores in contrast to ANT.

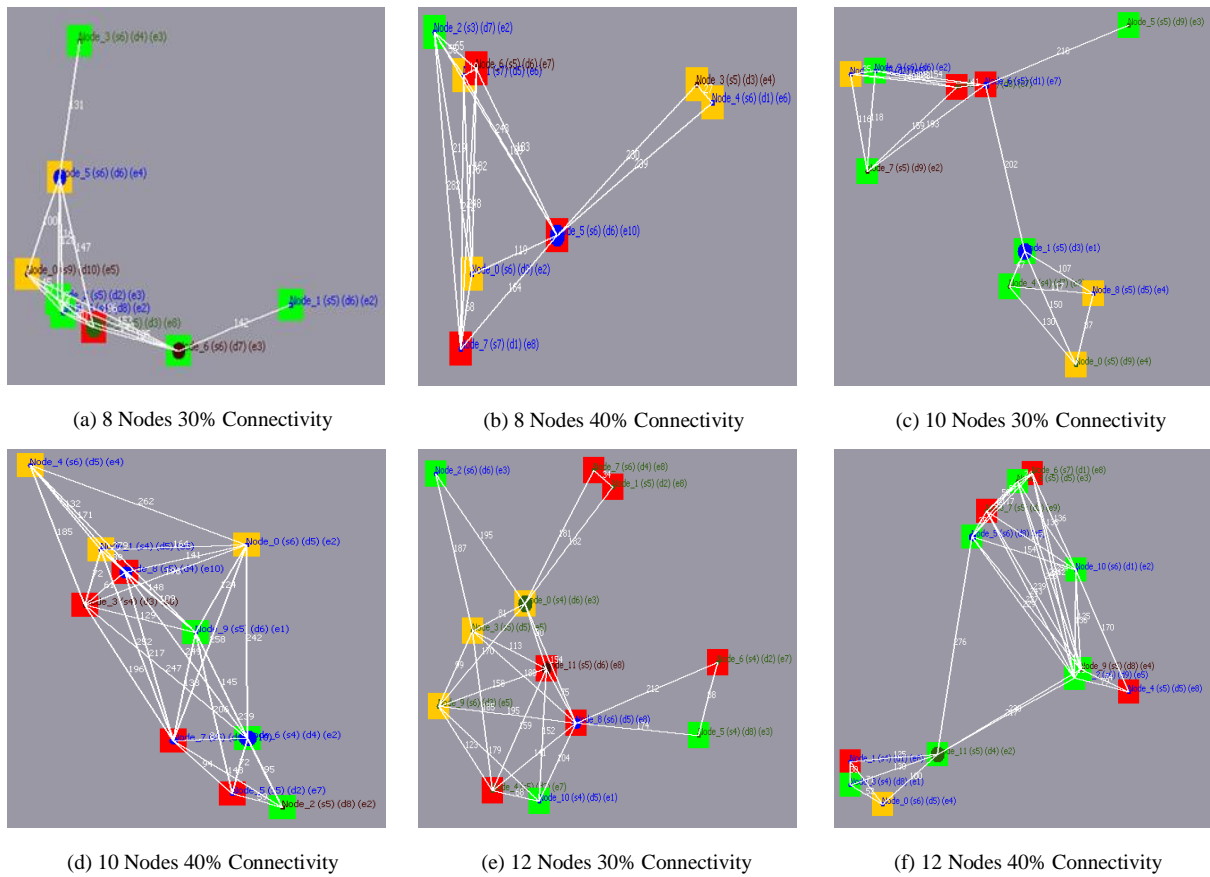


Figure 5.36. Simulated Smart City Networks

When GA is applied to Smart City A (Figure 5.36-a) the network's robustness improved by 24.97%, which is 0.4% more than ANT, and when GA was applied to Smart City D (Figure 5.36-d) the network's robustness improved by 13.28% which is 4.13% greater than ANT. On average GA had a 1.4% better optimal robustness score for scenarios in comparison to ANT. Each of these mutated optimal solutions not only increases the robustness of each scenario's topology, but also increases the network's communication security visualised in Figures 5.37-b and 5.37-c.

While we see minor fluctuations in network security for Smart Cities B and F when both GA and ANT was applied, when we analyse all instances, the applied algorithms advance security from the first reported improved candidate and maximise communication security for each optimum evolution. For both applied algorithms, after the first reported candidate fluctuations in security never drop below 97%, and only 4 of the evolved improved network candidates report a security score that does not equal 100% as evidenced in Figures 5.37-b and 5.37-d.

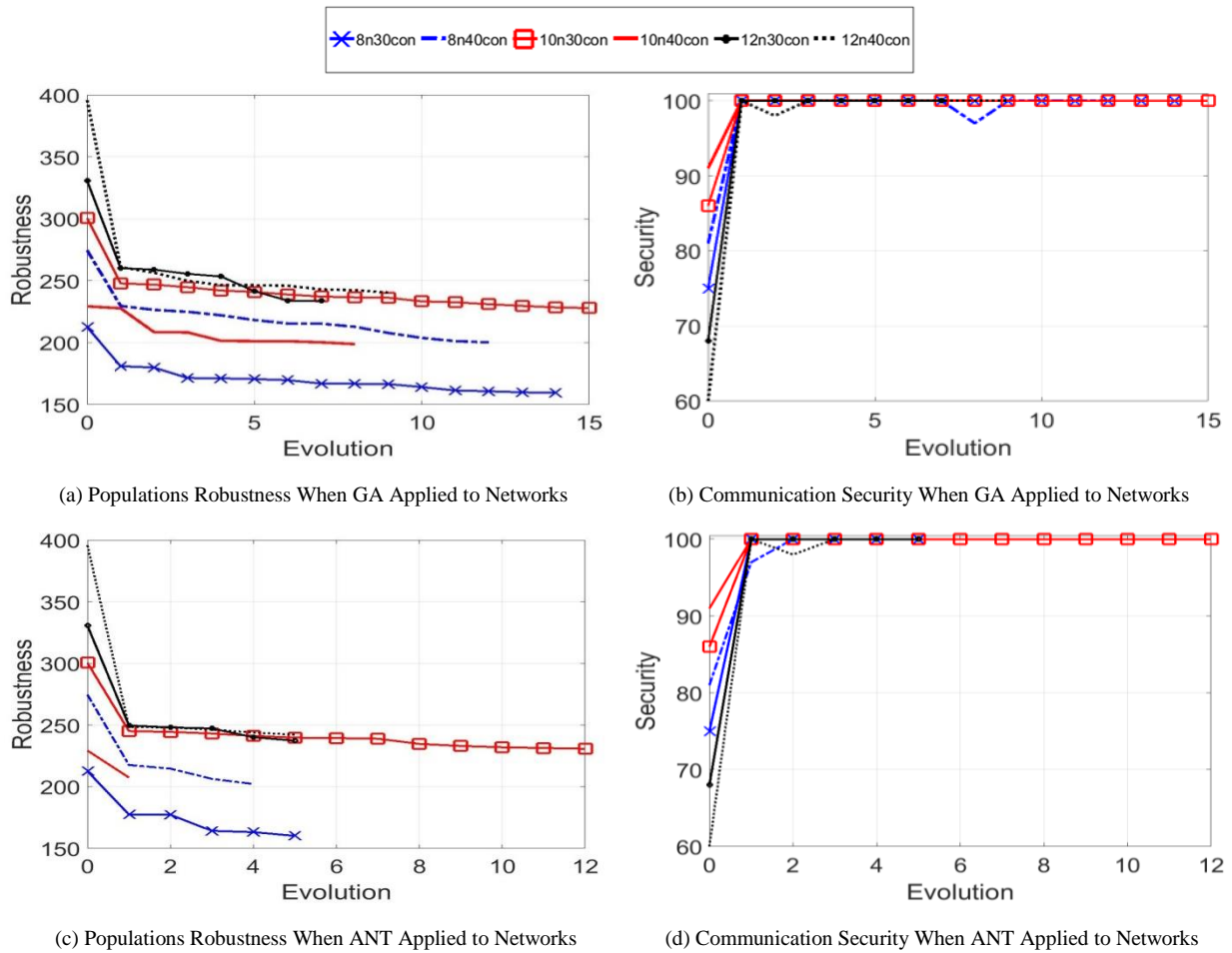


Figure 5.37. Simulated Smart City Sector Robustness and Security Comparison

While the replacement and removal of communication links balances connectivity with advances to security and robustness, these improvements again impact the overall cost of the communication network. In some instances we note that evolution can decrease or cause minimal cost increases, while in these instances we note a considerable increase to the overall network’s cost (Table 5.20). When GA and ANT was applied to Smart City D (Figure 5.36-d), both processes reduced the cost of the network from 5153 to 3869 (GA) and 4768 (ANT), and when GA was applied to Smart City B (Figure 5.37-b) this instance also generated a low costing network reducing costs from 2950 to 2899, while ANT increased the cost of communication to 3293. Similarly, through the analysis of the reported evolvments for each network, there were alternative cheaper reported evolved candidates.

Analysing the degree centrality for the simulated Smart City environments, we ascertain that the applied algorithms during the security risk mitigation process have evolved the networks and selected only configurations that lower and maintain low degree centrality with the exception of Smart City F (Figure 5.36-f) when GA was executed. After the GA was applied to the network, degree centrality had increased from 0.272727 to 0.290909, which is a 6.67% increase. While we are attempting to improve and maintain low centralities, this reported increase in this instance is adequately low and its value is in an acceptable range, as the candidates have an improved robustness score, security level,

minimum path average, and bridging centrality. These reports provide sufficient data and initiate warnings, so minor fluctuations and increases are thoroughly reported and identified to assist with all decision making processes.

Table 5.20. Simulated Smart City Security Evolution Results

Evolution	Cost	Robustness	Degree	Min Path Average	Bridging	Security
8 node 30% connectivity (Network A)	1664	212.5835	0.190476	192.8214	0.028571	75
GA Evolution 14 Optimum	2801	159.5087	0.095238	181.7143	0.019841	100
ANT Evolution 5 Optimum	2745	160.1424	0.095238	181.9286	0.022321	100
8 node 40% connectivity (Network B)	2950	274.5589	0.47619	253.25	0.02551	81
GA Evolution 12 Optimum	2899	200.2066	0.047619	241.6429	0.030612	100
ANT Evolution 4 Optimum	3293	202.1708	0.285714	231.5	0.017375	100
10 node 30% connectivity (Network C)	2317	300.4715	0.333333	296.6	0.085714	86
GA Evolution 15 Optimum	5416	227.9408	0.111111	264.7778	0.015649	100
ANT Evolution 12 Optimum	3628	230.7374	0.305556	269.9778	0.037037	100
10 node 40% connectivity (Network D)	5153	229.3223	0.361111	219.9778	0.016354	91
GA Evolution 8 Optimum	3869	198.8707	0.194444	233.6222	0.020779	100
ANT Evolution 1 Optimum	4758	207.4377	0.138889	231.9778	0.025641	100
12 node 30% connectivity (Network E)	3669	330.5918	0.4	275.106	0.023428	68
GA Evolution 7 Optimum	6980	233.8506	0.163636	263.9394	0.015873	100
ANT Evolution 5 Optimum	5939	237.3287	0.327273	272.394	0.017863	100
12 node 40% connectivity (Network F)	4783	395.7888	0.272727	271.697	0.043512	60
GA Evolution 14 Optimum	6763	240.4815	0.290909	270.7121	0.014606	100
ANT Evolution 4 Optimum	6113	242.2783	0.236364	283.1818	0.018939	100

Minimum path length for each of the optimum solutions reported in Table 5.20; demonstrate that the applied processes have assisted in evolving each of the networks and ensured that only candidates that improve the network or maintain centralities (i.e. centralities considered with an acceptable range) are selected as suitable reported candidates. In all but three instances minimum path average is reduced. When we analyse Smart City D (Figure 5.36-d), we note that when both algorithms were applied minimum path average increased from 219.9778 by 6.2% using GA and by 5.46% using ANT. Similarly, when ANT was applied to Smart City F (Figure 5.36-f) there was a notable increase to minimum path average, increasing by 4.23% from 271.697 to 283.1818. Again, this small increase in comparison to the evolved candidate's improved robustness, security, and other centrality scores, is in an acceptable range. Due to network enhancements we cannot guarantee that evolvments will not negatively impact centrality scores, what is evident is that the algorithms and processes are ensuring that only acceptable negative centralities are considered as part of the wider evolvment process and robustness evaluation.

The evolution of the communication links within each scenario greatly influences bridging centrality, and throughout all evolutions for each network we noted fluctuations of bridging centrality scores, which is expected due to the removal and replacement of communication links. In all instances with the exception of Smart City B (Figure 5.36-b) when GA was applied and Smart City D (Figure 5.36-d) when both algorithms were utilised, we see a decrease in bridging centrality for all optimal evolutions. The applied algorithms and processes when establishing secure communication links between nodes are influenced by the security score of the node and data access control. The mutated networks reflect the decisions of the applied algorithms and processes, along with the positions of the nodes within each of the network's topologies and the communication links which nodes are reliant upon for data transfer. While Smart City B (Figure 5.36-b) increased bridging centrality by 20% when GA was applied, and Smart City D (Figure 5.36-d) increased bridging centrality by 27.06% utilising GA and by 56.79% utilising ANT, the new mutated path structure ensures that there are an adequate number of secure links between secure nodes for data to traverse, and that communication security has increased and robustness levels have been positively improved, along with maintaining centrality values that are within acceptable ranges.

5.5 Summary

This chapter has conveyed the implementation of the SCRAM application, and the associated principles and techniques that have been incorporated into the framework. It also presents details regarding the operation of the simulated environment that is used to evaluate the applied theoretical principles, and discusses the vulnerability methods incorporated into SCRAM in order to better quantify the security of the SoS being enhanced, which directly improves the applied algorithm's methods. In addition, we have presented a case study in order to demonstrate the ability of the implemented theoretical principles to be extended and applied to monitor and incorporate node energy levels into the security risk methods, in an attempt to extend the network life of SoS alongside mitigating risks and securing the SoS.

Chapter 6

Multi-Level SoS Security Analysis and Evaluation

As discussed in Section 5, the proposed theoretical principles corroborate that an evolutionary approach to network security is practical. In this chapter, we extend this work further and examine the security challenges of connecting and reconfiguring communication links between multi-level SoS, and discuss the associated issues of security enhancement on individual SoS prior to integration, compared to mitigating risk and enhancing the security of a multi-level SoS as a single entity.

In order to prove the effectiveness of the proposed theoretical techniques and evaluate the applied method's usefulness when applied to large dynamic multi-level SoS, we evolved the SCRAM framework to simulate a significant number of realistic ICT devices and their associated vulnerabilities. Using SCRAM, accurate multi-level SoS are generated, which once constructed have the implemented theoretical principles presented in Section 4 applied in order to reconfigure and improve multi-level SoS security, along with generating realistic data. All the experiments detailed in this section were generated and conducted within the SCRAM framework, representing specific infrastructures and their characteristics. The collated data from these scenario specific experiments is analysed and presented in this section.

6.1 Multi-Level SoS Security Challenges

One of the most important aspects of SCRAM is its ability to reconfigure the SoS network(s) searching for the optimum structure in order to strengthen and secure network communication, and identify vulnerabilities within the topology that have the potential to expose the entire collaborative infrastructure, in order to mitigate the associated risks. When we begin to connect SoS together we have to consider the security of the connecting nodes between the distinct networked SoS, the impact on its bridging centrality, and whether the node is being forced to connect with a vulnerable and insecure node, or a blocked node that violates data access policy principles.

Figure 6.1 presents three differing SoS environments that will be joined to form a larger dynamic multi-level SoS. Each networked environment is composed of multiple device types, assigned the appropriate parameters, and connected via a series of primary links. Each distinct SoS was first

security enhanced to mitigate risks, prior to establishing collaborative communication links between the distinct infrastructures forming the multi-level SoS. Table 6.1 presents a comparison of the initial security assessment generated by the framework, compared to the new evolved results for each of the SoS.

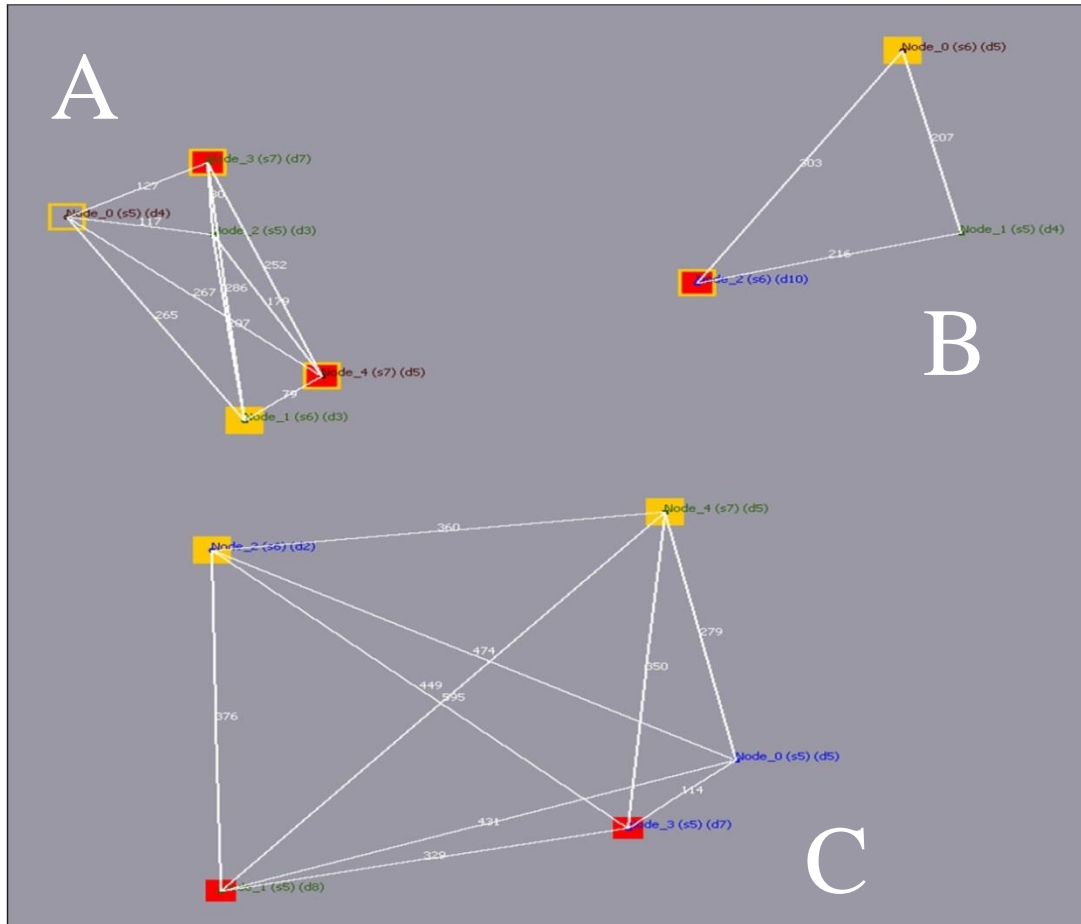


Figure 6.1. Reconfigured Secure Simulated Smart City Networks

These results determine that security has been maintained or improved, centrality factors have been marginally reduced, the robustness levels of the networks have decreased meaning the network's reconfiguration is more appropriate, minimum path average has been reduced, while these improvements are directly reflected by the increase to network cost.

Establishing the communication path connection rules for collaboration is problematic, while it would be simple to enforce a rule that guaranteed to only connect networks together via secure nodes to ensure that security is not jeopardised. This simplistic process could place all of the collaborative networks at risk. If this rule is enforced it could lead to single nodes being responsible for the entire secure communication routes between collaborative SoS. In addition networks might not have secure nodes that meet the security thresholds. Should single nodes and links be relied upon to maintain collaboration, then these devices and links become SPoF. As should a single link be removed or node

fail, then data transfer between networks would discontinue and the entire multi-level SoS could fail to meet objectives with varying consequences.

Table 6.1. Simulated Smart City Networks Security Evolution Results

SoS	Number of Nodes	Connection	Robustness	Security	Bridging Centrality	Degree Centrality	Cost	Minimum Path Average	Insecure/Blocked nodes (%)
A original	5	40%	161.95	100%	0.021	0.1667	1573	186	60% / 40%
A optimal	5	-	150.49	100%	0	0	1859	185.9	60% / 40%
B original	3	30%	265.67	100%	0.083	1	423	282	67% / 33%
B optimal	3	-	170.61	100%	0	0	726	242	67% / 33%
C original	5	30%	1038.34	58%	0.2	0.1667	1450	775.2	40% / 40%
C optimal	5	-	300.23	100%	0	0	3707	370.7	40% / 40%

It is vital to consider the risks that new communication paths pose to nodes with which they connect, as these new links while supporting connectivity introduce additional risk factors. Therefore, it is essential that we re-analyse the security grade of every node connecting to an external SoS.

6.1.1 Multi-Level SoS: Calculating Connecting Node Security Grades

While a collaborative relationship exists first we consider independently operated SoS that do not divulge their entire security risk analysis with their collaborative partners, instead share the security grades quantified by the principles incorporated into SCRAM. This simulates organisations that are reluctant and unwilling to share detailed vulnerability analysis of their SoS, in case those with malicious intent try to exploit them, for example, insider threats are just as problematic and as likely as malicious external attackers. As we know the source and reliability of each score there is no need to use a weighted average, however, we do see these new communication links as additional vulnerabilities so they are incorporated into the new quantified connecting node security score.

To determine the connecting node security score S , assume node N has an initial quantified security grade of G , with c connections collated based on the network discovery process. Denote the values of the c connections by g_1, g_2, \dots, g_c which are the assigned node security grades of the identified external connected nodes. Each communication path is assigned a risk probability score p , based on the type of node with which it connects (i.e. secure, vulnerable, blocked, or unscanned node), defined as:

$$S = \left(G + \left(\frac{g_1 + g_2 + \dots + g_c}{c} \right) + \left(\frac{p_1 + p_2 + \dots + p_c}{c} \right) \right) / 3 \quad (6.1)$$

In order to make the connecting node security score more accurate, we can combine this algorithm with the original equation presented in Section 4.6.1.4 which is used by SCRAM in order to assign the initial node security grade. By modifying this equation it allows us to quantify both the internal and external risks to the node more accurately, as these new communication paths and the nodes which they are linked with are significant vulnerabilities which expose them to risk. Again we assume that the external SoS are only sharing their security grades quantified by SCRAM and not divulging their entire device details and their identified vulnerabilities.

The connecting node security will only be quantified for a node if it has been identified as having communication links to external SoS. To determine the connecting node security score S , assume node N will have n vulnerabilities collated based on its risk assessment. Denote the values of n vulnerabilities by v_1, v_2, \dots, v_n which are the assigned values of the identified vulnerabilities [252].

Assume node N will have c connections collated based on the network discovery process. Denote the values of the c connections by g_1, g_2, \dots, g_c which are the assigned node security grades of the identified external connected nodes. Each communication path is assigned a risk probability score p , based on the type of node with which it connects (i.e. secure, vulnerable, blocked, or unscanned node), defined as:

$$S = \left(\left(\frac{v_1 + v_2 + \dots + v_n}{n} \right) + \left(\frac{g_1 + g_2 + \dots + g_c}{c} \right) + \left(\frac{p_1 + p_2 + \dots + p_c}{c} \right) \right) / 3 \quad (6.2)$$

For complete accuracy, connecting node security scores could be quantified utilising the equation presented in Section 4.6.1.4 which is used by SCRAM to assign the initial node security grades. However, this would require the SoS with which they are externally connected to share the complete node vulnerability parameters and all identified vulnerabilities, for every established communication link. In this situation some companies might be unwilling to divulge these facts, or in crisis situations permission to access these details might be time consuming and depend on the type of organisations collaborating and their data/security levels, etc.

6.1.2 Multi-Level SoS: Connecting Node Security Analysis

Figure 6.2 represents three distinct generated example multi-level SoS graphs, which visualise the necessary connections between the SoS networks. Table 6.2 depicts the visualised parameters used to generate the multi-level SoS undirected graphs. Other combinations of communication link placements are possible, and for these scenarios we assume that connecting nodes must be identified

as secure and have the permitted data access level. In addition, to prevent SPoF each network must have two different external connection points to ensure data transfer and maintain collaborative relations. If there is no suitable secure and ready node, then the least vulnerable node with the most appropriate data access level will be selected as an alternative connection point, blocked nodes are only considered if no suitable alternative is available.

When we apply Equation 6.1 to each of the connecting nodes, we see on the first connection Combination 1 (Figure 6.2-a), that 2 of the connecting node security grades have been quantified and identified as insecure and 1 has been re-categorised to secure. The connecting node security grade for node 0 in Network B has increased from 6 (insecure) to being quantified as secure with a security grade of 5. While connecting node 2 in Network A has been quantified as insecure with a new security grade of 7 compared to its original assigned grade of 5, and connecting node 0 in Network C has been quantified with a lower security being assigned a new score of 7 compared to its original security score of 5. All other connecting nodes have been re-quantified and identified as no change to initial assigned security grades.

Applying the advanced scoring technique (Equation 6.2) to the same SoS environment (Figure 6.2-b) SCRAM produces a more complex and precise security grade for each of the connecting nodes. The new security grade scoring identifies that only 2 connecting nodes have had their security grades altered, which are node 2 Network A and node 0 Network C, with all other connecting nodes security grades remaining the same, despite the introduction of new vulnerabilities and access points.











When utilising both scoring techniques we only apply the methods once, this is to prevent the scoring technique from entering into a loop. Combination 2 and Combination 3 show similar characteristics as Combination 1, with multiple nodes being identified as either more insecure or secure due to the new connections to external nodes. Combination 2 (Figure 6.2-c), indicates when the simplistic scoring technique was applied there are 2 new node security grade reassignments, and only 1 when using the advanced method (Figure 6.2-d). While Combination 3 (Figure 6.2-e), indicates that 3 connecting nodes have had their security grades reassigned, yet only 1 is reassigned when the advanced method is applied (Figure 6.2-f).

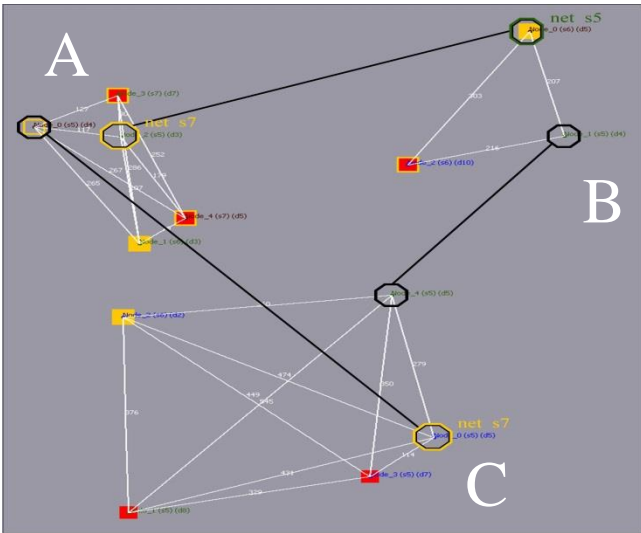
While one scoring technique is more accurate than the other, these SoS examples with the applied scoring techniques corroborate the impact and potential exposure that new connections can have upon a node. Incorporating all external nodes' security grades and by assigning connection parameters based on the status of the node being connected, it is possible to incorporate these parameters into the security grade assignment to accurately quantify the impact on security which these new connections will have.

In addition, as the networks were individually reconfigured, in order to determine the impact new connections have had upon the security and node centralities such as bridging, the entire multi-level

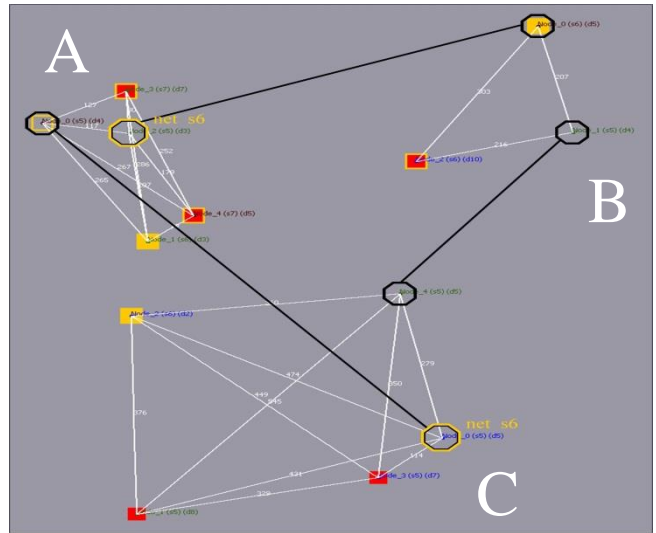
SoS would require additional risk assessment and potentially might need to be further security enhanced with the risks mitigation process having to be reapplied.

Table 6.2. SoS Visualised Security Vulnerabilities and Parameters

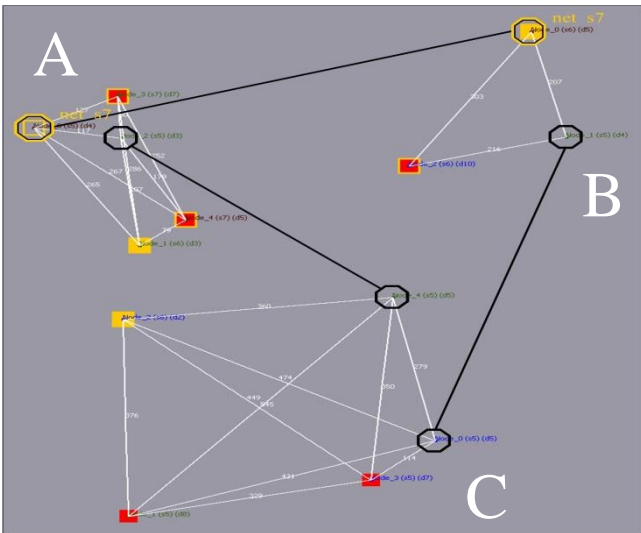
Graph	Parameter	Symbol	Description
All graphs	Scanned node no vulnerabilities.		Dark green node/tag.
	Scanned node with unresolved identified vulnerabilities.		Blue node/tag.
	Unscanned node.		Dark red node/tag.
Node size represents quantified bridging centrality, i.e. small nodes low and large nodes equal high.			
Security	Insecure node.		Node encased with a solid orange box.
	Blocked node.		Node encased with a solid red box.
	Blocked and insecure node.		Node encased with a solid red box with orange border.
	Node quantified secure and unscanned.		Node encased with a non-solid orange box.
Multi-Level SoS	External network connected to node, no change to connecting node security grade.		Node encased with a non-solid black octagon border.
	External network connected to node, negative change to connecting node security grade.		Node encased with a non-solid black and orange octagon border.
	External network connected to node, connecting node security grade remains secure.		Node encased with a non-solid black and green octagon border.



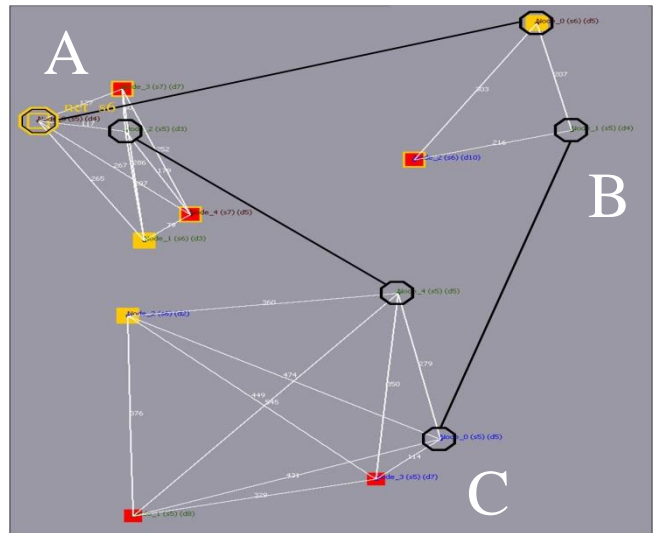
(a) Combination 1 with Simplistic Connecting Node Scoring



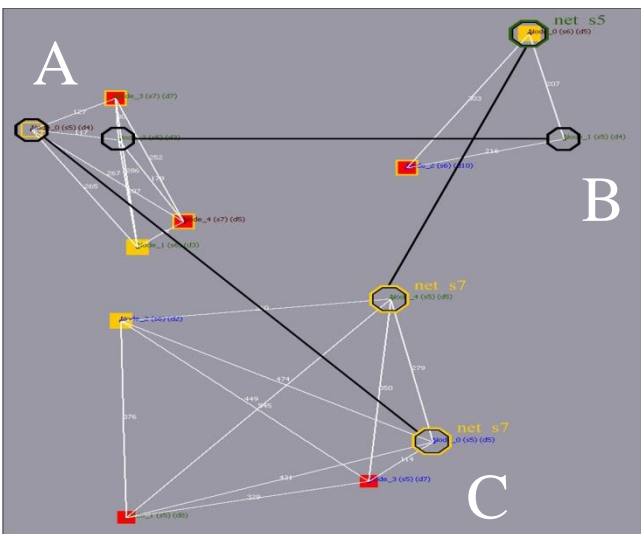
(b) Combination 1 with Advanced Connecting Node Scoring



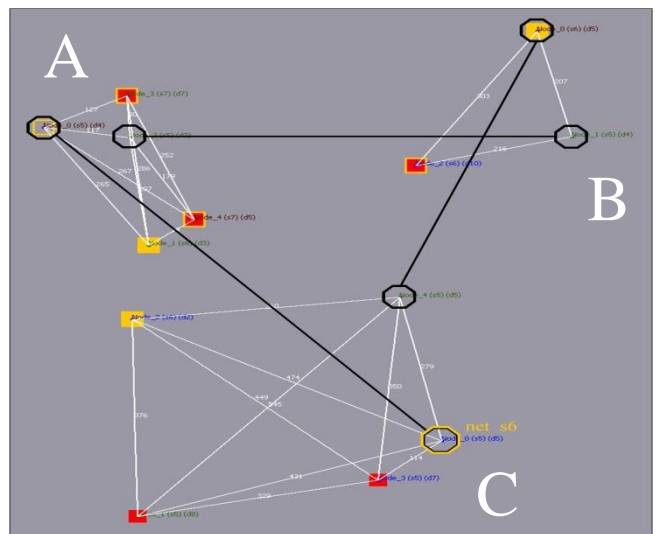
(c) Combination 2 with Simplistic Connecting Node Scoring



(d) Combination 2 with Advanced Connecting Node Scoring



(e) Combination 3 with Simplistic Connecting Node Scoring



(f) Combination 3 with Advanced Connecting Node Scoring

Figure 6.2. Simulated Smart City Networks

6.1.3 Multi-Level SoS: Calculating Connecting Node Security Grades

In order for the security risk mitigation process to be effective we have to consider both the impact of changing security due to external communication links and their associated risks, in addition to reconfiguring communication paths during the entire security enhancement process, not only for one SoS but for the multi-level SoS. Which allows us to identify the optimum configuration for all external connecting nodes, while configuring the internal network of each SoS to support these new links between distinct SoS. This will also ensure that we can critically analyse centralities such as bridging centrality, and limit the effects on the entire collaborative infrastructure security, along with reducing the introduction of additional and avoidable vulnerabilities.

Figure 6.3 provides an example of multiple distinct SoS which have been integrated together to form a multi-level SoS. This infrastructure has been generated within the SCRAM framework, and consists of four distinct SoS each containing four devices and has 30% connectivity. Each device has been assigned with the relevant node software, hardware, firmware, and vulnerability parameters, including being assigned a security level, data access grade, and connected to the other devices within its SoS via a series of communication links. In addition, each SoS is randomly connected to each other via a series of external communication links.

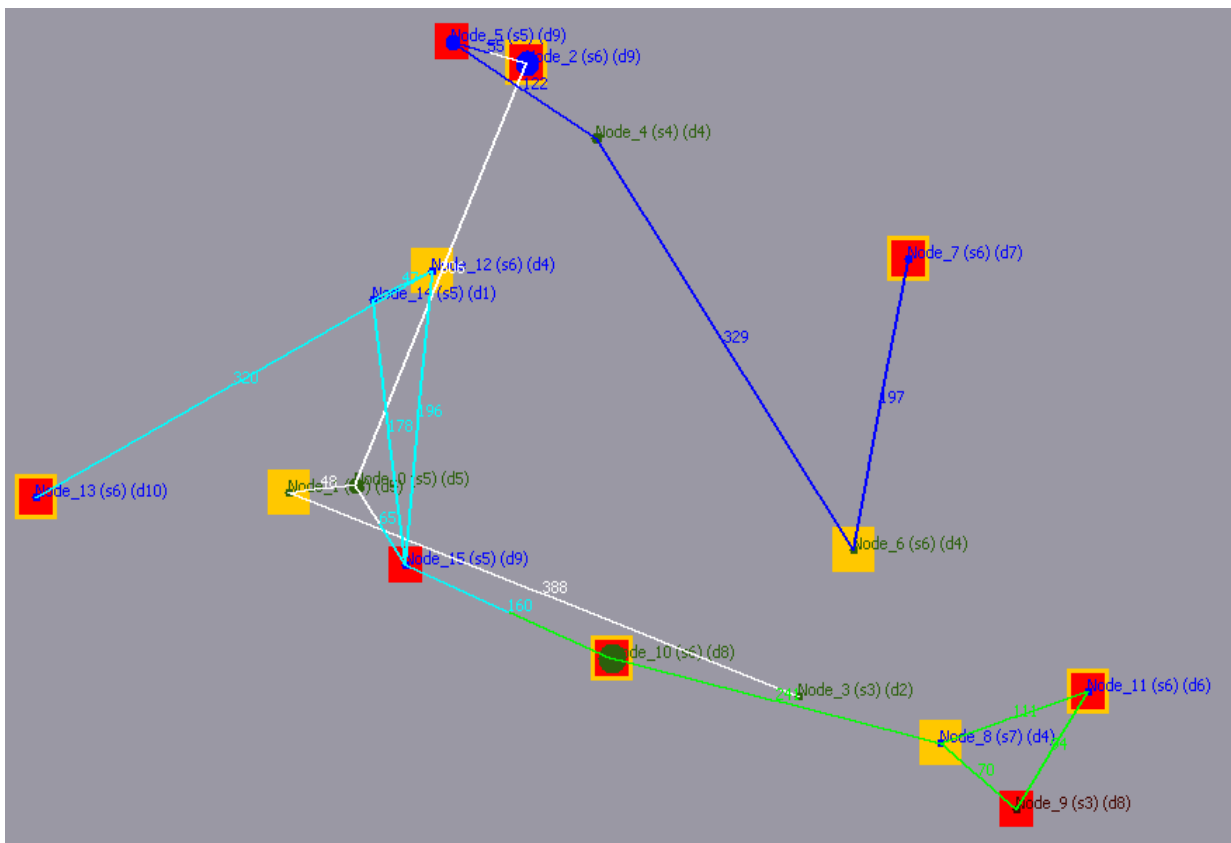
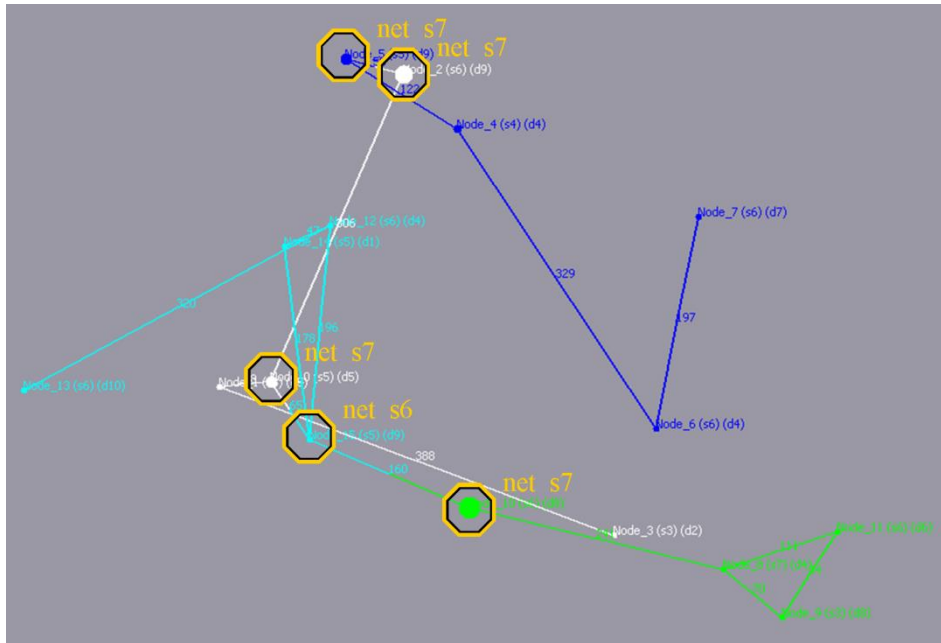
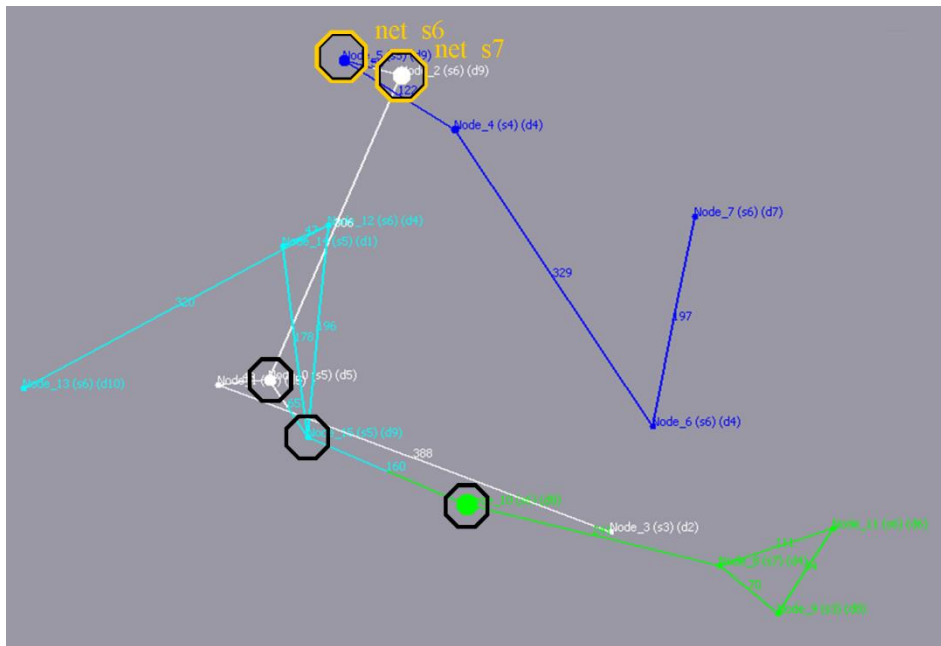


Figure 6.3. A Multi-Level SoS Example



(a) Multi-Level SoS with Applied Simplistic Connecting Node Security Scoring

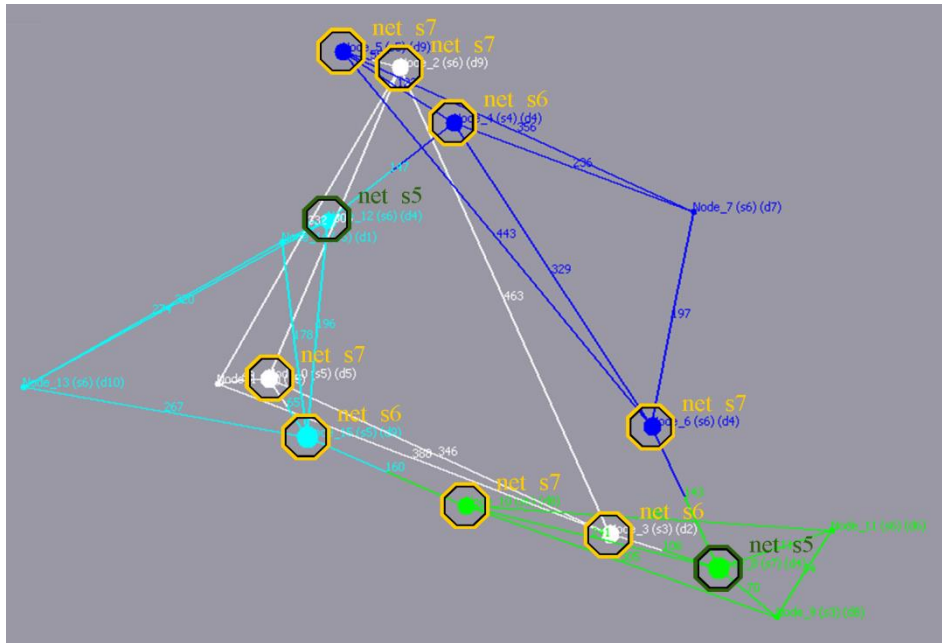


(b) Multi-Level SoS with Applied Advanced Connecting Node Security Scoring

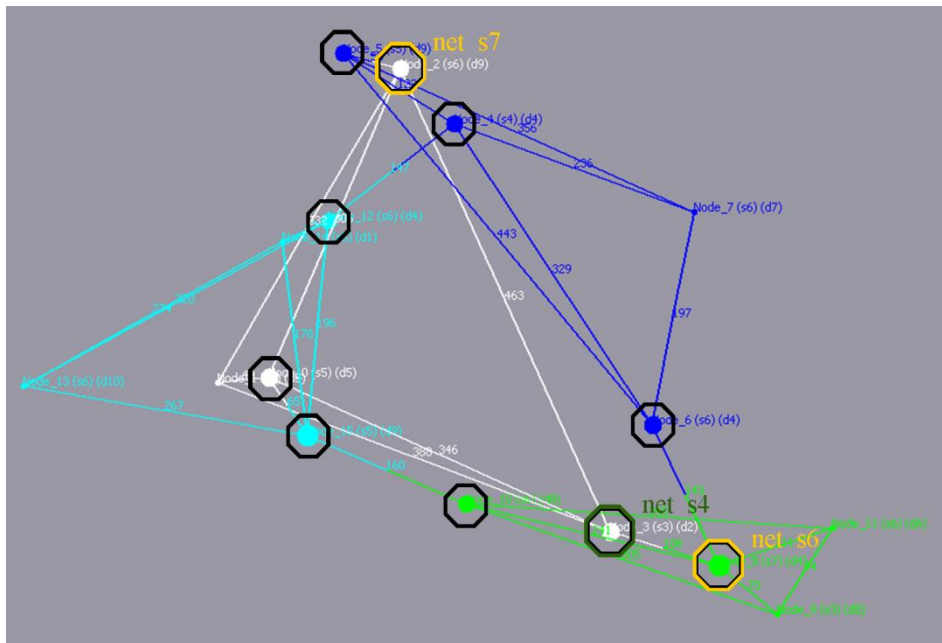
Figure 6.4. Multi-Level SoS Primary Connecting Node Scores

When we apply the simplistic connecting node security scoring technique (Figure 6.4-a), we can quickly see that all 5 connecting nodes have had their security quantified indicating that their scores have been negatively impacted due to the addition of external connecting SoS nodes. For example, Nodes 0, 5, and 15 have all increased from security grade 5 to grade 6, meaning these secure nodes are now deemed insecure. Nodes 2 and 10 have increased from grade 6 to security grade 7. While these grades were already deemed insecure, the higher grade reflects their added risk to the entire collaborative infrastructure.

Applying the more advanced connecting node security technique to the same multi-level SoS (Figure 6.4-b) it is established that only 2 of the connecting nodes are quantified as having their security scores negatively impacted, with node 2 increasing from security grade 6 to 7 and node 5 increasing from grade 5 to 6, reflecting the accuracy of the advanced scoring technique and the impact of the connecting links on the initial SoS configuration.



(a) Multi-Level SoS with Applied Simplistic Connecting Node Security Scoring



(b) Multi-Level SoS with Applied Advanced Connecting Node Security Scoring

Figure 6.5. Reconfigured Multi-Level SoS Connecting Node Scores

Within the SCRAM framework we apply the evolutionary principals to reconfigure the network and improve the multi-level SoS security and mitigate the associated risks. Figure 6.5 visualises the

security enhanced network for both of the connecting node security score techniques, and we intuitively identify that new communication paths have been established in order to guarantee a series of secure data paths between secure nodes across all of the SoS infrastructures. This is in compliance with the data access policies requirements, and reduces the risk of data being transferred via insecure nodes in any of the collaborative infrastructures.

The simplistic connecting node security score technique when applied (Figure 6.5–a) identifies that 8 nodes are now considered insecure, node 3 is insecure by an additional 3 grades decreasing from grade 3 to 6, nodes 0, 4, and 5 have increased by 2 grades, with nodes 0 and 4 now being re-categorised as insecure changing from grades 5 to 7 and 4 to 6 consecutively, and nodes 2, 6, 10, and 15 have all increased security grades by a single grade, with node 15 while being a blocked node is now considered insecure changing from security grade 5 to 6. In addition, nodes 8 and 12 have been re-categorised as secure, with node 8 being quantified as having a security grade of 5 from 7, and node 12 being quantified as scoring 5 decreasing from 6. In this instance, no connecting node has maintained their original assigned security grade.

Analysing the security grades when the advanced connecting node security score technique is applied (Figure 6.5–b), only 3 out of 10 connecting nodes have been quantified as having different security grades. Both nodes 2 and 8 have been increased by one grade and remain insecure, and node 3 has decreased by one grade but is still quantified as secure changing from grade 3 to 4. When we compare node 8 for both scoring techniques we note that the simplistic technique re-categorises the node as secure, while the advanced scoring method quantifies the node as insecure due to the method's accuracy. Demonstrating the need for not only reassessing connecting nodes' security when joined to a collaborative network, but also the requirement for a precise and in depth scoring technique, i.e. the more details and vulnerability parameters, along with the accuracy of risk scoring of those parameters means scores are more precise and reliable giving an insight into the true security standing of the network, ensuring that risks are not underestimated and overlooked due to poor scoring and identification.

6.1.4 Multi-Level SoS: Security Evaluation

The SCRAM framework, like the single infrastructure security assessment, produces in addition to the undirected graphs, reports on the multi-level SoS security and node centralities, network cost, minimum path average, and robustness of the entire multi-level SoS. For example, in Table 6.3 we can see that the reported degree centrality for the graph has reduced by 33.33% from 0.14285716 to 0.09523809, while communication security for the entire multi-level SoS has increased by 182.14% from 28% to 79%. Due to the large number of insecure nodes and blocked nodes forming the SoS, this is directly reflected in the overall score for network security.

Table 6.3. Multi-Level SoS Evolution Results

Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.4	4	4	30%	0.14285716	28%	2927	687.48334	50% / 50%
6.5	4	4	-	0.09523809	79%	6685	355.075	50% / 50%

As stated while it is ideal to only have connections between secure nodes, the security status and topology of the network are going to influence the optimum configuration for connecting communication links together. Likewise, to ensure there is enough built-in redundancy to limit nodes becoming isolated and data transfer failing within networked infrastructures and between distinct SoS due to node and link failures, collaborative links may have to be formed between insecure or blocked nodes. The addition of communication links is reflected in the increase to network cost, which has increased by 128.39%. Enhancing the network security has also resulted in minimum path average reducing by 48.35%.

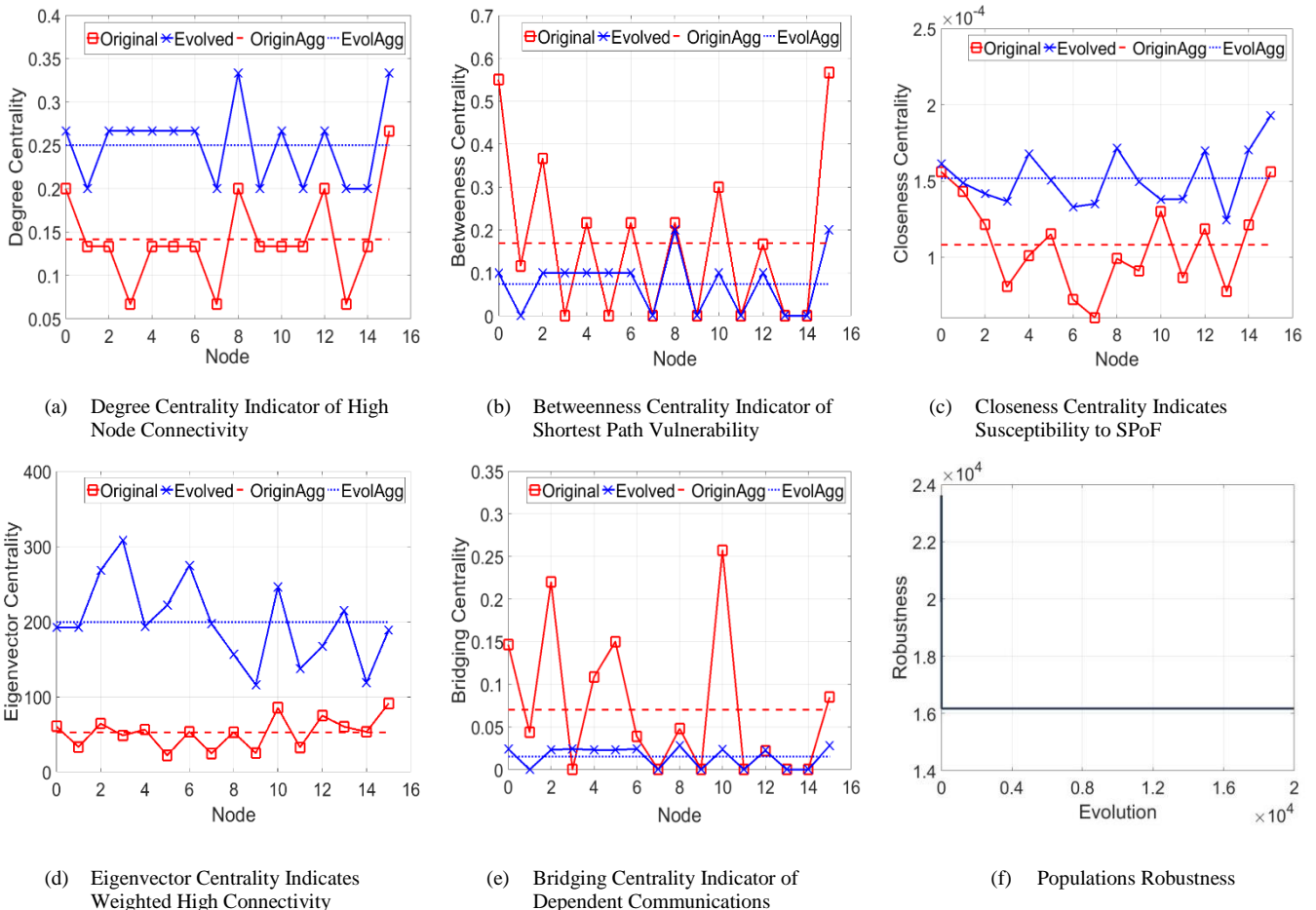


Figure 6.6. Multi-Level SoS Topological Security Vulnerabilities and Robustness Comparison

In this instance, when we applied the evolutionary risk mitigation method to evolve the entire multi-level SoS and enhance the security of the collaborative infrastructure, the optimum solution was generated within the first round of the process. In Figure 6.6 when we view the robustness graph

(Figure 6.6-f), we see that the robustness level for the multi-level SoS reduced by 31% from 23626.94388 to 16180.0003, meaning the reconfiguration is more appropriate.

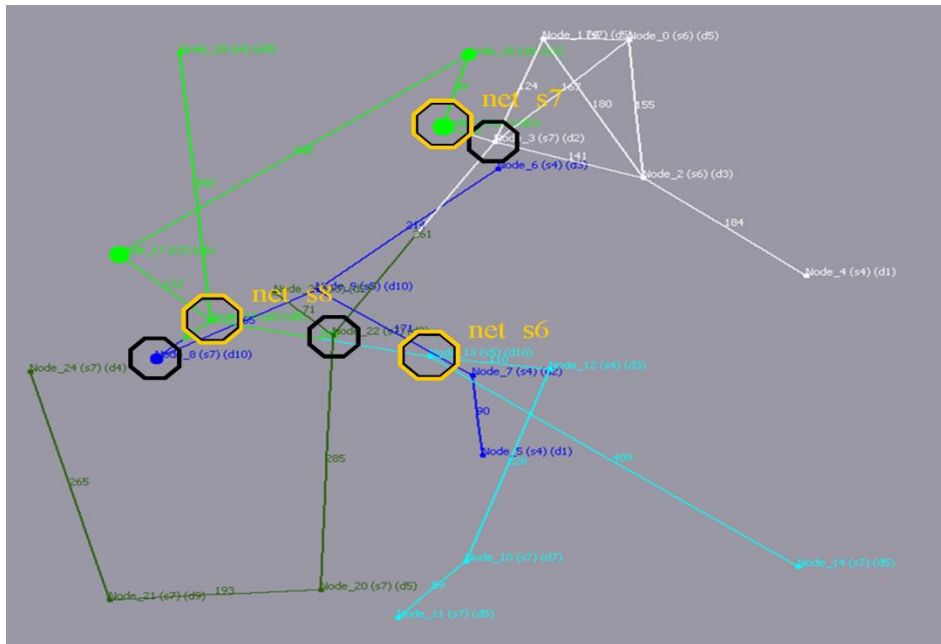
When we analyse the centralities of the SoS, we can view both the aggregated node centrality score for the multi-level SoS, or evaluate the impact that network reconfiguration has upon each of the distinct nodes' centrality factors. Analysing degree centrality (Figure 6.6-a), we note that the aggregated node centrality has increased by 76.47% from the original configuration of 0.142 to the reported optimum evolved candidate which scored 0.25, in contrast to the overall degree centrality of the graph which actually decreased.

SCRAM reports the degree centrality for each of the nodes for both the original collaborative infrastructure and the evolved multi-level SoS topology. These results allow us to compare the changes to centrality values that occurred to the evolution, and we can compare the increase or decrease to the centrality score against the aggregated node centrality score in order to establish if there is a significant difference between the two values. For example, node 3 in the original network configuration scores is 52% lower than the aggregated degree centrality score, and after evolution scores 6.67% more than the new aggregated degree centrality score, demonstrating the changes and impacts caused due to multi-level SoS security evolution, and we can intuitively see in the graph that no node has adversely been negatively impacted, with degree centrality scores remaining within tolerable ranges.

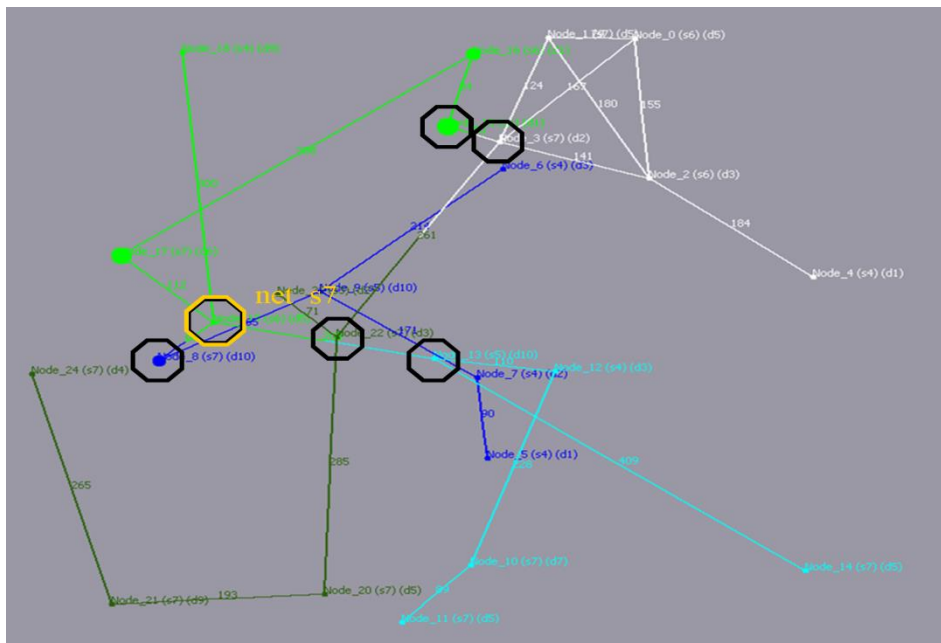
Reviewing the betweenness centrality (Figure 6.6-b), we see that the evolved SoS has reduced the aggregated betweenness centrality score by 32.08% from 0.25 to 0.16979. In addition, when we evaluate nodes 0 and 15 for example, it is noted that both of the node centrality scores have significantly reduced. The reconfigured optimum evolved candidate has reduced the betweenness centrality for node 0 by 81.82% and it is only 33.33% above the evolved aggregate node betweenness centrality score, compared to the original network configuration where node 0 was 223.93% above the aggregated node betweenness centrality. This graph conveys that the additional communication paths have reduced this centrality, meaning there has been a reduction of dominant nodes which were previously relied upon to maintain communications across the SoS and were high risk and potential SPoF.

The optimum candidate reports that closeness centrality has increased by 40.44% (Figure 6.6-c), with results indicating the nodes with the shortest paths, while eigenvector centrality (Figure 6.6-d) has increased by 281.1% with the results identifying the influential nodes within the network. Interestingly, bridging centrality (Figure 6.6-e) has decreased by 78.21%, and when we review the graph we can intuitively see that there are considerably fewer fluctuations between node scores and aggregated centrality score in the evolved candidate in comparison to the original network's configuration. For example, node 10 is quantified as being 267.3% higher than the aggregated node

identifies all external connecting nodes and their re-quantified security grades using the SCRAM simplistic and advanced connecting node security score methods.



(a) Multi-Level SoS with Applied Simplistic Connecting Node Security Scoring

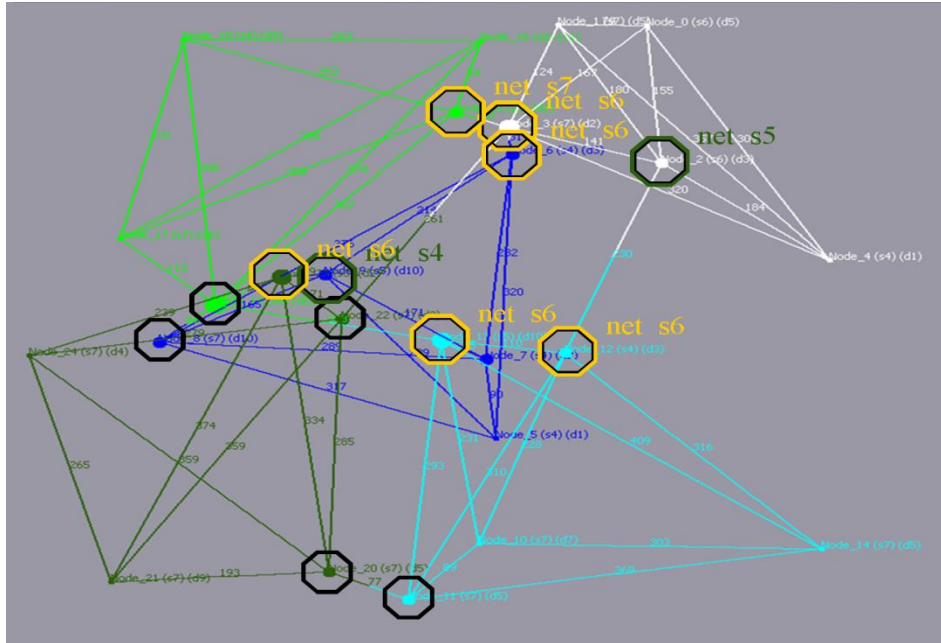


(b) Multi-Level SoS with Applied Advanced Connecting Node Security Scoring

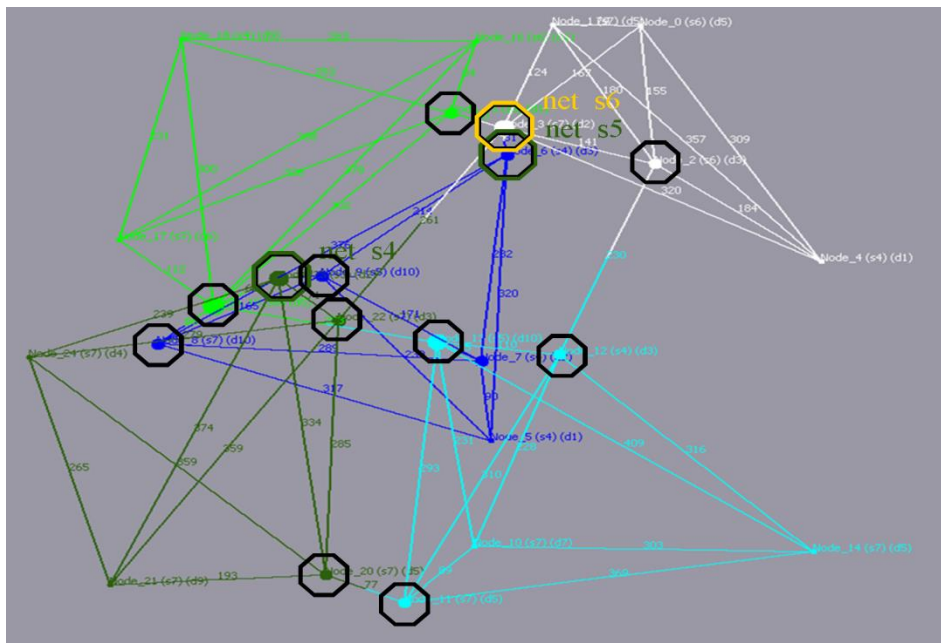
Figure 6.8. Simulated Multi-Level SoS Primary Connecting Node Scores

When each scoring method is applied, firstly using the simplistic method (Figure 6.8-a) we see that 3 of the connecting nodes are quantified as having their grades altered due to the external connections and the risks that they introduce, with node 13 having been originally quantified as secure, but being quantified as insecure due to its external data link to another SoS, while nodes 15 and 19 remain categorised as insecure but have had their security grades increased, demonstrating their increased

risk to the entire multi-level SoS. When we compare the advanced scoring method for connecting nodes (Figure 6.8-b) only a single node has been quantified as insecure. Moreover, node 19 is identified as increasing its grade by 1 when the advanced method is applied, compared to advancing by two grades using the simplistic scoring technique.



(a) Multi-Level SoS with Applied Simplistic Connecting Node Security Scoring



(b) Multi-Level SoS with Applied Advanced Connecting Node Security Scoring

Figure 6.9. Security Enhanced Simulated Multi-Level SoS Connecting Node Scores

Similarly, when we apply the evolutionary principles to evolve the collaborative infrastructure and reconfigure the communication links in order to mitigate risks, enhance the security of network communications, improve network robustness, and limit SPoF, the simplistic scoring method (Figure

6.9-a) re-categorises 8 of the 13 established connecting nodes, while the advanced method (Figure 6.9-b) re-categorises only 3. This corroborates that the advanced scoring technique presented in Section 6.1.1 is not only more accurate and reliable, but a necessary technique to be used as part of the risk assessment of the SoS. Should grades be over or under quantified, then risks could remain unidentified exposing infrastructures, or resources could be wasted to improve the security of a node that is not required.

Throughout our experiments, we have placed emphasis on the SCRAM robustness function which combines five key parameters, to assist the evolutionary security risk mitigation algorithm to produce new improved solutions, and provides a numerical number that can be used as a quick indicator to assess the overall appropriateness of the evolved SoS candidates. Analysing the robustness of the multi-level SoS (Figure 6.10-f), we see a notable reduction in the robustness score as security evolution is conducted, with the final optimum candidate reporting a reduction of 40.66%.

The improved robustness is reflected in the 125% increase in the multi-level SoS communication security, which increased from only 24% to 54%. As previously stated, we are forcing external connections between independent and distinct SoS environments, and SCRAM evolves the entire multi-level SoS reconfiguring communication paths, in order to ensure that there is enough built-in redundancy while assuring there are secure data routes between the collaborative infrastructures. Therefore, we determine that the accuracy of the security analysis method is a true reflection of the overall security, as it does not report an overly high security level (i.e. does not achieve 100% secure). As this level of security would not be attainable for this infrastructure, for instance we know this particular multi-level SoS is constructed of 60% insecure nodes, with only 5 nodes throughout the SoS being quantified as secure and not in breach of data access policies.

Table 6.4. Simulated Multi-Level SoS Evolution Results

Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.8	5	5	30%	0.1286232	24%	4784	924.86	60% / 28%
6.9	5	5	-	0.9963767	54%	13552	363.79666	60% / 28%

The reported optimum evolved candidate demonstrates that the SCRAM framework generated and selected a security configuration that reduces the minimum path average between nodes, and reduces both betweenness and bridging centrality. Minimum path average (Table 6.4) is reduced by 60.66% corroborating that the network configuration has increased in efficiency as the average number of steps along the shortest paths has been notably reduced. Aggregated betweenness centrality (Figure 6.10-b) decreased by 68.03% reducing the number of dominant nodes relied upon to maintain communications, while aggregated bridging centrality (Figure 6.10-e) decreased by 87.75% reducing the reliance upon single nodes for data transfer.

Similar to the previous experiment the aggregated centrality scores for degree, closeness, and eigenvector all marginally increased. Aggregated degree centrality (Figure 6.10-a) increased by 122.22%, closeness (Figure 6.10-c) increased by 94.66%, and eigenvector (Figure 6.10-d) increased by 424.3%, identifying the increase of neighbouring nodes, the nodes with the shortest paths, and the influence nodes have within the new configured multi-level SoS topology.

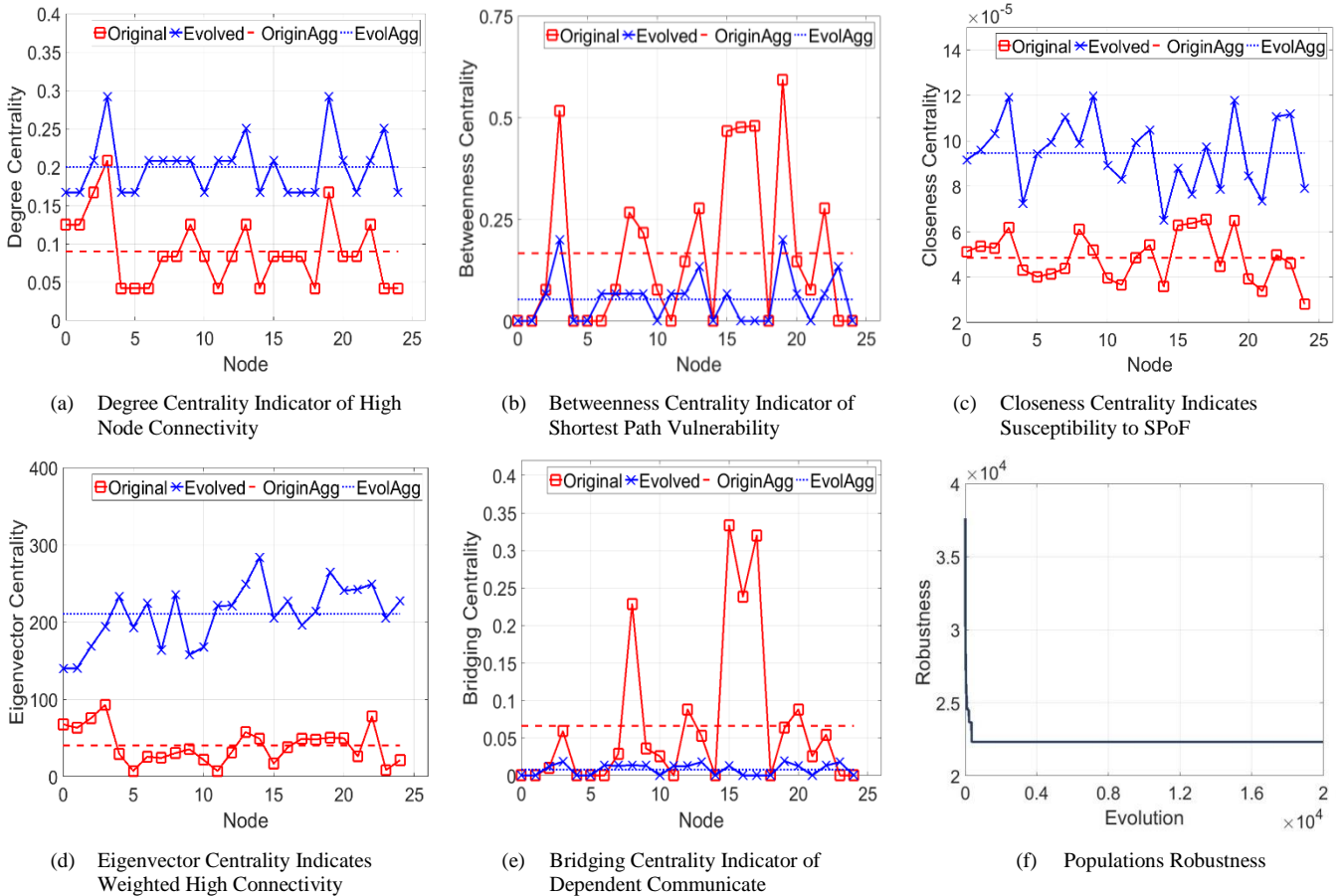


Figure 6.10. Multi-Level SoS Topological Security Vulnerabilities and Robustness Comparison

6.2 Multi-Level SoS Evaluation

To overcome the limitations of existing techniques, the theoretical proposed solutions were implemented as part of the SCRAM framework. In the previous sections, we have validated the usefulness of applying these principles against SoS. To validate the applied principles that attempt to increase SoS security, identify and reduce vulnerabilities, and limit potential SPoF, in this chapter we conduct a series of experiments to corroborate the SCRAM framework and implemented techniques against a series of differing complex multi-level SoS. The results will assist us to evaluate the SCRAM framework further, and will objectively assist to determine if our research aims and objects have been achieved.

6.2.1 Multi-level SoS: SCRAM Evaluation

A fundamental question is whether we can reconfigure multiple distinct SoS, evolving their networked communication paths (both internal and external) in unison to ensure the entire multi-level SoS security is assured and robust. As we force new connections between distinct SoS we have proven the impact that can occur to the connecting nodes and wider repercussions to security, network centralities, cost in terms of distance between nodes, etc. Analysing the multi-level SoS together means there is no need to enhance security and reconfigure and search for the optimum solution for each distinct SoS prior to their integration into the multi-level SoS, meaning that we do not have to run security risk mitigations methods more than once.

Analysing the entire multi-level SoS as a single entity using the principles outlined in this thesis, means we can monitor connections as they are reconfigured and removed. For example, if vital connections are removed or there are too few connections between collaborative infrastructures, then the overall robustness of the network will decrease. However, to increase robustness we can't simply just add new connections to strengthen the SoS, we have to consider that the introduction of new data paths will introduce new vulnerabilities and attack points that previously did not exist, connections between external connecting nodes will impact the security grades of each connecting node, and we have to consider the status of nodes that connections are formed between as we have to prioritise secure nodes and those that do not breach data access requirements.

While priority is given to meet these demands, in some instances providing there is a single secure route between nodes, alternative node connections are considered and presented as optimum solutions. This is due to the applied techniques within SCRAM analysing the network cost, minimum path average, and the impact on centralities. In these rare instances however, the visualised undirected graphs highlight the connections between insecure nodes and provide a detailed report, which in turn would allow decision makers to consider applying methods to secure the nodes prior to the suggested reported optimum solution being implemented.

To evaluate the applied principles we generated a series of random multi-level SoS within the SCRAM framework with varying SoS sizes, communication connections, varying vulnerabilities, security grades, and data access levels. In the following sections we analyse the results of these experiments conducted on varying multi-level SoS topologies, which we have grouped into four different collections.

6.2.1.1 SCRAM Positive Multi-Level SoS Vulnerability

Performance

The first collection we present focuses on multi-level SoS topologies that only consider vulnerabilities within the topology and do not apply the data access policy. These types of multi-level SoS characterise WSN and IoT topologies, where data exchange between components and devices is essential and a fundamental part of their purpose.

Figures 6.11, 6.12, and 6.13 visualise each of the nine conducted experiments for this collection. In these instances we have omitted connecting node indicators and the individual identifiable network colours, as seen in Figure 6.9, as while these pointers can be easily seen on a computer screen they fail to translate into small printed diagrams and produce too much indeterminate visual disturbance, especially as we increase the scale of the multi-level SoS being tested.

These graphs visualise the diverse topology structures being assessed and their enhanced security reconfiguration, in the first graph (example Figure 6.11-a) for each multi-level SoS we present the collaborative infrastructure with all identified vulnerable nodes, in the second graph example (example Figure 6.11-b) we have stripped away node status to allow us to visualise clearly the communication links between nodes and nodes with significant bridging centralities, the final graph (example Figure 6.11-c) visualises the security enhanced solution's new structure and clearly shows the increase and decrease to the number of communication paths and alterations to node bridging centralities.

Table 6.5 presents the original multi-level SoS properties for each experiment and Table 6.6 reports the evolved security assessment results for the optimum reported candidate. Figures 6.14, 6.15, and 6.16 present a comparison of the aggregated centrality and robustness scores for each infrastructure.

Table 6.5. Multi-level SoS Unevolved Vulnerability Performance Properties Comparison

Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.11-a	8	8	30%	0.061955966	18%	13445	718.02875	47% / 0%
6.11-d	8	8	40%	0.08038917	28%	25969	608.01886	48% / 0%
6.11-g	8	8	50%	0.060931906	31%	36697	514.435	66% / 0%
6.12-a	10	10	30%	0.050711192	18%	25308	772.0301	53% / 0%
6.12-d	10	10	40%	0.048855904	23%	51621	642.13654	54% / 0%
6.12-g	10	10	50%	0.039167188	25%	77221	526.33594	58% / 0%
6.13-a	12	12	30%	0.035260525	14%	44753	741.0788	61% / 0%
6.13-d	12	12	40%	0.03171478	35%	84209	542.64014	60% / 0%
6.13-g	12	12	50%	0.039167188	25%	77221	526.33594	58% / 0%

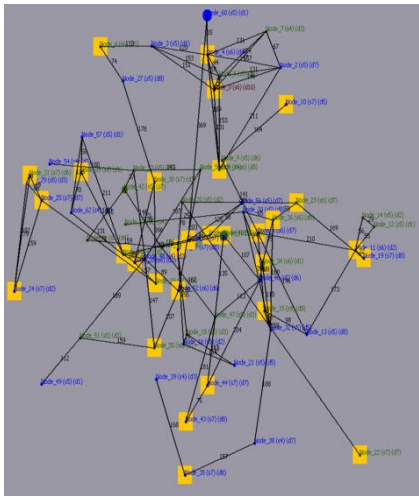
Table 6.6. Multi-level SoS Evolved Vulnerability Performance Comparison

Figure	Number of Nodes in Network	Number of Networks	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average
--------	----------------------------	--------------------	----------------------------	----------------------------------	------	----------------------

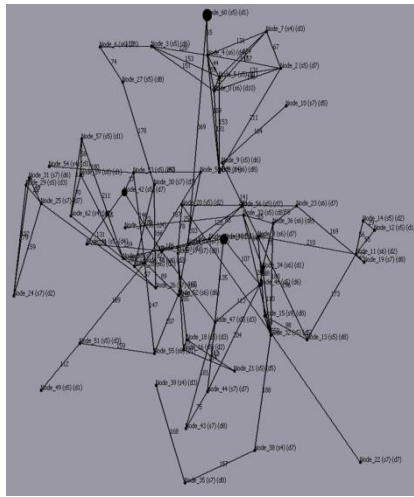
6.11-c	8	8	0.059907857	27%	23828	566.9797
6.11-f	8	8	0.04147466	28%	25178	559.05206
6.11-i	8	8	0.045570917	37%	23296	556.87994
6.12-c	10	10	0.023088017	33%	39339	608.5289
6.12-f	10	10	0.03133374	23%	42377	632.9644
6.12-i	10	10	0.0206143	26%	43412	584.5719
6.13-c	12	12	0.026593126	24%	61189	629.63983
6.13-f	12	12	0.03013882	49%	67573	595.8983
6.13-i	12	12	0.0206143	26%	43412	584.5719

From these figures we can see that in every instance the security and robustness level of the multi-level SoS has improved to varying degrees or been maintained, when the evolutionary security risk mitigation methods are applied to the entire collaborative infrastructure. When we analyse Multi-SoS E (Figure 6.12-d) in Table 6.6, while security has been maintained at 23%, secure communication alterations to the infrastructure has resulted in other positive outcomes. Firstly, there is a reduction to network cost reducing communication by 17.91% from 51621 to 42377, a minor 1.43% decrease to minimum path average, and a 35.86% decrease to the degree centrality of the graph.

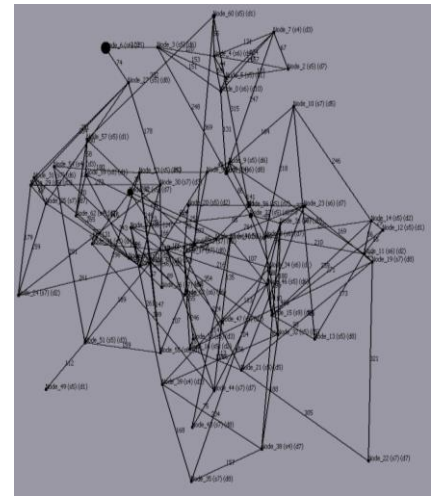
Analysing the aggregated centralities (Figure 6.15) we also continue to see improvements to the multi-level SoS in Set B, with aggregated node degree, eigenvector and bridging centrality all indicating reductions to their scores. This demonstrates that the removal of the redundant communication paths has not unduly impacted the security of the distinct SoS forming the multi-level SoS, and the multi-SoS remains robust and secure, having mitigated associated risk. With the SCRAM framework quantifying and visualising vulnerable nodes and risks associated with centralities within the multi-level SoS topology, as these issues can expose the entire collective infrastructure to various risks, zero-day attacks, and if left unidentified, vulnerabilities could be exploited or cause cascading failure within the networked systems. Detecting these issues allows for actions to be taken prior to them being exploited or failing, assuring that the multi-level SoS can be further strengthened by early risk identification. In addition, the improvements and maintenance of low centrality scores means that SCRAM has maintained an adequate number of neighbouring nodes, maintained short paths, and has not overtly increased bridging centrality, meaning nodes are not excessively relied upon for the transfer of data or maintaining the structure of the communication network.



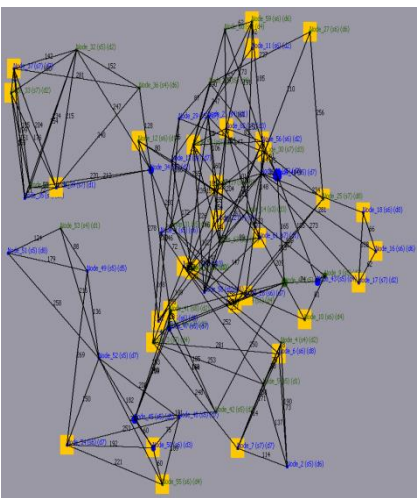
(a) Multi-Level SoS A with Node Status



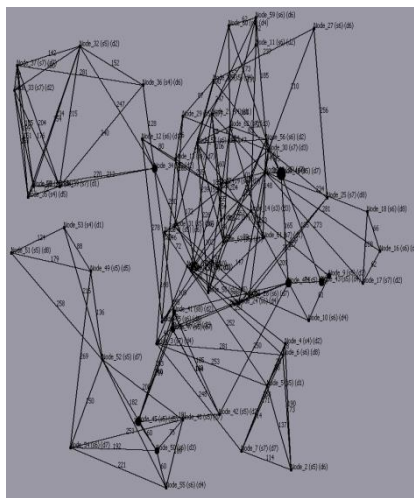
(b) Multi-Level SoS A Topology



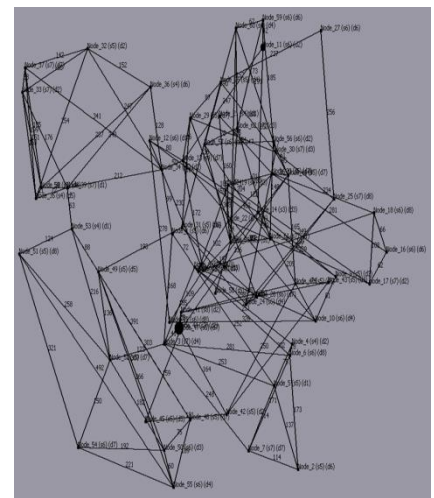
(c) Multi-Level SoS A Optimum Candidate



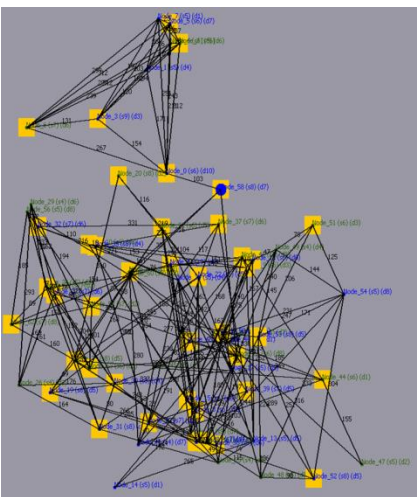
(d) Multi-Level SoS B with Node Status



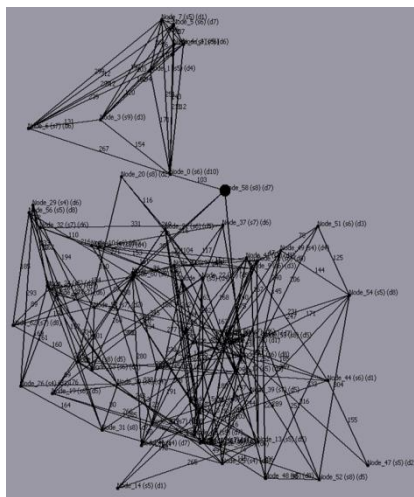
(e) Multi-Level SoS B Topology



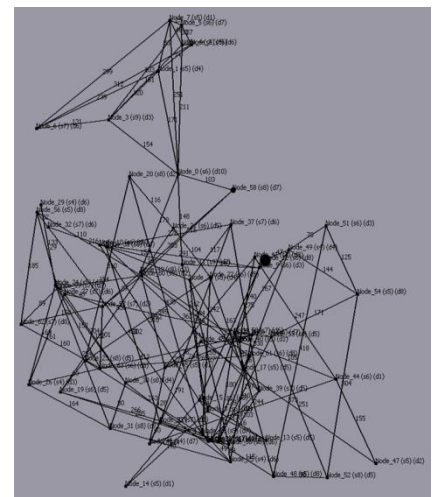
(f) Multi-Level SoS B Optimum Candidate



(g) Multi-Level SoS C with Node Status

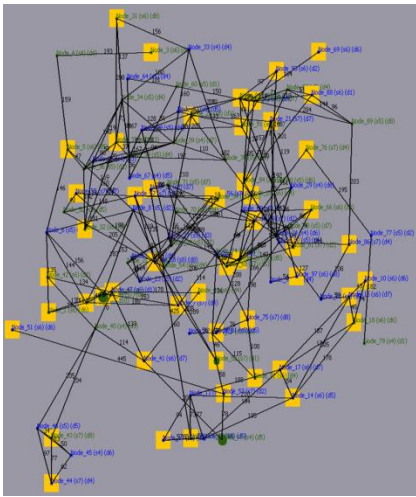


(h) Multi-Level SoS C Topology

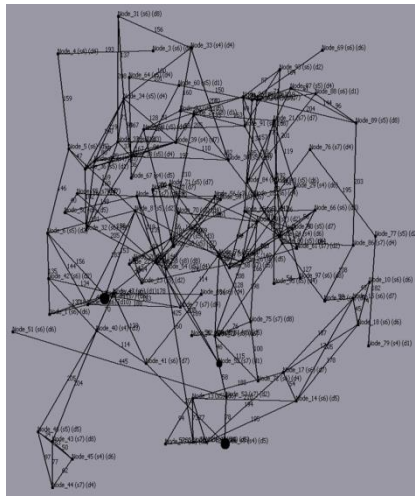


(i) Multi-Level SoS C Optimum Candidate

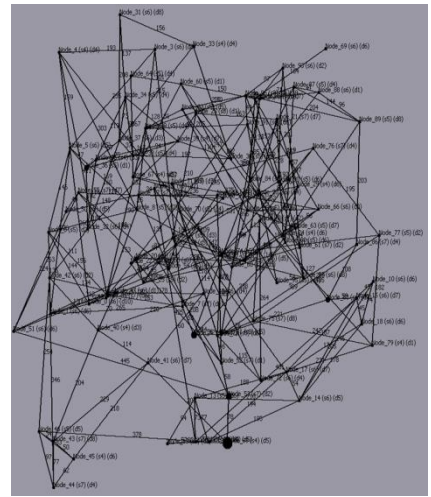
Figure 6.11. Set A of Multi-Level SoS Used in the Experiments (see Appendix A)



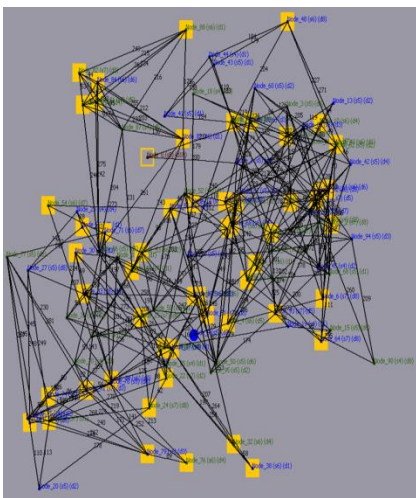
(a) Multi-Level SoS D with Node Status



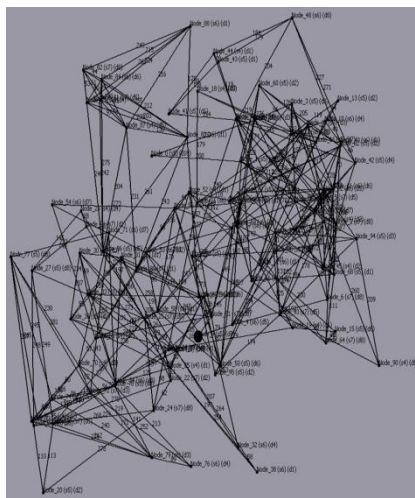
(b) Multi-Level SoS D Topology



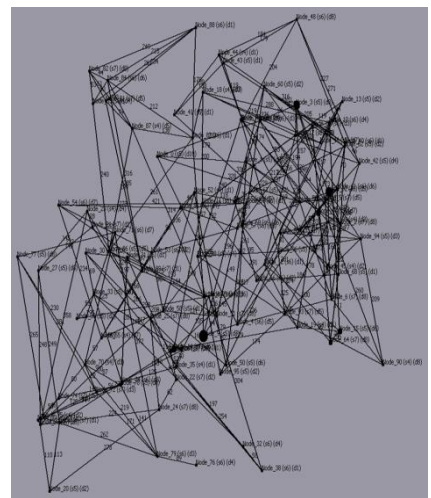
(c) Multi-Level SoS D Optimum Candidate



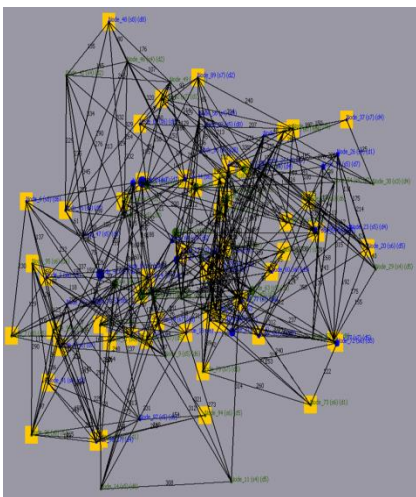
(d) Multi-Level SoS E with Node Status



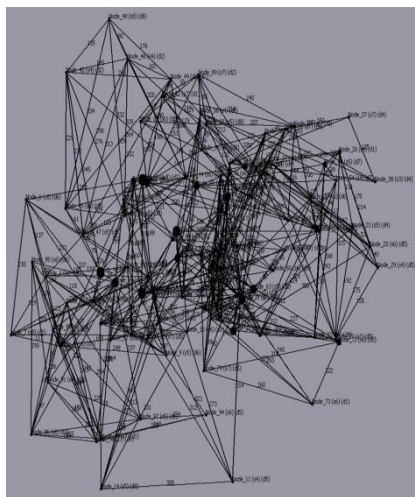
(e) Multi-Level SoS E Topology



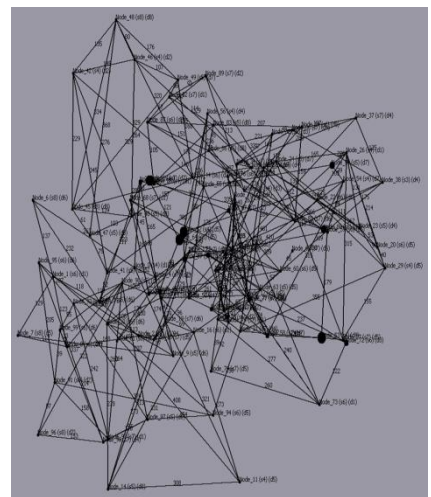
(f) Multi-Level SoS E Optimum Candidate



(g) Multi-Level SoS F with Node Status

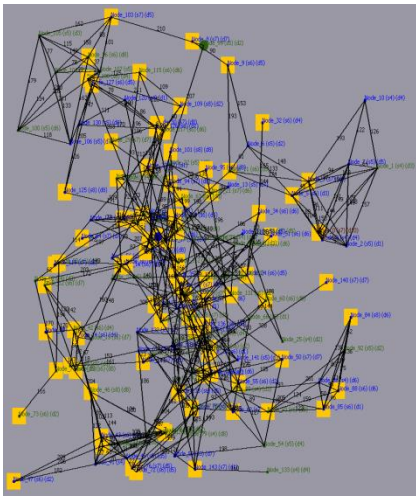


(h) Multi-Level SoS F Topology

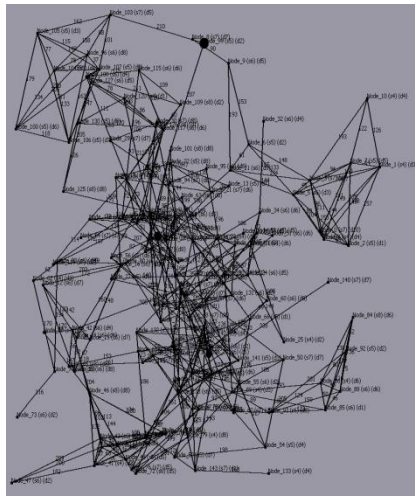


(i) Multi-Level SoS F Optimum Candidate

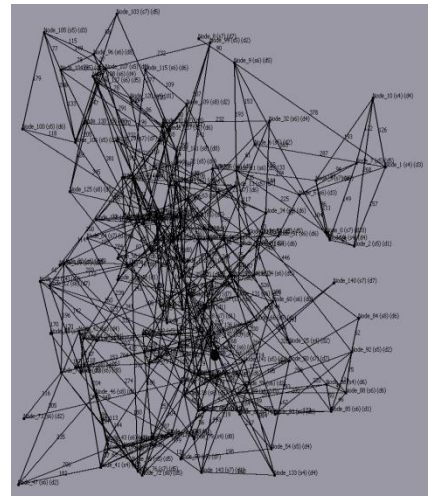
Figure 6.12. Set B of Multi-Level SoS Used in the Experiments (see Appendix A)



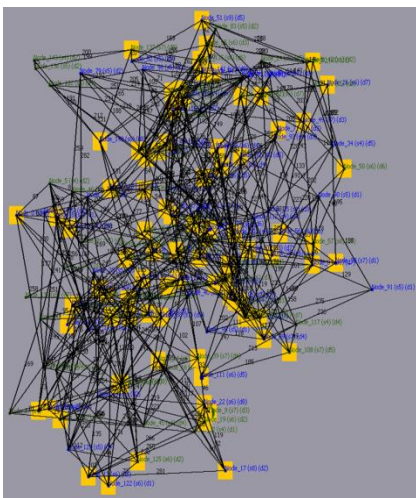
(a) Multi-Level SoS G with Node Status



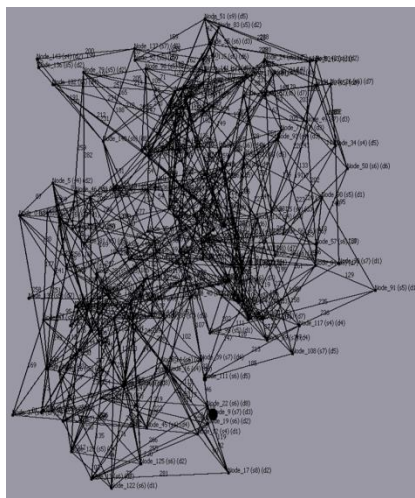
(b) Multi-Level SoS G Topology



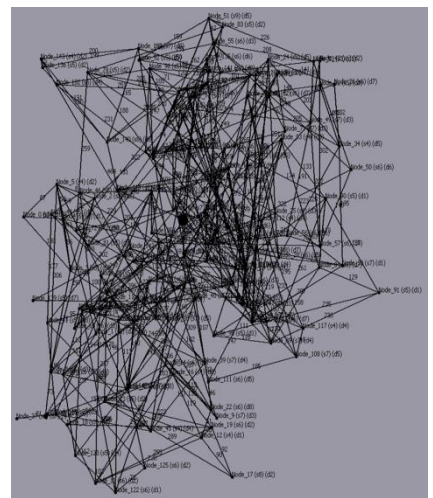
(c) Multi-Level SoS G Optimum Candidate



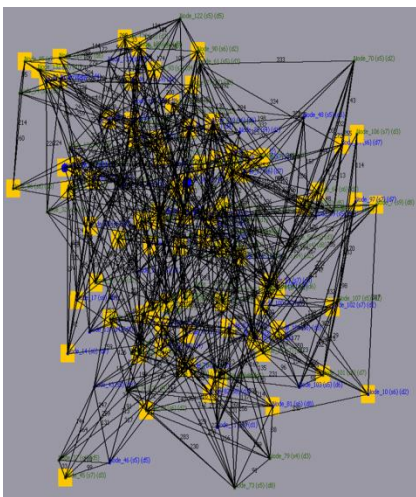
(d) Multi-Level SoS H with Node Status



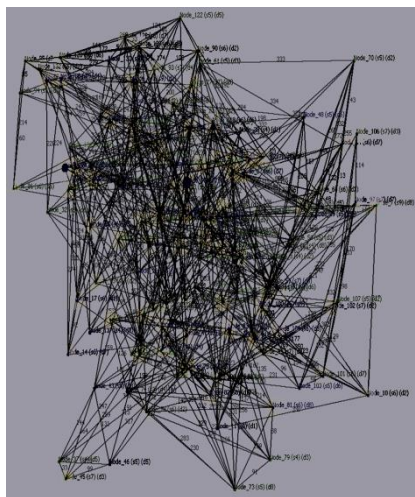
(e) Multi-Level SoS H Topology



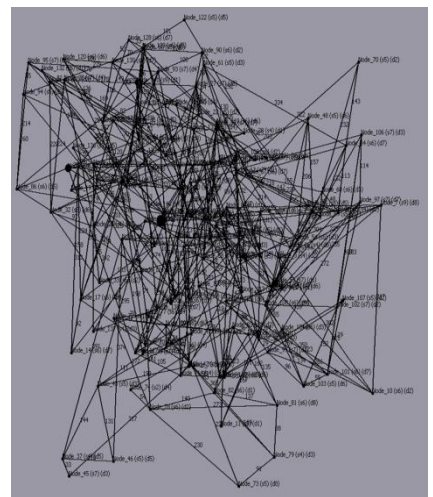
(f) Multi-Level SoS H Optimum Candidate



(g) Multi-Level SoS I with Node Status



(h) Multi-Level SoS I Topology



(i) Multi-Level SoS I Optimum Candidate

Figure 6.13. Set C of Multi-Level SoS Used in the Experiments (see Appendix A)

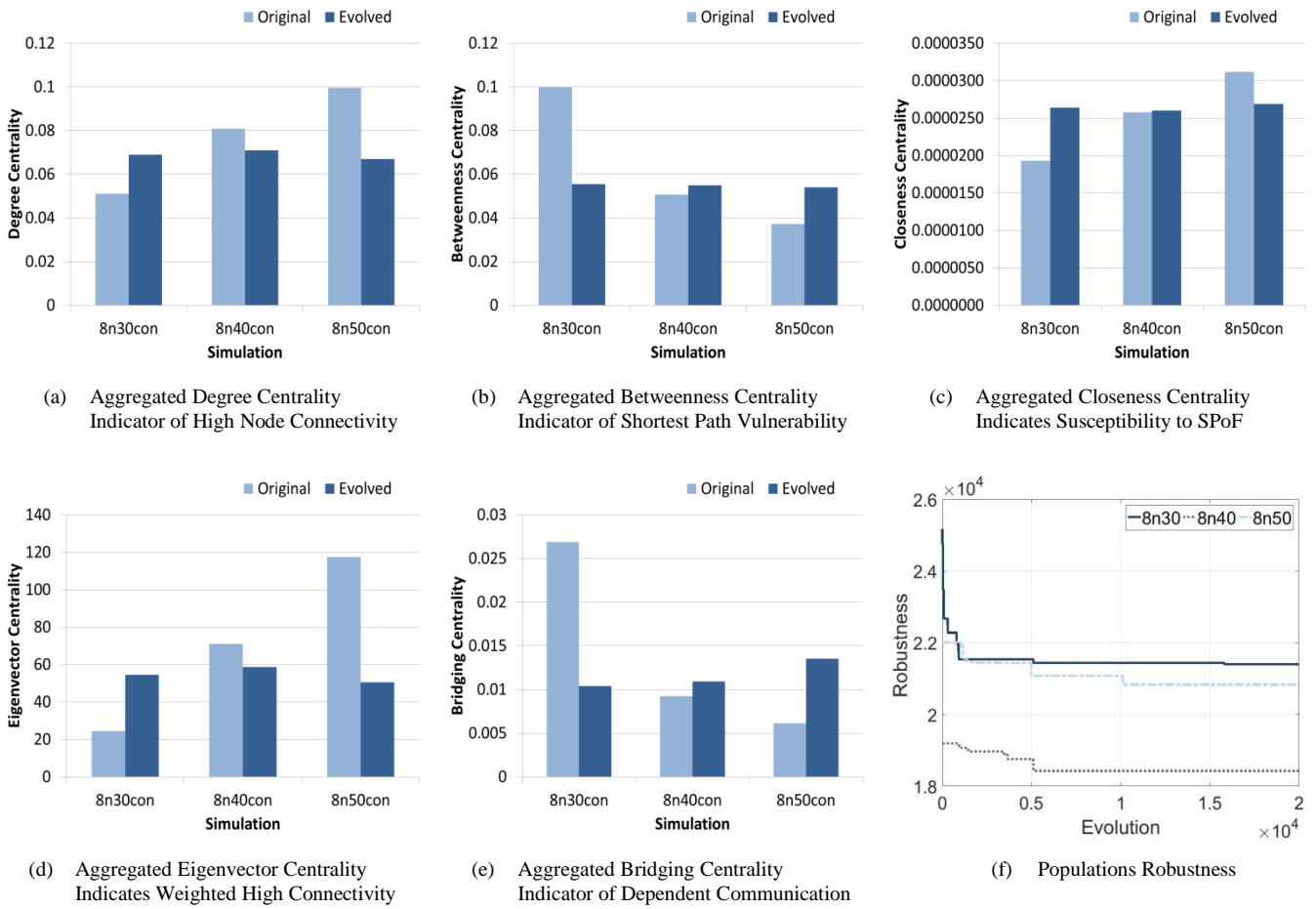


Figure 6.14. Set A Multi-Level SoS Security Vulnerabilities and Robustness Comparison

When we critically review multi-level SoS H (Figure 6.13-d) in Table 6.6, in this instance we see a 40% increase in the overall communication security for the entire collaborative infrastructure. While the degree centrality for the graph has marginally decreased by 4.97%, and the cost of communications has reduced by 19.79%. This reduction to network cost and significant reduction to the number of connecting data paths is evidenced by the minor increase of 9.81% to the minimum path average, which can be considered acceptable considering the positive increase to network robustness and security, and the reduction to communication costs, especially should decision makers be forced to consider security enhancement when having to comply with financial constraints and savings.

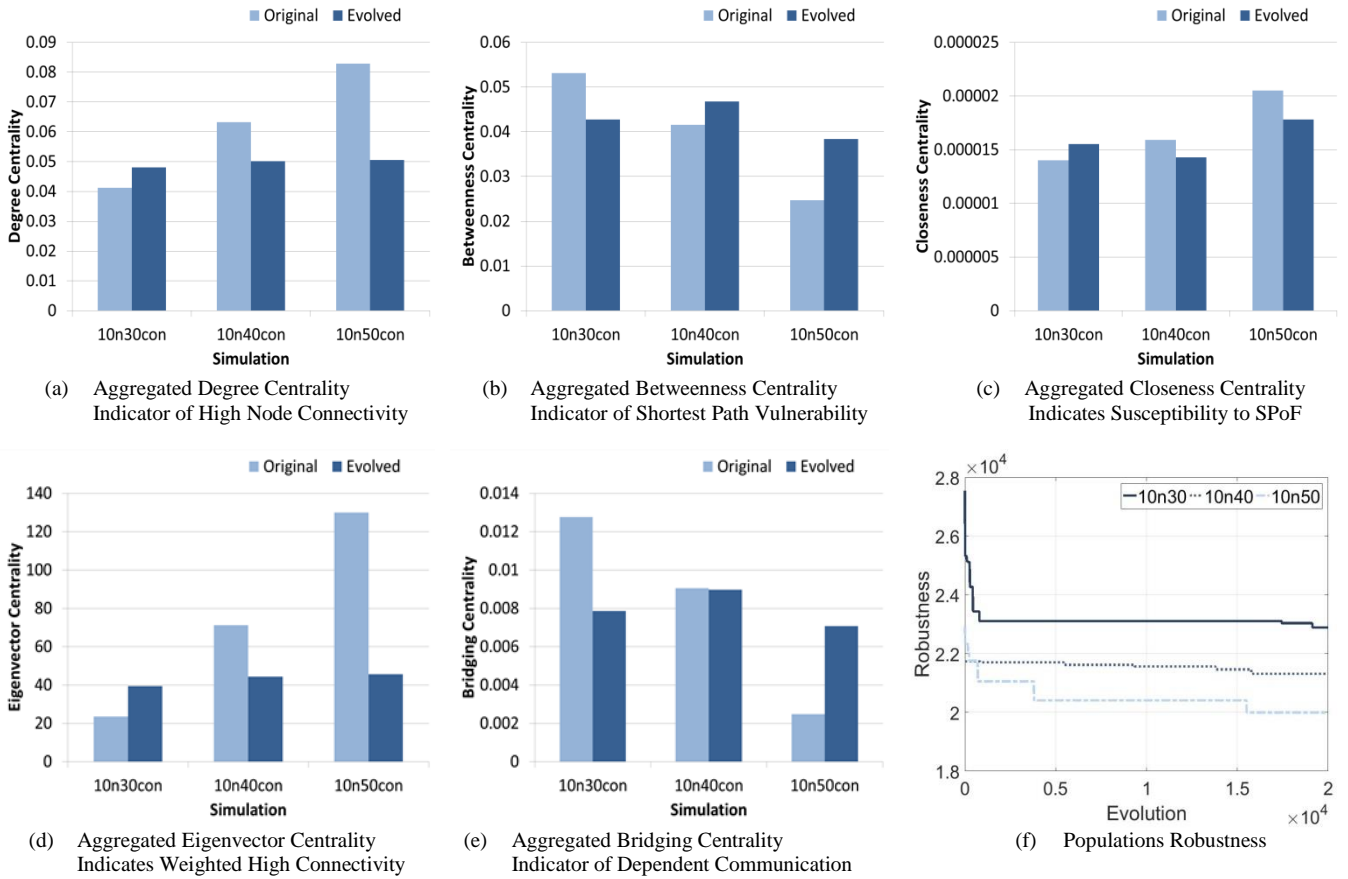


Figure 6.15. Set B Multi-SoS Topological Security Vulnerabilities and Robustness Comparison

Analysing the aggregated centralities (Figure 6.16) positive improvements continue to be seen, with aggregated node degree, and eigenvector centrality all indicating positive scores, with minor reductions to closeness centrality. Similar to multi-level SoS E, this supports the evolutionary process being used in this type of topology to mitigate risk and secure multi-level SoS in an attempt to assure security and data flow, and that the removal of communication paths and the addition of new paths have not unduly impacted the overall appropriateness and security of the collaborative infrastructure. We are presented with an increase to the betweenness centrality which increases marginally from 0.021 to 0.028, and bridging centrality which increases from 0.0027 to 0.0047. However, these parameters fall within the threshold limits and while parameter scores have increased, the SCRAM evolutionary security risk mitigation process has ensured that security evolution does not excessively increase the burden on single nodes or cause nodes to have more influence within the network. Ensuring that these potentially influential nodes are monitored and identified, as their removal or failure would have significant impact within the entire multi-level SoS infrastructure.

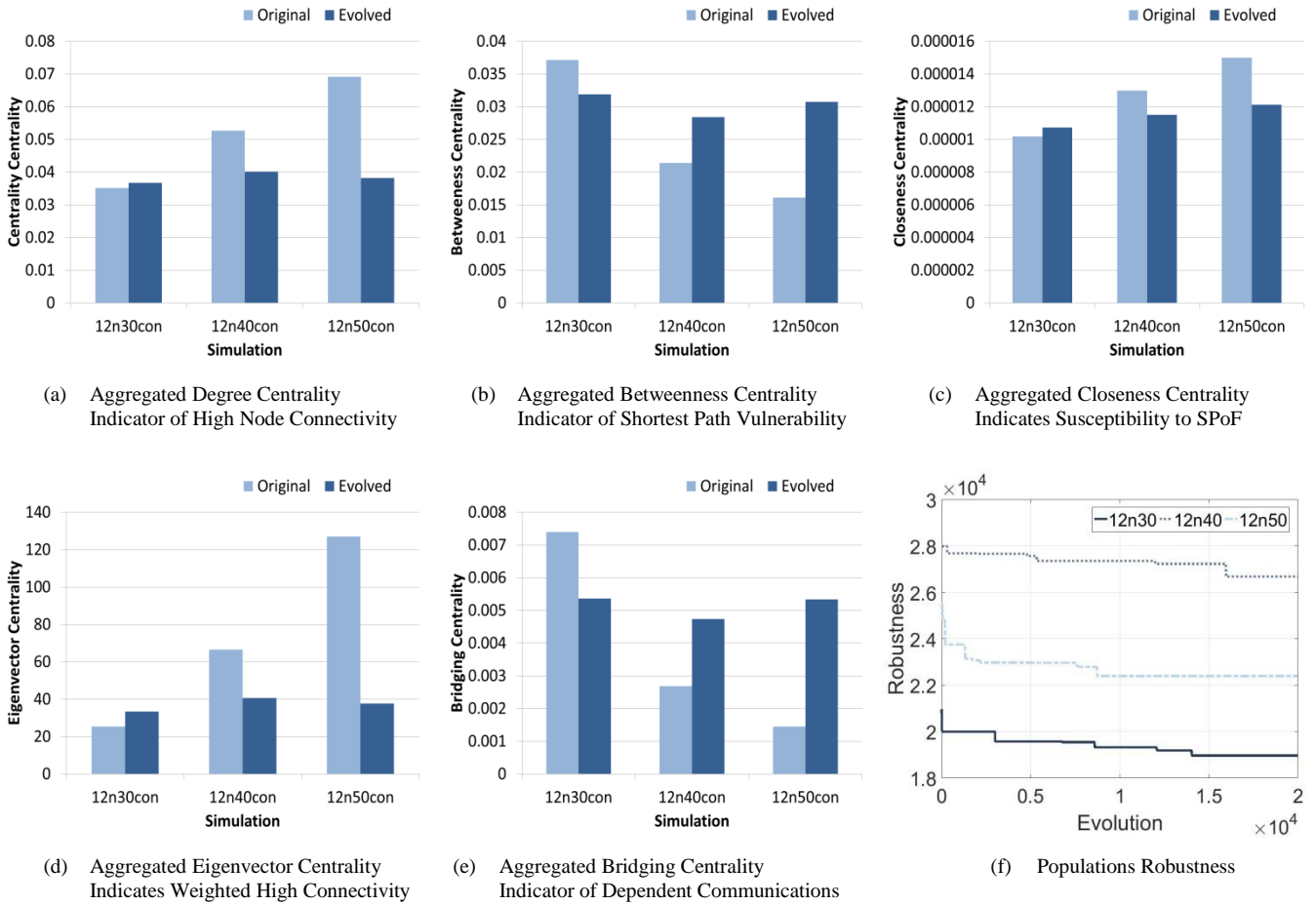


Figure 6.16. Set C Multi-SoS Topological Security Vulnerabilities and Robustness Comparison

6.2.1.2 SCRAM Positive Multi-Level SoS Vulnerability and Data Access Performance

This second collection of multi-level SoS topologies presented in this section considers both the vulnerabilities and data access grades, i.e. both parameters are incorporated into the evolutionary security risk mitigation algorithm. These types of SoS characterise not only WSN and IoT but other ICT infrastructures including critical infrastructures and Smart Cities for example, where it is vital that data exchange only occurs between components and devices that uphold the data access policy requirements, in order to assure data security and prevent unauthorised access and exposure to sensitive data as it traverses across the insecure and unencrypted collaborative infrastructure.

Figures 6.17, 6.18, and 6.19 visualise each of the nine conducted experiments within this section. These undirected graphs have had the connecting node indicators and individual identifiable network colours omitted, to ensure communication paths and nodes are identifiable in this thesis. Each

network is represented via three graphs, the first graph visualising the quantified insecure nodes, and the nodes that will be blocked as they are in breach of data access policies.

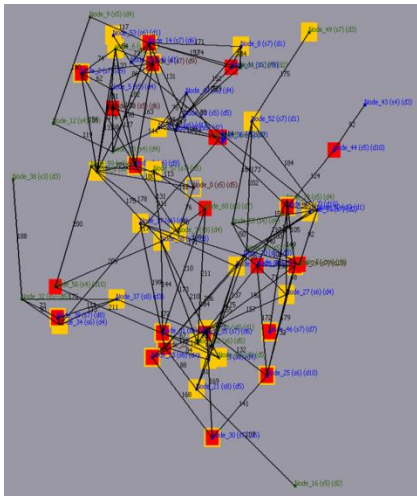
The second graph is stripped bare so we can intuitively examine the communication paths and node bridging centrality, with the third graph representing the security enhanced optimum candidate visualising the increased/decreased communication paths and alterations to the topology of the complete multi-level SoS. Table 6.7 presents the original multi-level SoS properties and Table 6.8 provides the evolved security assessment results for the optimum candidate. Figures 6.20, 6.21, and 6.22 provide a comparison for the aggregated centrality and robustness scores for each collaborative infrastructure.

Table 6.7. Multi-Level SoS Unevolved Vulnerability and Data Access Performance Comparison

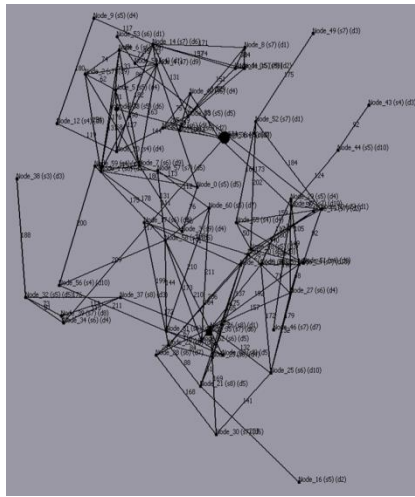
Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.17-a	8	8	30%	0.06554021	16%	16433	677.57043	55% / 36%
6.17-d	8	8	40%	0.120839745	19%	21774	523.48267	48% / 44%
6.17-g	8	8	50%	0.0701485	28%	38297	456.69592	50% / 47%
6.18-a	10	10	30%	0.05483405	17%	30166	639.54224	68% / 51%
6.18-d	10	10	40%	0.052566476	30%	47479	605.318	45% / 47%
6.18-g	10	10	50%	0.02411873	25%	79029	548.8606	51% / 48%
6.19-a	12	12	30%	0.03791983	15%	50657	578.7408	55% / 49%
6.19-d	12	12	40%	0.032601204	18%	85986	673.7375	56% / 46%
6.19-g	12	12	50%	0.038904767	19%	131533	506.0106	56% / 53%

Table 6.8. Multi-Level SoS Evolved Vulnerability and Data Access Performance Comparison

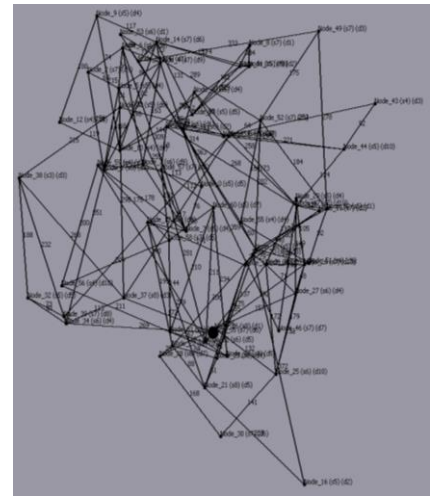
Figure	Number of Nodes in Network	Number of Networks	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average
6.17-c	8	8	0.057347693	23%	21892	515.13245
6.17-f	8	8	0.06093192	24%	23719	545.2525
6.17-i	8	8	0.0870456	30%	24023	517.1161
6.18-c	10	10	0.03236446	18%	38119	599.359
6.18-f	10	10	0.032982886	30%	41343	639.1073
6.18-i	10	10	0.030096881	25%	39649	594.9689
6.19-c	12	12	0.039791193	16%	55356	569.92065
6.19-f	12	12	0.039003253	19%	66202	666.18134
6.19-i	12	12	0.025509696	20%	70383	637.1491



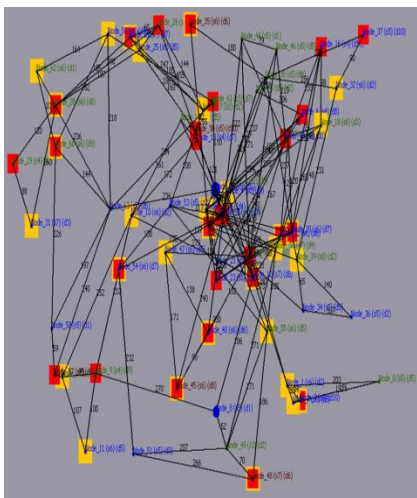
(a) Multi-Level SoS J with Node Status



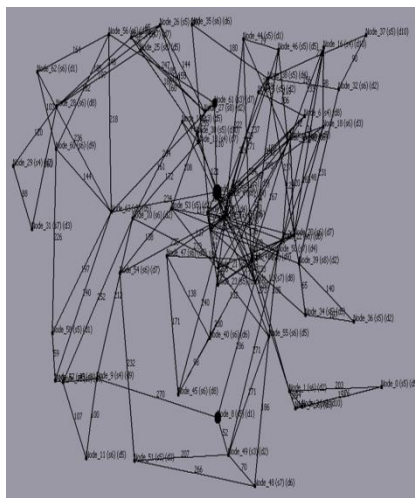
(b) Multi-Level SoS J Topology



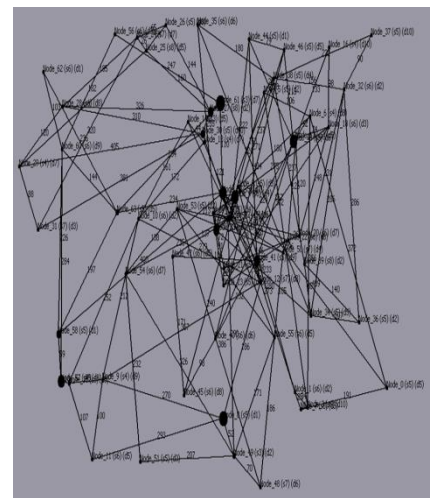
(c) Multi-Level SoS J Optimum Candidate



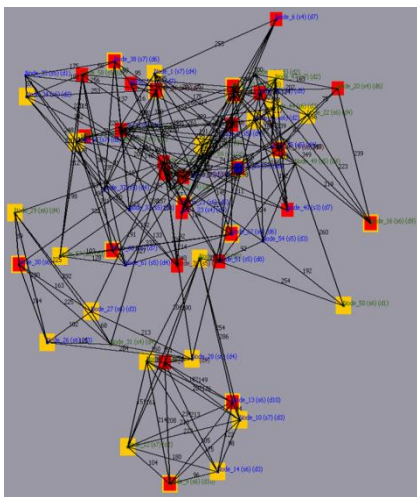
(d) Multi-Level SoS K with Node Status



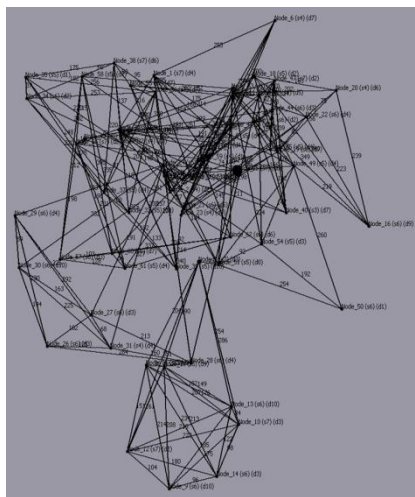
(e) Multi-Level SoS K Topology



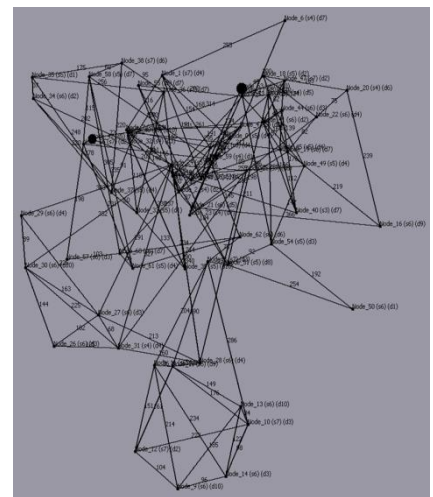
(f) Multi-Level SoS K Optimum Candidate



(g) Multi-Level SoS L with Node Status

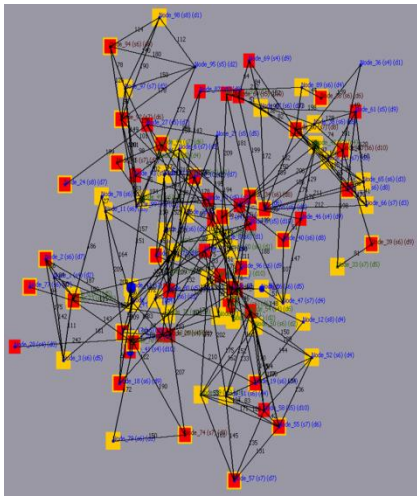


(h) Multi-Level SoS L Topology

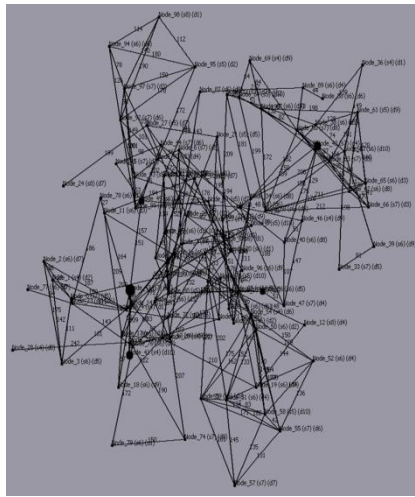


(i) Multi-Level SoS L Optimum Candidate

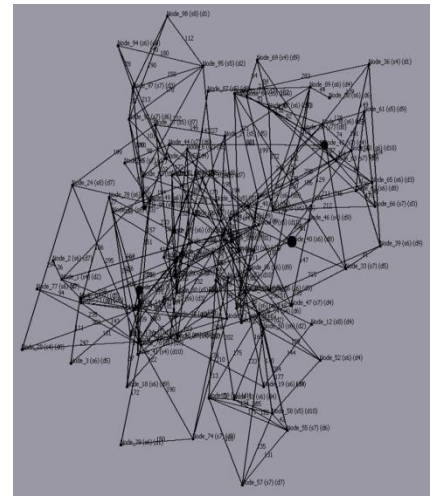
Figure 6.17. Set D of Multi-Level SoS Used in the Experiments (see Appendix B)



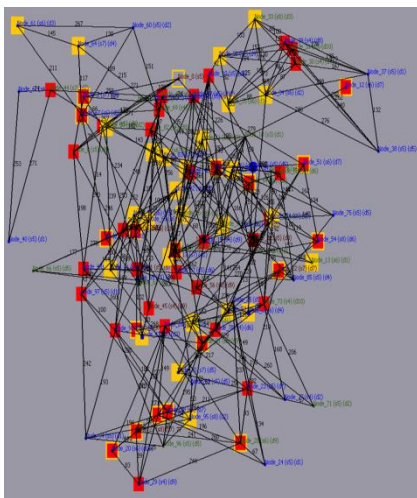
(a) Multi-Level SoS M with Node Status



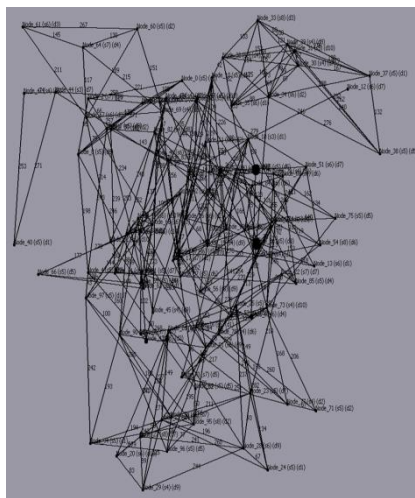
(b) Multi-Level SoS M Topology



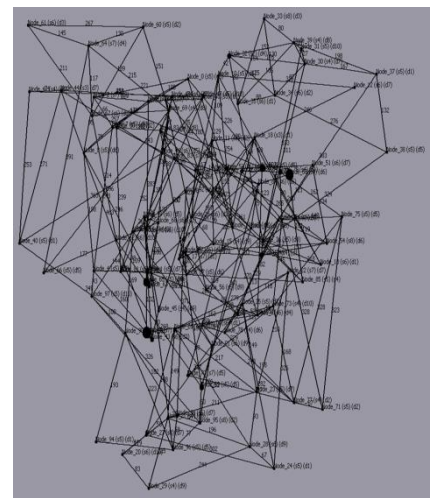
(c) Multi-Level SoS M Optimum Candidate



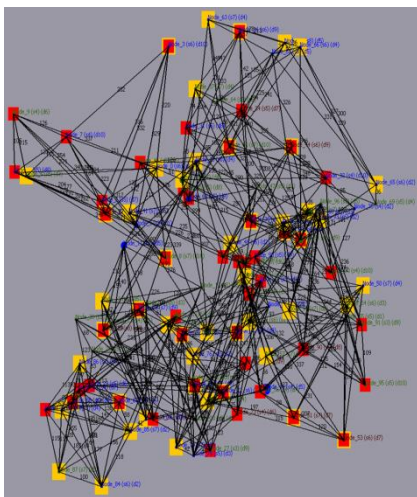
(d) Multi-Level SoS N with Node Status



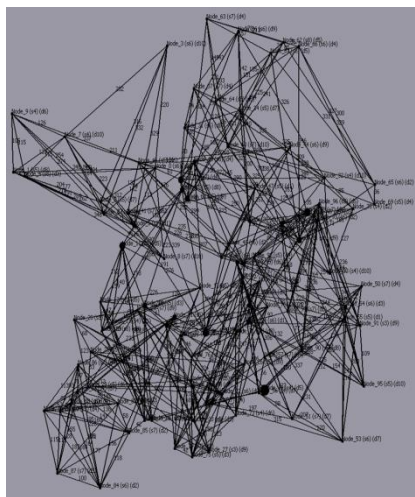
(e) Multi-Level SoS N Topology



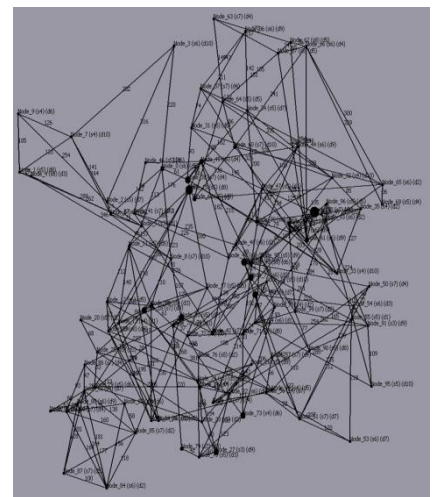
(f) Multi-Level SoS N Optimum Candidate



(g) Multi-Level SoS O with Node Status

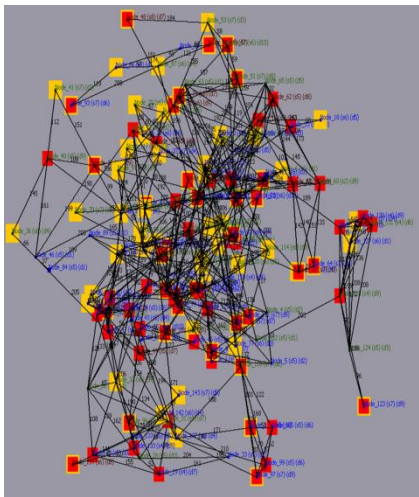


(h) Multi-Level SoS O Topology

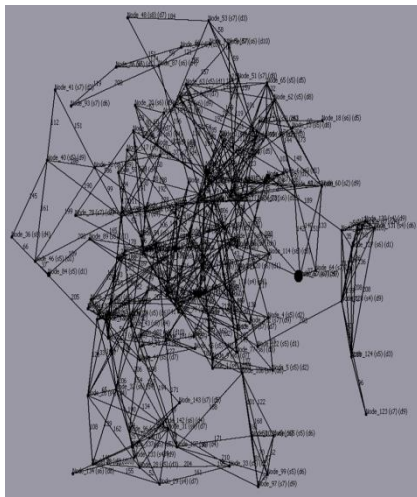


(i) Multi-Level SoS O Optimum Candidate

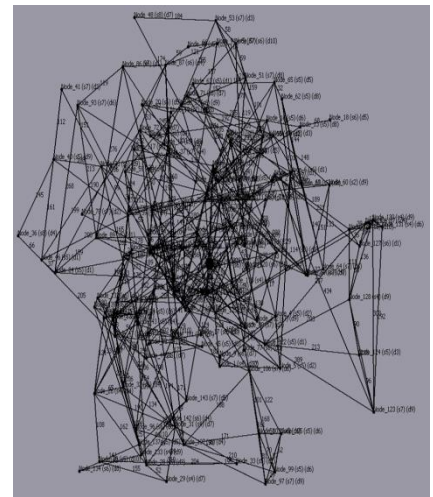
Figure 6.18. Set E of Multi-Level SoS Used in the Experiments (see Appendix B)



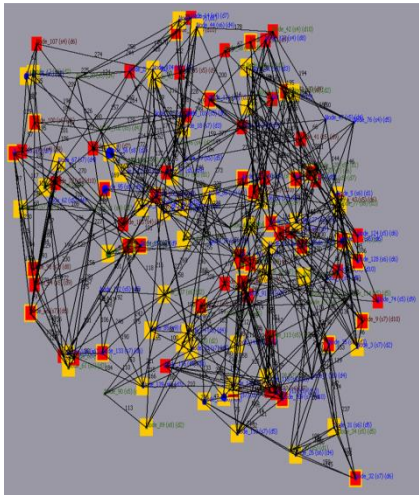
(a) Multi-Level SoS P with Node Status



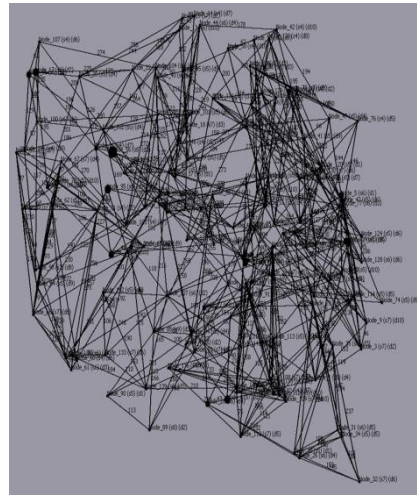
(b) Multi-Level SoS P Topology



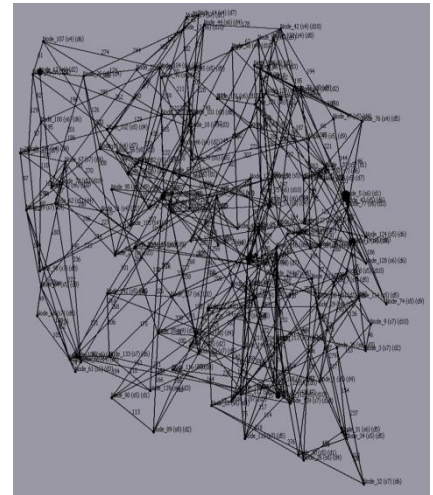
(c) Multi-Level SoS P Optimum Candidate



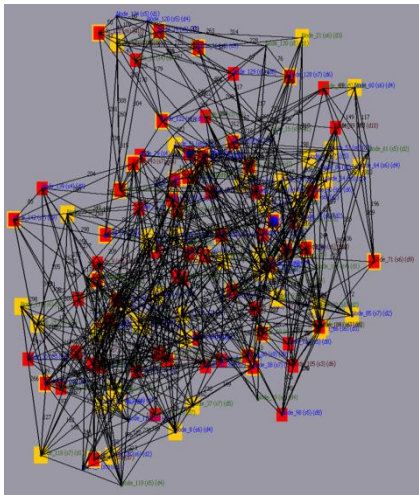
(d) Multi-Level SoS Q with Node Status



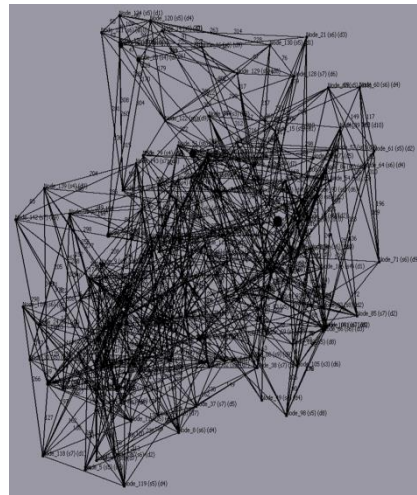
(e) Multi-Level SoS Q Topology



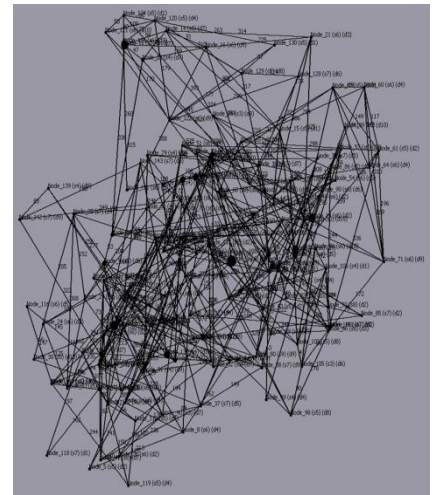
(f) Multi-Level SoS Q Optimum Candidate



(g) Multi-Level SoS R with Node Status



(h) Multi-Level SoS R Topology



(i) Multi-Level SoS R Optimum Candidate

Figure 6.19. Set F of Multi-Level SoS Used in the Experiments (see Appendix B)

Analysing these results, we ascertained that in all instances the security enhancement and risk mitigation process either improved or maintained the overall level of network communication security for the entire multi-level SoS, this is the reason these graphs have been grouped together in this set.

Due to the diverse complex topologies and utilising an evolutionary security risk mitigation approach to evolving the network’s communication paths to increase security, there is no guarantee that significant improvements will be achievable, i.e. not every SoS topology can be improved. In these instances we do see a range of improvements, an excellent example of this is when we review the evolved multi-level SoS O (Figure 6.18-g). This is compiled from 10 SoS each of which contains 10 nodes, with an initial network connectivity of 50%, and has been quantified as being comprised of 51% insecure nodes and 48% blocked nodes which violate the data access policy requirements.

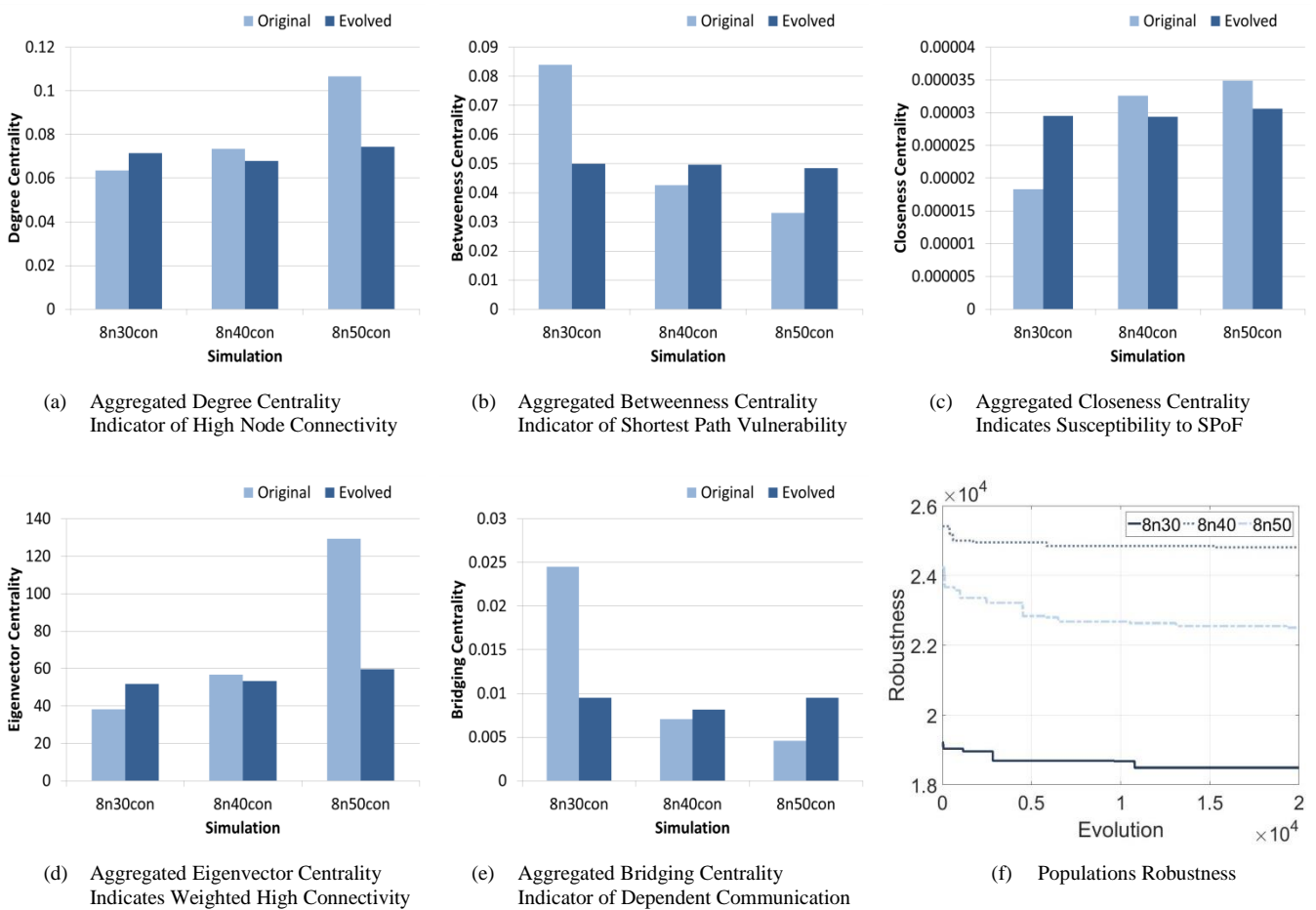


Figure 6.20. Set D Multi-SoS Topological Security Vulnerabilities and Robustness Comparison

While in this instance we do not see an improvement in the network’s security instead it is maintained at 25%, SCRAM is able to quantify and identify all nodes that are insecure and require securing in order to mitigate the risk that they pose to the entire SoS. The multi-level SoS is also evolved to ensure that a secure data path is established between all nodes and SoS, to ensure that data is not forced to traverse via insecure or blocked nodes. In addition, the reported optimum candidate reduces the overall cost (Table 6.8) of the SoS by 49.83% from 79029 to 39649, with a number of irrelevant

and redundant communication paths being removed while establishing and maintaining the necessary links to secure the multi-level SoS.

While there has been a significant reduction in communication links reflected by the network cost, we do see a minor increase of 8.4% to minimum path average and an increase to degree centrality for the graph increasing to 0.030 from 0.024 (Tables 6.7 and 6.8). These values while marginally increased are within an acceptable range, bearing in mind security level has been maintained while reducing the overall associated costs.

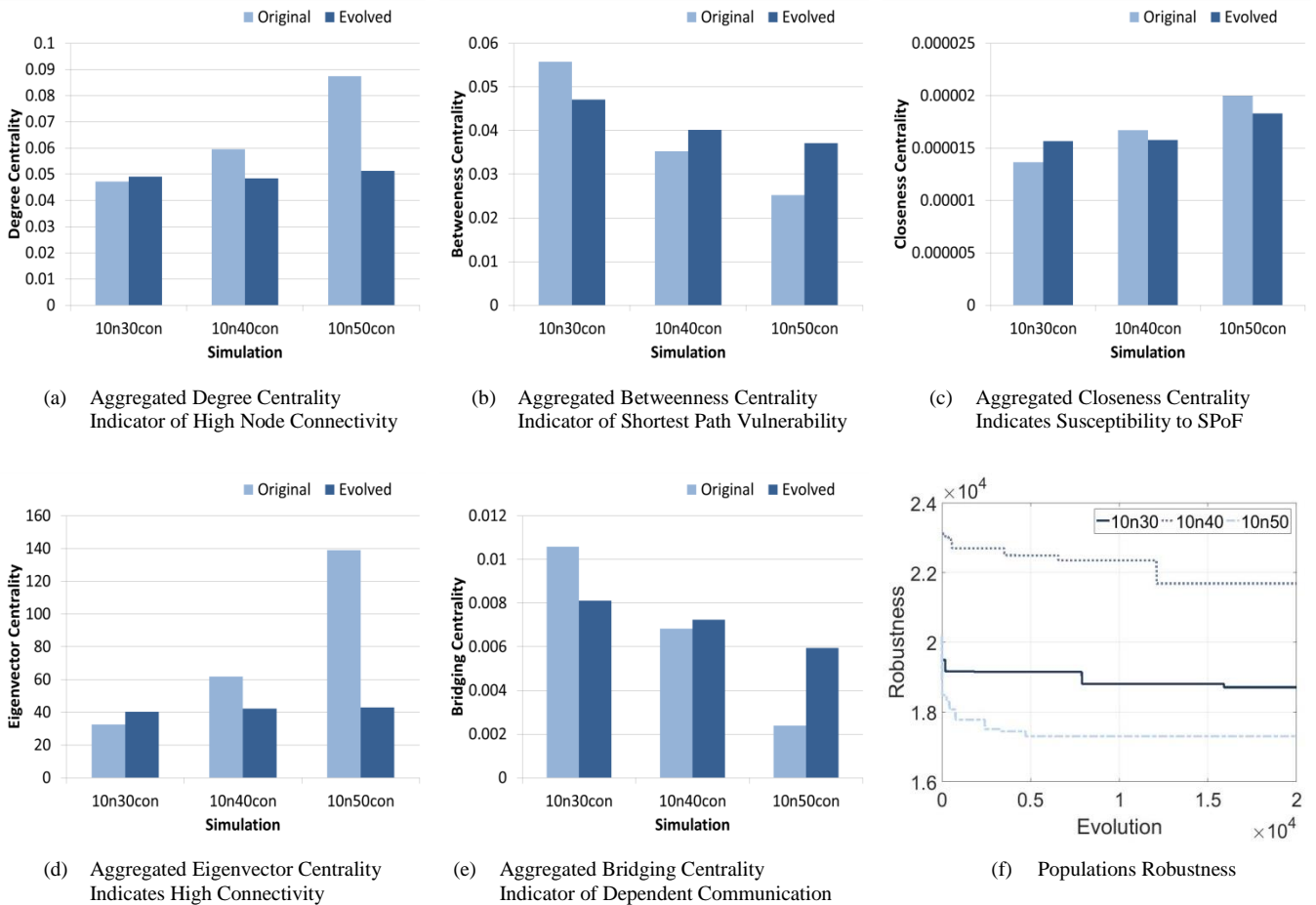


Figure 6.21. Set E Multi-SoS Topological Security Vulnerabilities and Robustness Comparison

Figure 6.21 validates the appropriateness of the proposed optimum candidate for multi-level SoS O, identifying that the network's robustness (Figure 6.21-f) has reduced by 14.28%. There are also reductions to aggregated degree, and eigenvector centralities, with a minor decrease to closeness centrality, with aggregated degree centrality (Figure 6.21-a) reducing by 41.34%, closeness centrality (Figure 6.21-c) decreasing by 9.13%, and eigenvector centrality (Figure 6.21-d) reducing by 69.16%. The reduction to the number of communication links is reflected in the reduction of degree centrality, however it must be noted that the multi-SoS maintains a strong communication network and adequate number of links, i.e. only excessive redundant links have been removed so the lower centrality score is to be expected. The reduction of eigenvector centrality also reflects the strengthening and

appropriateness of the network, as it validates that the number of influential nodes that have the potential to expose the collaborative infrastructures should the nodes be removed or fail, have been significantly reduced.

It is reported though that both aggregated betweenness and bridging centrality has increased for multi-level SoS O, with betweenness centrality increasing from 0.025 to 0.37, and bridging centrality increasing by 148.4% from 0.0024 to 0.0059. In this case we have seen a number of communication links removed from the entire SoS, and the optimum reported candidate has been evolved and selected based on the communication paths between secure and unblocked nodes. With the network consisting of a large quantity of inappropriate nodes for data to traverse across, the security risk mitigation process is limited in how the network can be evolved. In addition, the size, complexity and physical location of the nodes within the topology will also influence how the network is advanced. The SCRAM framework attempts to apply the principles and consider all of these aspects while balancing security priority and risk mitigation, without unduly impacting centrality values, minimum path average, and the overall robustness of the infrastructure, evolving the collaborative environment using only the resources available, in an attempt to enhance the security and robustness of the entire multi-level SoS.

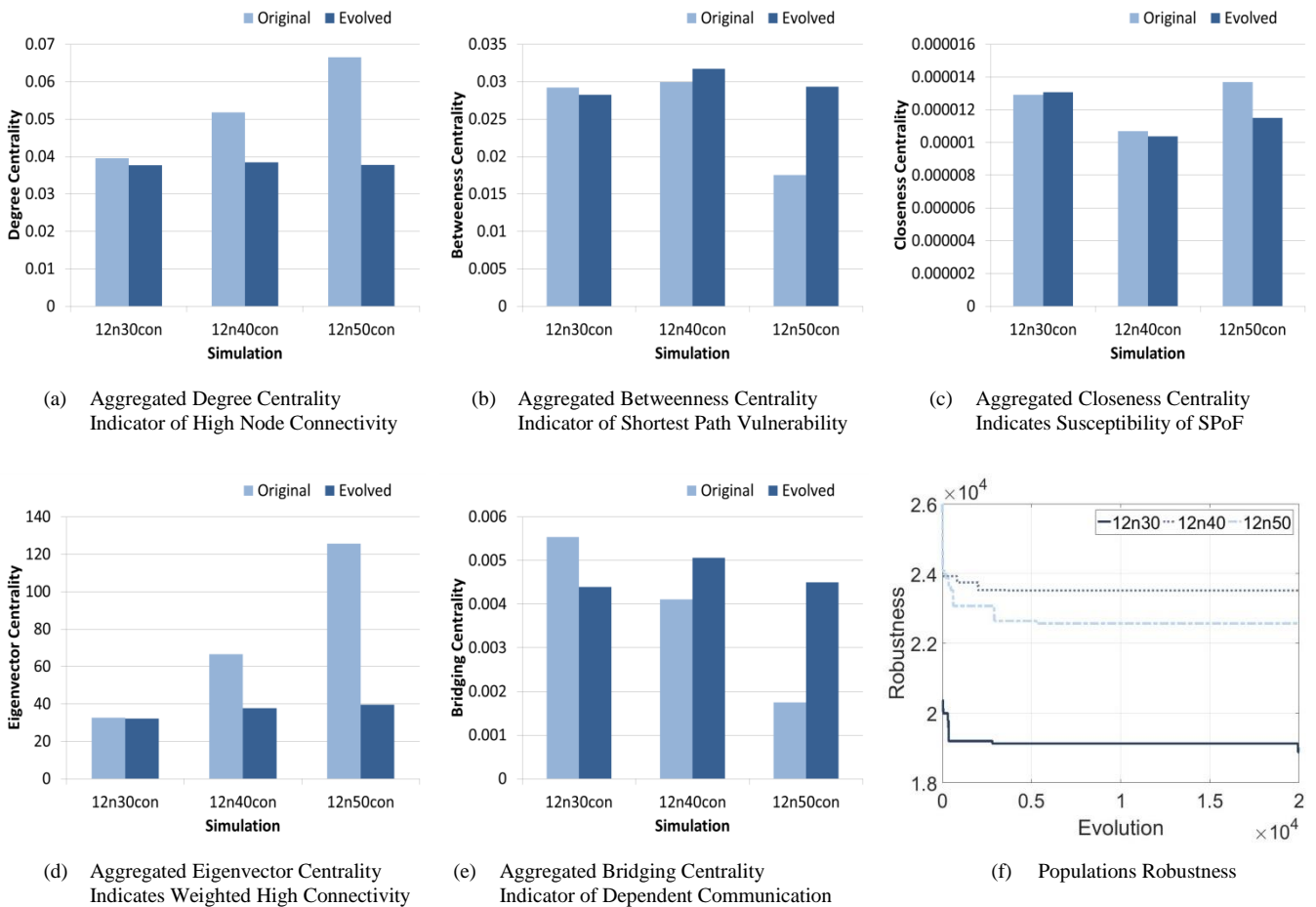


Figure 6.22. Set F Multi-SoS Topological Security Vulnerabilities and Robustness Comparison

6.2.1.3 SCRAM Negative Multi-Level SoS Vulnerability

Performance

This third presented collection of multi-level SoS topologies considers only the vulnerabilities within the topology of the entire multi-level SoS and does not apply the data access policy, characterising topologies such as SoS devised from WSN and IoT. The multi-level SoS contains a minimum of eight and a maximum of twelve distinct SoS, and each has an initial network connection of 40% or 50%. Figures 6.23, 6.24, and 6.25 visualise each of the six conducted experiments that have been categorised under this section.

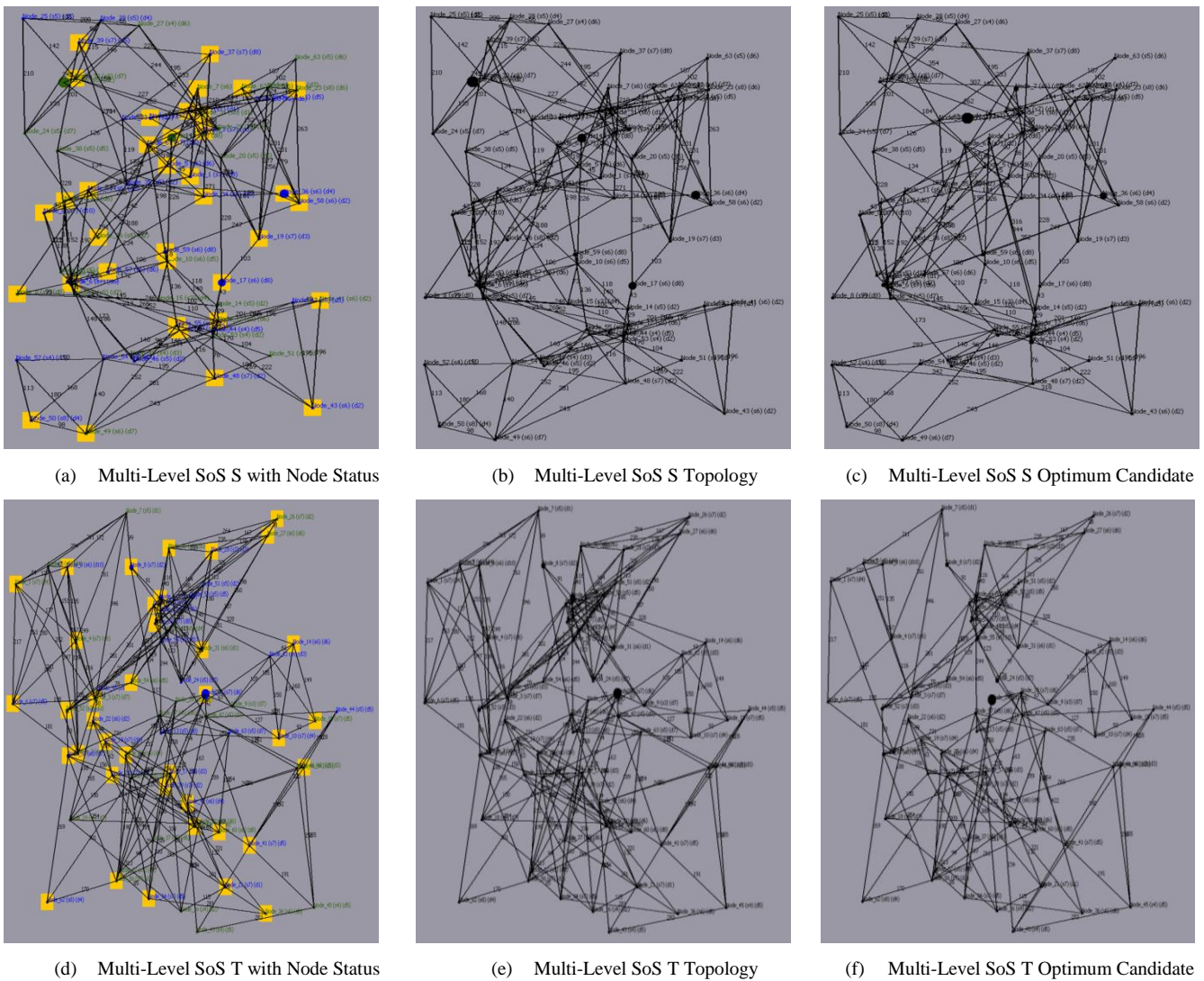


Figure 6.23. Set G of Multi-Level SoS Used in the Experiments (see Appendix C)

These undirected graphs do not visualise the connecting node indicators or individual network colours to allow for network paths and nodes to be imaged clearly and the topology of the SoS to be identifiable. Once more, the first graph visualises the topology and quantified insecure nodes, the second graph clearly presents all nodes (proportional to their computed bridging centrality) and

communication paths, and the third undirected graph presents the optimum evolved candidate topology.

The original multi-level SoS properties for each infrastructure is presented in Table 6.9, with the enhanced security assessment results for the optimum candidate presented in Table 6.10. Figure 6.26 provides a comparison for the aggregated multi-level SoS centrality scores, and Figure 6.27 visualises the population robustness scores for each multi-level SoS.

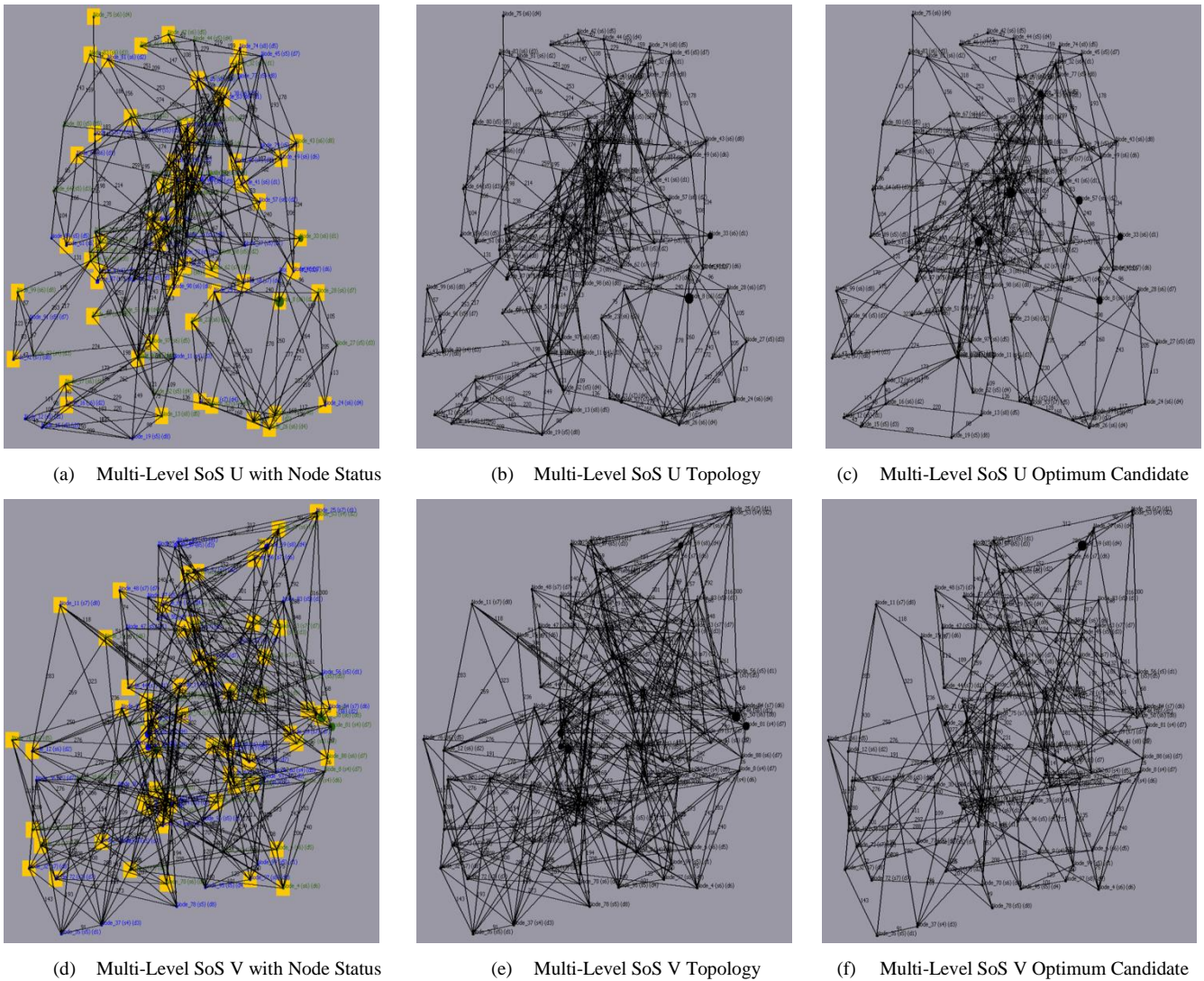


Figure 6.24. Set H of Multi-level SoS Used in the Experiments (see Appendix C)

When we analysed each of these multi-level SoS we quickly ascertained that all of the instances reported a decrease in the overall SoS communication security to varying degrees, hence the reason for being grouped together in this set.

The reason for the decrease in security is due to the evolvement process, while redundant communication links are removed to assist with decreasing network costs and in order to reduce vulnerabilities. We are forcing new connections between collaborative infrastructures in order to increase the connectivity. As previously stated we can't rely upon a single node or communication

path as they increase the potential for SPoF. These single connections to nodes would also increase a node's influence within the network and increase the node's bridging centrality. Should these vulnerable nodes be removed or fail within the SoS, then data transfer would also fail. Potentially this could cause cascading failures to ripple across the infrastructures as objectives fail to be met.

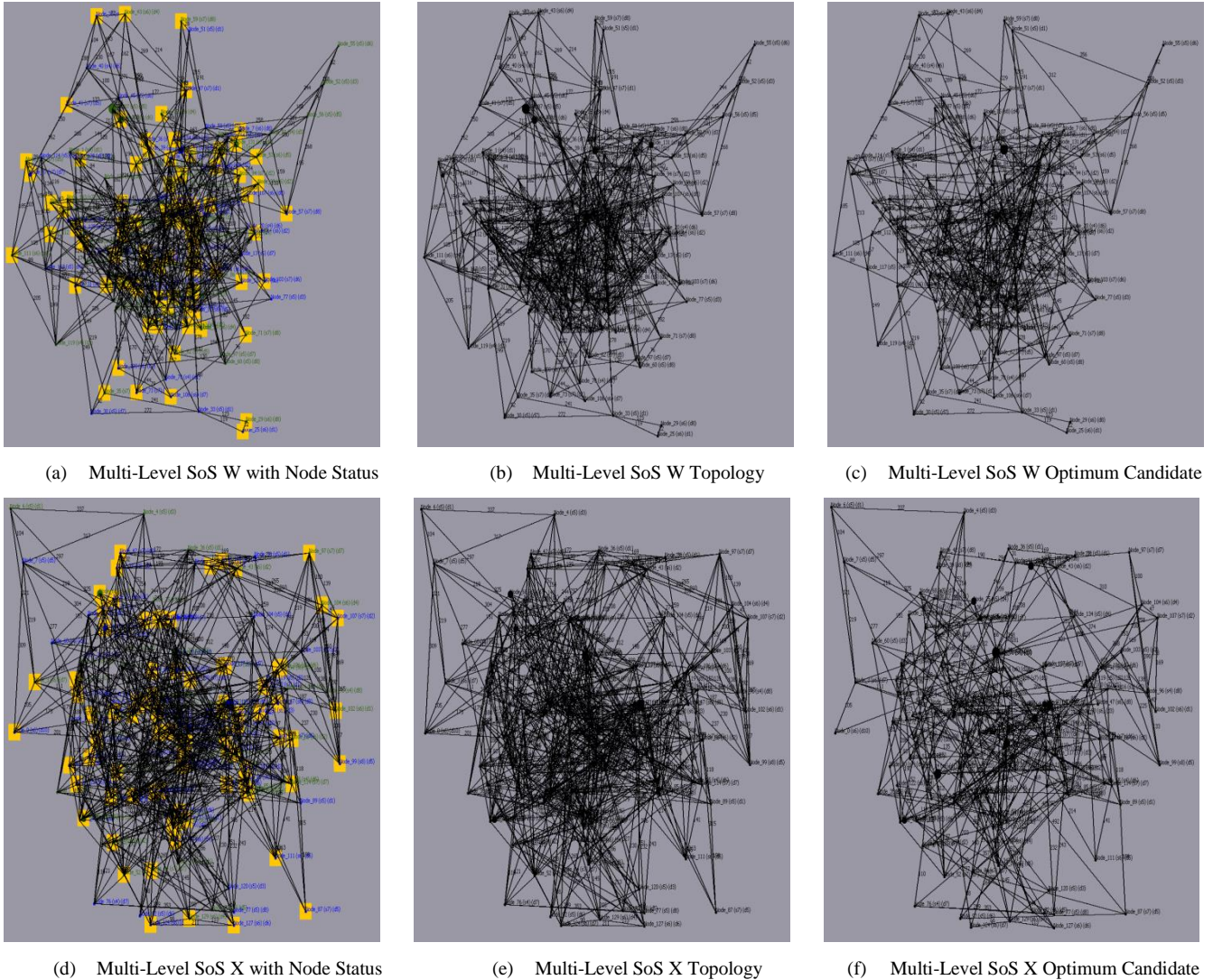


Figure 6.25. Set I of Multi-Level SoS Used in the Experiments (see Appendix C)

Forcing these new connections between SoS in order to establish a robust multi-level SoS, ensures that SoS have an increased robustness against single nodes failing, but in these instances each of the multi-level SoS consisted of infrastructures formed with over 50% of their nodes being quantified as insecure. Therefore, when new connections are established to insecure networks it will immediately impact the overall communication security for the entire multi-level SoS. In these instances the SCRAM framework accurately identifies and reports these issues, including identifying the nodes and connections which are insecure and require immediate attention, as visualised in the undirected graphs (Figures 6.23 - 6.25).

The physical topology of the collaborative infrastructures, i.e. static node location, can also impact security enhancement reconfiguration and the placement of communication paths, as the SCRAM framework attempts to balance risk mitigation, security and node vulnerabilities, while trying to not unduly impact centrality factors and minimum path average, for example. While these multi-level SoS have been categorised as negative as when the evolutionary security risk mitigation principals were applied the security was quantified as insecure, the SCRAM framework reconfiguration of the SoS does result in several positive outcomes, validating the accuracy and corroborating the usefulness of the applied methods and the SCRAM framework.

Table 6.9. Multi-Level SoS Sets G-I Unevolved Infrastructure Properties Comparison

Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.23-a	8	8	40%	0.07629292	59%	26120	465.45438	56% / 0%
6.23-d	8	8	50%	0.04454687	32%	37299	551.18005	53% / 0%
6.24-a	10	10	40%	0.05751392	48%	51040	524.9497	64% / 0%
6.24-d	10	10	50%	0.053597197	33%	76226	535.05634	62% / 0%
6.25-a	12	12	40%	0.04225352	32%	90461	482.6998	60% / 0%
6.25-d	12	12	50%	0.030040385	31%	134069	488.31506	58% / 0%

Table 6.10. Multi-Level SoS Sets G-I Evolved Infrastructure Properties Comparison

Figure	Number of Nodes in Network	Number of Networks	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average
6.23-c	8	8	0.057859723	39%	23065	503.7738
6.23-f	8	8	0.03993855	31%	25531	594.5109
6.24-c	10	10	0.042053193	31%	40136	589.0148
6.24-f	10	10	0.052154213	29%	44348	603.9772
6.25-c	12	12	0.030532854	23%	65822	538.7511
6.25-f	12	12	0.025805186	24%	69480	614.1758

In all instances we see improvements to the multi-level SoS overall robustness score, which is quantified using the five key parameters (security grade, highest bridging centrality, degree centrality, minimum path average, and cost) as discussed in Section 4.6.3. This decrease in the robustness scores corroborates that each of the evolved multi-level SoS is more appropriate. For example, each of the networks in this set has a reduced eigenvector centrality score, meaning there has been a reduction in influential nodes in the network that cause dependencies within the SoS topology. While we do see notable increases in the aggregated betweenness and bridging centralities, these are not excessively high and remain within tolerable boundaries. With the removal of excessive and wasteful redundant communication paths the moderate increase to these values is to be expected. Furthermore, these minor increases will not unduly impact the functionality of the network or excessively increase the risk to particular nodes.

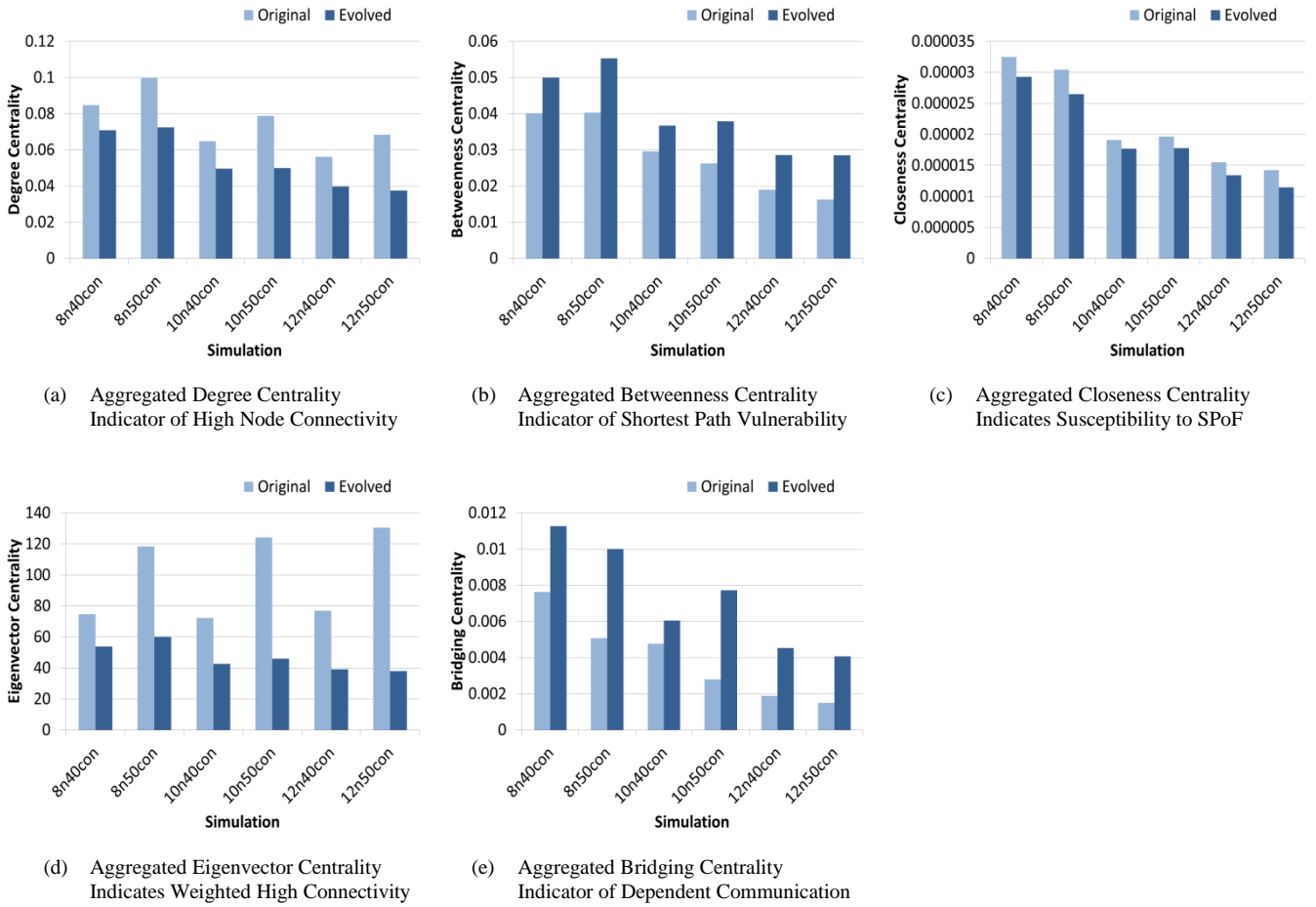


Figure 6.26. Set G-I Multi-Level SoS Topological Security Vulnerabilities Comparison

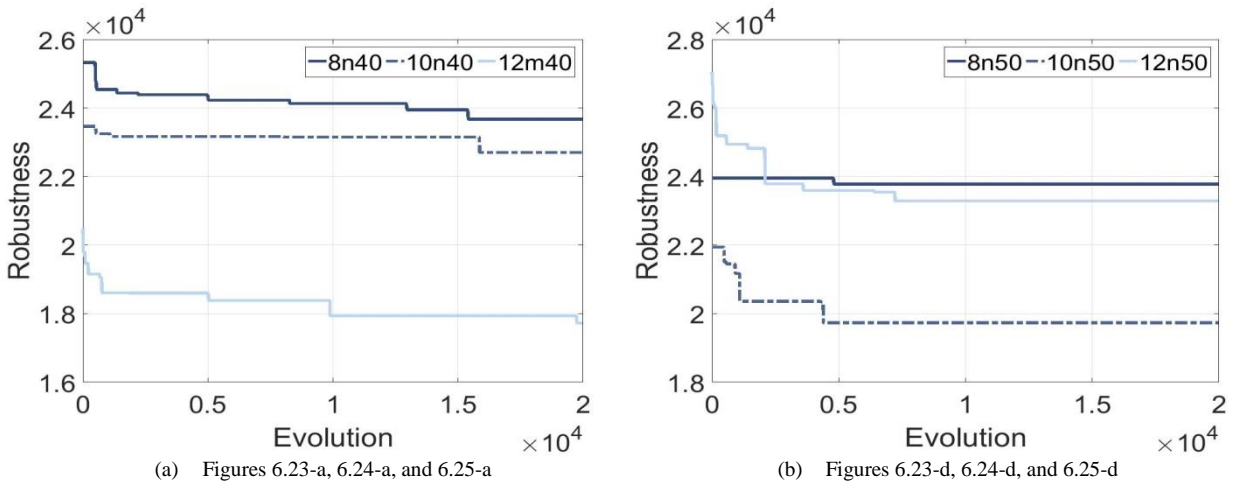


Figure 6.27. Sets G-I Populations Robustness Comparison

Due to the random nature of the evolutionary network evolution along with the dynamic nature of the SoS, potentially we could re-run the security risk mitigation process against the same set of multi-level SoS which could potentially result in positive outcomes, where security is either maintained or improved. Additionally, the SCRAM framework has also quantified and identified all nodes within the multi-level SoS that are insecure. By taking action and rectifying the vulnerabilities of insecure nodes, they would no longer pose a risk to the infrastructure or the data which is to traverse across the

distinct SoS. Therefore in theory, these nodes would be quantified as secure, the security level of the collaborative infrastructure would be increased due to the risks being mitigated, and then these nodes could be considered during the multi-level SoS security enhancement configuration when the security risk mitigation process is applied.

6.2.1.4 SCRAM Negative Multi-Level SoS Vulnerability and Data Access Performance

The final presented collection of multi-level SoS topologies considers both the vulnerabilities and data access problem, with both of these elements being prioritised in the security risk mitigation technique. These multi-level SoS characterise topologies formed from varying ICT networks, with each multi-level SoS being formed from various sizes of SoS and nodes, with different initial network connections percentages. Figures 6.28, 6.29, and 6.30 visualise each of the six conducted experiments which have been categorised under this section. These undirected graphs again do not visualise the connecting node indicators or individual identifiable network colours, with the first graph visualising the topology of the infrastructure including vulnerable nodes and those which violate the data access policy, the second graph presenting clear unobstructed communication paths and node bridging centralities, and the final graph visualising the reported optimum candidate.

When the multi-level SoS are first generated the SCRAM techniques quantify the properties of the multi-level SoS, these results are presented in Table 6.11. In Table 6.12 we present the security enhanced results generated via the security risk mitigation process, and Figures 6.31 and 6.32 present the aggregated multi-level SoS centrality scores and the robustness for each infrastructure consecutively.

Like the negative simulations in Section 6.2.1.3, the reported optimum candidates for all multi-level SoS in these experiments negatively impact the infrastructure's communication security in each instance to varying degrees, thus are grouped together in this set of experiments. For each multi-level SoS the optimum reported candidate reports a decrease in security and an increase in minimum average path length, due to the removal of excessive and redundant communication paths, and the forced establishment of new connections between nodes in distinct SoS. Each of the experimental collaborative infrastructures shows that over 50% of the collaborative devices are quantified as insecure, and report that over 47% of the nodes in the infrastructure violate data access policies.

These highly complex and insecure SoS are then forced to establish new data links between each other in order to not only guarantee a secure communication route across the entire multi-level SoS between secure nodes and those which do not breach data access, but also must establish connections

between SoS via connecting nodes which are deemed insecure or inappropriate due to them being the only secondary option available.

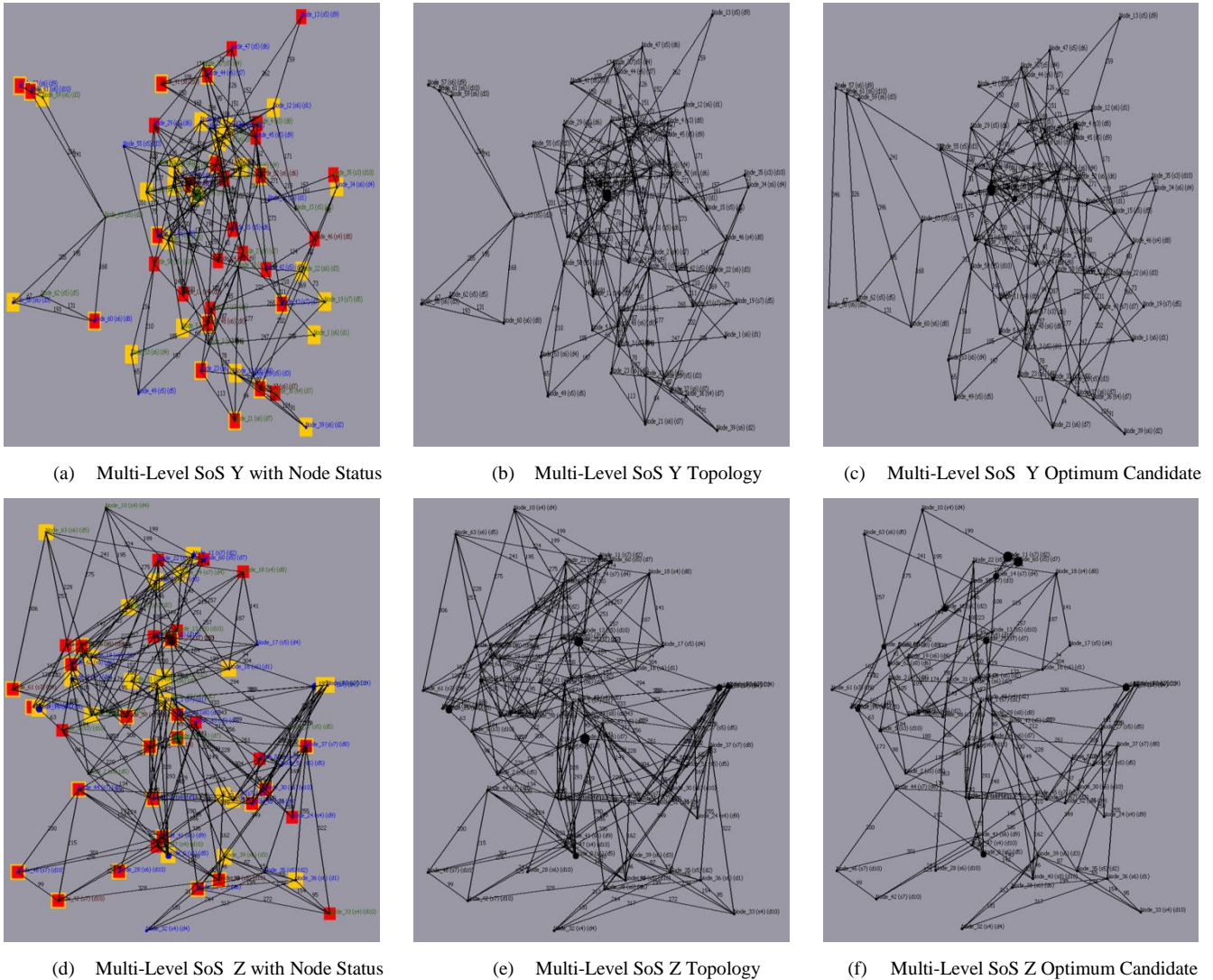


Figure 6.28. Set J of Multi-Level SoS Used in the Experiments (see Appendix D)

The objectives of the SCRAM framework is to ensure that alternative data paths exist between infrastructures to prevent single nodes being relied upon, and to reduce dependencies and potential SPoF within the collaborative infrastructure, to mitigate risks and increase SoS robustness, and prevent cascading failures from rippling across the entire collaborative infrastructure. Therefore, establishing secondary connections between distinct SoS is essential.

When attempting to enhance the security of an SoS configured of a large number of insecure and inappropriate nodes, it is not surprising that the security risk mitigation process negatively impacts the security of both the distinct SoS and the entire multi-level SoS. Likewise, due to the evolutionary risk mitigation process randomly generating candidates, we could potentially run the same experiments on each of the multi-level SoS and the collaborative infrastructure could be enhanced and reconfigured into a different set of solutions with dissimilar end results.

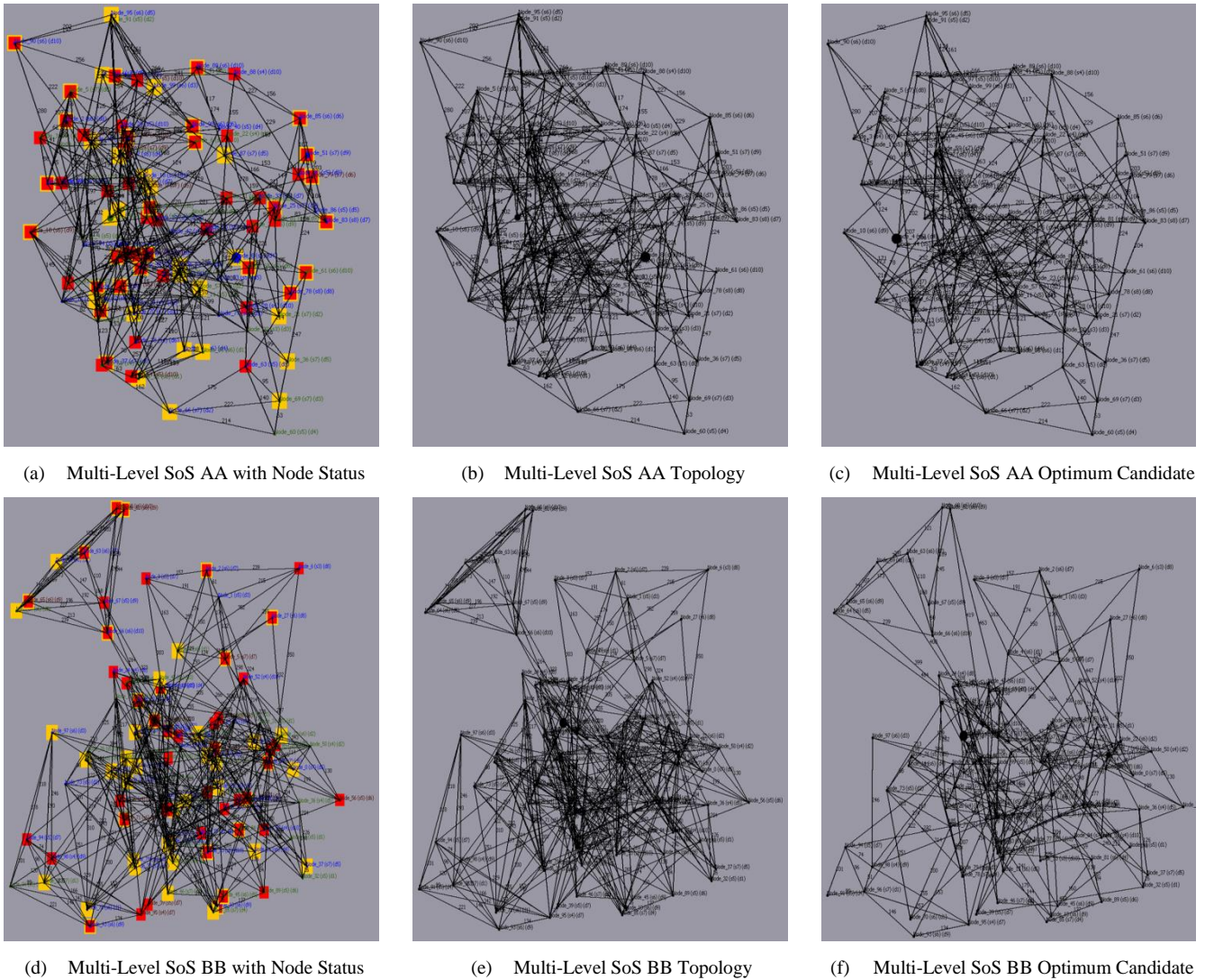


Figure 6.29. Set K of Multi-Level SoS Used in the Experiments (see Appendix D)

We know the reconfigured reported candidates are more appropriate due to the reported decrease in the robustness scores and by analysing centrality factors, for example. These experiments corroborate the functionality of the SCRAM framework and demonstrate its usefulness to identify and visualise vulnerabilities within the SoS topology and report issues.

Analysing the topology of multi-level SoS AA (Figure 6.29-a) for example, Node 60 at the bottom of the graph, after the security enhanced candidate is reported (Figure 6.29-c) this node remains isolated with no secure connection being established between the other secure nodes within the multi-level SoS. SCRAM upholds the principles that all nodes within a network must not be cut off from its own infrastructure. Viewing the topology of the multi-level SoS, Node 60 is primarily surrounded by insecure and blocked nodes, with 51% of the nodes within the collaborative infrastructure quantified as insecure and 55% identified as violating the data access policy. SCRAM during the security reconfiguration of the infrastructure has attempted to assure that the applied techniques did not unduly

impact network centralities and minimum path average etc., and when we review aggregated centrality scores (Figure 6.31) and robustness level (Figure 6.32) this is corroborated.

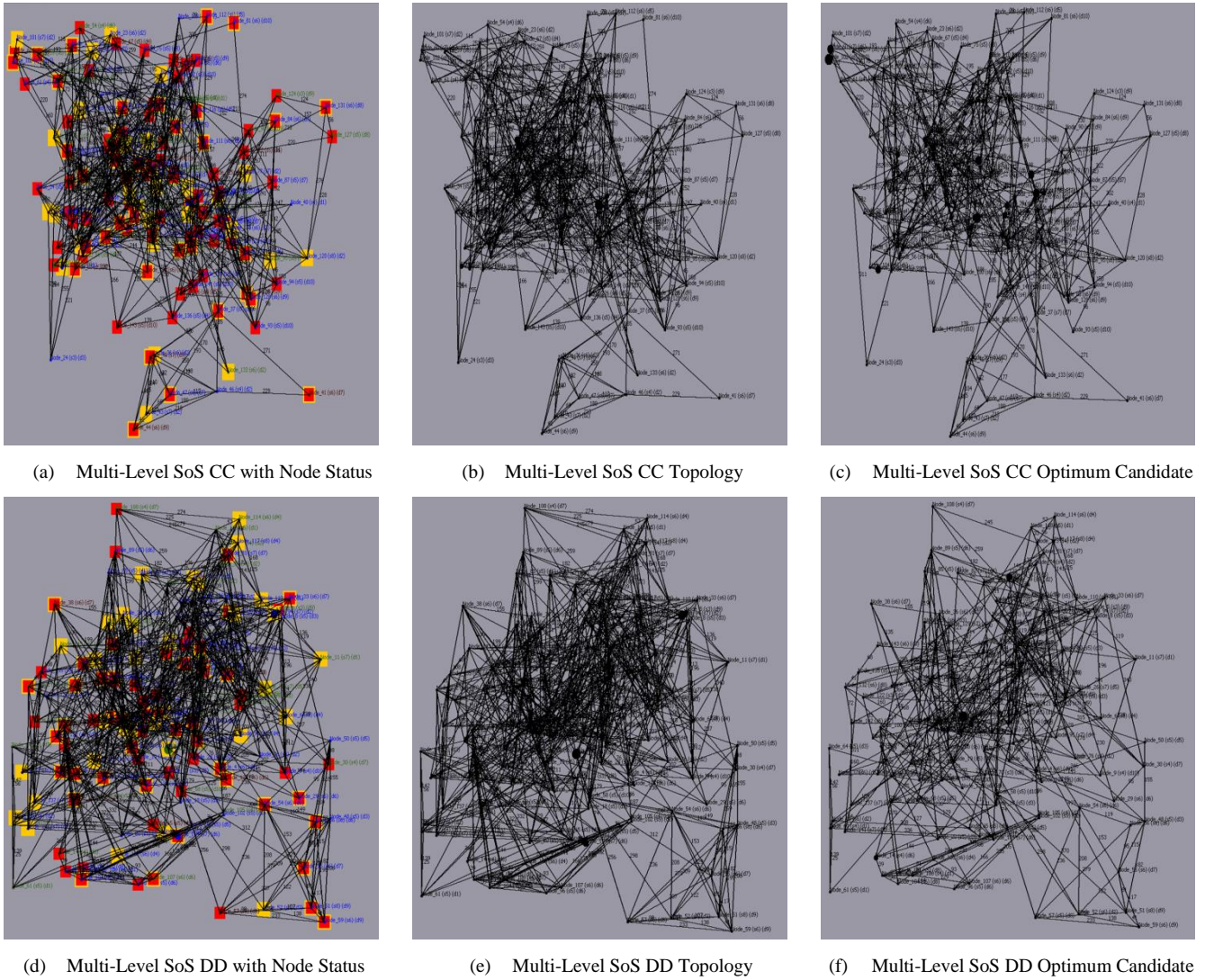


Figure 6.30. Set L of Multi-Level SoS Used in the Experiments

Table 6.11. Multi-Level SoS Sets J-L Unevolved Infrastructure Properties Comparison

Figure	Number of Nodes in Network	Number of Networks	Connection	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average	Number of insecure/blocked nodes
6.28-a	8	8	40%	0.06502819	46%	23961	503.27878	55% / 47%
6.28-d	8	8	50%	0.056835655	35%	39648	525.0496	65% / 50%
6.29-a	10	10	40%	0.043496177	45%	52487	485.4875	51% / 55%
6.29-d	10	10	50%	0.060606036	26%	77185	526.11694	52% / 49%
6.30-a	12	12	40%	0.040874634	40%	93165	486.76895	51% / 65%
6.30-d	12	12	50%	0.04136708	25%	136466	449.02203	53% / 49%

Table 6.12. Multi-Level SoS Sets J-L Evolved Infrastructures Properties Comparison

Figure	Number of Nodes in Network	Number of Networks	Degree Centrality of Graph	Communications Security of Graph	Cost	Minimum Path Average
6.28-c	8	8	0.047107022	23%	21486	529.6607
6.28-f	8	8	0.045058876	32%	22045	562.04565
6.29-c	10	10	0.051535763	21%	39528	555.4374
6.29-f	10	10	0.052772615	24%	42666	616.7204
6.30-c	12	12	0.031123796	20%	65657	540.9086
6.30-f	12	12	0.032502703	19%	67818	589.45776

As the framework has identified that Node 60 is isolated, as stated we could run the risk mitigation process again and force additional reconfiguration to consider alternative solutions. We also have an opportunity to rectify the identified issues of vulnerable nodes, and mitigate their associated risks to assure their security. This would immediately enhance the overall security of the entire collaborative infrastructure, and would reduce the number of insecure nodes in the SoS, therefore, increasing the number of secure nodes for security reconfiguration and construction of secure communication routes across the SoS topology.

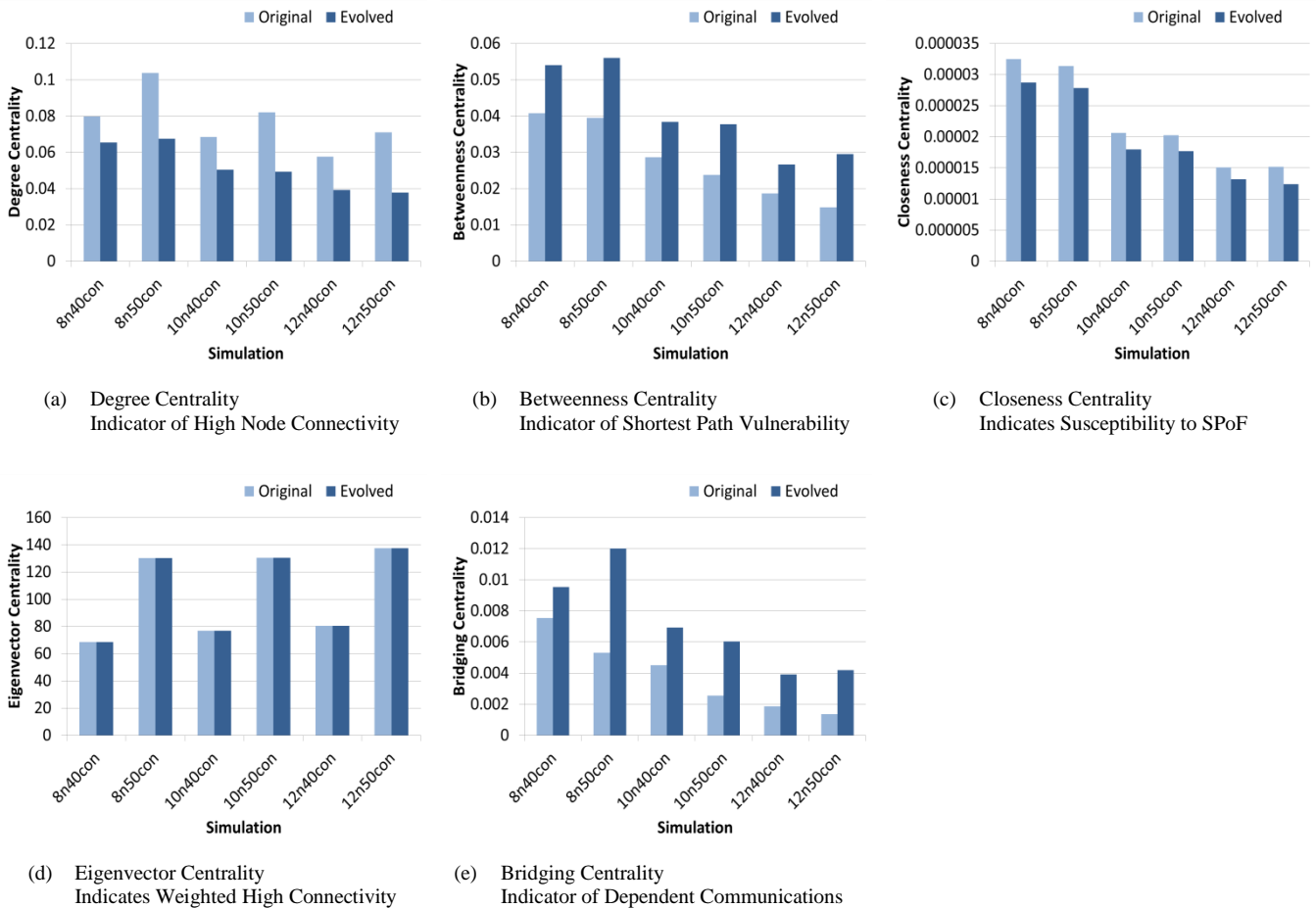


Figure 6.31. Set J-L Multi-Level SoS Topological Security Vulnerabilities Comparison

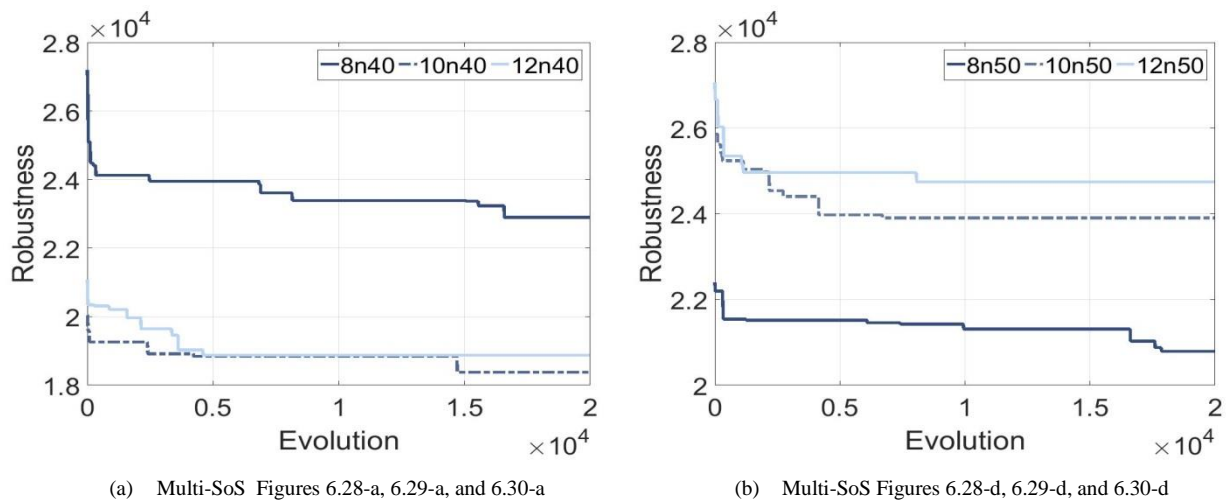


Figure 6.32. Sets J-L Multi-Level SoS Robustness Comparison

6.3 Summary

The simulations used in these experiments are the closest representations of real world multi-level SoS we can use to evaluate the theoretical principles proposed. The complexity and dynamic nature of SoS means applying new assessment methods directly to the physical SoS could result in serious consequences and failings. We can't simply turn off a deployed SoS environment in order to conduct vigorous testing. This verifies the most effective means to ensure issues within the proposed SCRAM framework do not arise, is by vigorously testing and evaluating the techniques, prior to investing considerable time and resources into applying the methodology to a large physical multi-level SoS.

This chapter presented evidence corroborating the usefulness of the applied theoretical techniques, demonstrating the benefits in regards to the framework's accuracy and capabilities when applied to multi-level SoS. Although we must note that similar to the above chapter, while analysing and evaluating these experiments we see encouraging results, the framework has not been fully distributed against a larger physical multi-level SoS and truly only reflects the fixed set of vulnerabilities and simulated multi-level SoS configured environments. Therefore, it is difficult to ascertain if the results are specific to the configured simulated environments in which they have been evaluated or are typical examples.

Chapter 7

Conclusion and Future Work

The dynamic nature, complexity, and size of SoS, makes it extremely difficult to identify risks and assure the security of these heterogeneous multi-level SoS. As organisations and cities continue to see the financial gains of collaborating and integrating external ICT with their systems in order to take advantage of the many benefits it affords, the size and complexity of these types of collaborative infrastructure will continue to increase. For this reason, monitoring these large complex multi-level SoS will pose additional challenges, and increases to the associated risks will need to be considered such as those attributed to operational independence, managerial independence, evolutionary development, emergent behaviour, and geographic behaviour. If left unresolved and new solutions are not researched and developed, then we will see additional failings to deployed SoS such as those that are directly attributed to unidentified vulnerabilities, interdependency, complexity, and cascading failures.

Similarly, as those with malicious intent continue to recognise the true value of data and its disruption, and as callous attacks continue to evolve and malicious attackers take advantage of the weakness and associated vulnerabilities of SoS, it is vital that novel solutions are developed in order to assure the security of system components and the data which traverses and is stored within SoS environments, and methods that increase the overall robustness of the collaborative infrastructures are advanced.

Existing techniques are in general highly theoretical or have failed to be applied to such a large scale dynamic decentralised environment that consists of a large number of diverse and distinct infrastructures. This thesis has presented a SeCurity Risk Analysis and Mitigation (SCRAM) framework, along with evolutionary techniques and security risk mitigation methods, and discussed how these principles can be applied to dynamic complex SoS and multi-level SoS in order to overcome the limitations of existing solutions that attempt to identify and mitigate risks within SoS environments.

The presented experiments in this thesis consider specific SoS types and configurations, and quantify risks into security rankings based on a collection of identified real world vulnerabilities. We acknowledge that further work must be undertaken to both corroborate initial results and assess the true effectiveness of the framework and principles on physical dispersed complex multi-level SoS, and to establish the accuracy of the evaluated methods presented.

In this chapter we discuss this thesis and present an overview of the novel contributions developed in order to overcome the limitations of existing methodologies, corroborating the constructive gains of this proposed work. The limitations of our proposed framework and principles are also summarised in the section, along with future research and developments that could be conducted in order to advance our work and debate other areas in which the principles could be applied. Finally, our concluding observations are conferred, highlighting the accomplishments of the work presented in this thesis.

7.1 Thesis Summary

Chapter 1 of this thesis provides a brief overview of SoS, introducing the concept and providing an insight into the risks that expose these infrastructures and the challenges that impede present security and risk solutions used to secure and mitigate risks within their topologies. In this chapter, we also present the aims and objectives of this thesis, and discuss the motivation for our conducted research. Additionally, the novel contributions achieved in order to overcome the limitations of current research and developments, and our research findings are outlined.

Chapter 2 of this thesis focuses on providing sufficient background for the reader in order for them to comprehend the research area. This chapter describes the rewards and challenges associated with SoS, and the inadequacies that currently exist within this area of research. This chapter also outlines the associated risks and assessment methods that attempt to identify and quantify the risks associated with large dynamic multi-level SoS. Including methodologies that endeavour to model vulnerabilities and the SoS architecture, such as attack graph generation techniques.

Chapter 3 of this thesis provides a critical review of conducted research associated with SoS security, and SoS risk analysis, assessment, and modelling. Reviewing how these techniques and methodologies are applied in regards to assuring SoS security, their effectiveness in identifying vulnerabilities, and how they can be improved to provide effective solutions to the challenges outlined.

Chapter 4 of this thesis presents an outline and justification of the research methodology and our novel principles and framework, which have been researched and developed in order to overcome the limitations of existing techniques and the associated challenges. Firstly, a detailed outline of the SCRAM framework is provided, which includes a comprehensive overview of the structure and design of the proposed SCRAM framework, including a detailed description of the framework's processing stages. In addition, the principal algorithms and methods which are implemented into the framework in order to meet our aims and objectives are discussed. This includes summarising the importance of the principles and how their operation will assist to meet our objectives.

Chapter 5 of this thesis defines how the theoretical principals and the proposed SCRAM framework were implemented, discussing the configuration of the essential methods and simulation environment. This section also provides initial evaluation of the proposed framework against the fundamental design requirements. In order to corroborate the effectiveness of the framework and principles, and ensure that the aims and objectives established in the thesis are achieved. The chapter concludes by providing a case study that validates the appropriateness of the SCRAM framework and applied techniques.

Chapter 6 of this thesis presents the experiments generated in order to evaluate the framework and integrated theoretical techniques. These simulated environments are generated and allow us to develop a rich topological environment formed from multiple distinct networks with varying devices and their associated vulnerabilities, and generate good data sets for analysis and evaluation. From these more detailed and complex environments, we can therefore determine that the SCRAM framework and principles are an adequate solution for identifying and mitigating risk within multi-level SoS.

7.2 Aims and Objectives Evaluation

Our primary objective was to conduct detailed research into the challenges, risks, and methodologies that expose SoS. This objective was met, and provided great insight into the inadequacies and inflexibility of existing methods, that fail to identify risks and which leave SoS exposed to attack and risk vectors. By identifying the weaknesses of techniques that have been researched and developed within multiple fields which include cyber security, risk analysis and management, optimisation, and attack graph generation, etc., we were able to ascertain methods that could be developed to support the identification and analyses of multiple risk vectors within SoS environments. Developing innovative methods that could utilise identified risks to accurately measure the security of distinct devices, the robustness of the infrastructure, and the security of the entire collaborative environment, which allowed for us to develop a solution to enhance security and mitigate risks utilising the infrastructure's existing resources only.

The development of the security risk analysis solution assisted greatly in meeting our second objective, evaluation of our experiments corroborate that we can analyse identified risks and calculate the security level for the entire SoS using vulnerability analysis, node property aspects, topology data, and other factors. The improved accuracy of security grades and risk identification allows for us to mitigate risks without introducing additional resources into the SoS infrastructure, and for those risks that cannot be mitigated provides us the means to accurately report them. Meaning risks that are not able to be mitigated could be managed more effectively prior to their exploitation or failure.

Our objective to develop a solution capable of analysing and quantifying the robustness of the SoS based on the relevant data captured from the security risk analysis solution was successfully implemented within our framework and evaluation of our experiments validates the usefulness of this method. This method quantifies a numerical score which represents the overall robustness or appropriateness of the entire collaborative infrastructure and can be used as a comparative variable during risk mitigation processes and used during decision making processes.

In order to successfully mitigate risk it was important to conduct a detailed investigation into optimisation techniques and algorithms, to establish which solutions are both capable of being applied to SoS and multi-level SoS, and that they are capable of mitigating risks effectively. Through critical analysis of existing methods and the application of three different optimisation techniques within the security enhancement and risk mitigation process, we were able to analyse solutions and determine their effectiveness. The accomplishment of this objective is corroborated by the results evaluated within our conducted experiments presented in this thesis.

Through the completion of a case study on WSN and the expansion of our framework to encompass a different risk vector, it establishes that our solution is expandable and overcomes the limitations of existing solutions that are rigid and cannot be easily adapted to encompass additional risk vectors. By meeting this objective we were able to establish the effectiveness of the proposed solution's ability to incorporate and identify dissimilar risk factors, and its capability to be forward compatible.

Our final objective was to validate the solution's ability for identifying and mitigating risks within multi-level SoS, while we were able to successfully meet this objective we had to further consider the implications of connecting SoS to external SoS under independent management, and the new risks that are introduced and have the potential to expose the multi-level infrastructure. While evaluating the framework's methods, we were able to measure security for every distinct device and the entire collaborative infrastructure, and evaluation of the experiments corroborated the effectiveness of the method's application to mitigate risks and enhance or maintain communication security for large complex multi-level SoS.

Fulfilment of these objectives heavily contributed to solving the problematic challenges associated with measuring security between interconnected 'things', the identification and mitigation of risks and interdependencies, and data security in insecure and unencrypted networks. Evaluation of the applied methods and experiments, established that we have accomplished the objectives we presented and that they assisted us in adhering to the established aims of the research. While evaluation of our methods established limitations with our solution (outlined in Section 7.4), the execution of our aims and objectives contribute to identifying and mitigating security risks in multi-level SoS environments.

7.3 Novel Contributions and Publications

Research presented in this thesis offers significant contributions in the field of SoS risk analysis and security.

- An evolutionary SeCurity Risk Analysis and Mitigation Framework, which overcomes the inadequacies and limitations of existing solutions and the challenges of monitoring complex SoS environments. The framework quantifies security scores for the distinct nodes that form the collaborative infrastructure and for the entire SoS. This is achieved by combining the use of established vulnerability scoring techniques and databases into the SCRAM framework. However, to ensure that we generate an accurate security grade for each node, based on the network discovery process, the SCRAM framework assigns every node with further risks probability scores based on the device's software, hardware, firmware, data access level, and, when relevant, external connections between collaborative networks, incorporating these multiple risks scores into a single security grade for each node, and considering centrality factors that further expose the entire SoS to risk, adding extra dimensions to SoS security. The framework can accurately identify, report, and visualise the nodes that pose the biggest threats to the infrastructure, including those which if removed or fail will have the greatest impact on network communication and security. This framework is considered novel as the literature review has not identified existing methods that use such extensive analysis, instead focusing on more specific vulnerabilities and risks. This novel framework considers not only vulnerabilities that expose the infrastructure to attack, but risks associated with the physical network that can impede data communication between collaborative devices, risks associated with dependencies that can result in full and partial cascade failings, the associated risks of high centralities, risks and vulnerabilities that can endanger SoS due to their physical structure and configuration, and risks that are introduced into multi-level SoS when distinct SoS are forced into collaborative relations.
- A statistical robustness algorithm that combines five distinct parameters to quantify the appropriateness of the SoS environment, and assists to determine the optimal network when combined with the evolutionary security risk mitigation algorithm. As emphasis is placed on the robustness level of the network, this value which represents the appropriateness in security and network security configuration can assist security risk mitigation evolutionary algorithms to produce the next generation of improved solutions. This individual score becomes a representative factor which establishes the suitability of the entire SoS topology, and can be used alongside the security level of the network during decision making. This technique is considered novel, as the literature survey did not identify other security methods that can overcome the associated challenges of complexity, the dynamic nature of SoS

topologies, and solutions that utilise such a large number of parameters, risks, and identified vulnerabilities, which when analysed are combined into a single representative robustness score demonstrating the appropriateness of the entire SoS or multi-level SoS topology. Instead existing solutions typically focus upon a specific type of attack or vulnerability, and generally do not analyse such large diverse collaborative topologies and apply their methods to such diverse collaborative infrastructure topologies.

- An evolutionary security risk mitigation algorithm that when applied to a SoS, searches for an optimal combination of communication connections in an attempt to assure data as it traverses across an unencrypted collaborative infrastructure. The applied algorithm overcomes many of the limitations associated with local search techniques, and the basis of the algorithm is to evolve the network via an evolutionary process till an end criterion is met. Ensuring as random mutations are made the older or inadequate solutions die out based on the robustness level of the network quantified using the novel robustness technique, security grade process, data access principles, and centrality values. The algorithm is capable of reconfiguring communication links searching for the optimal secure configuration of network paths for both the internal connections between distinct SoS and their associated devices. In addition the method configures the communication paths between collaborative infrastructures, reconfiguring the entire collaborative infrastructure and individual SoS as a single entity, and reports the optimum secure configuration via the framework to the end user. This algorithm is considered novel as the literature review has shown no existing approach that utilises such an extensive number of metrics for comparison and evaluation during security risk mitigation, nor do they apply their techniques to both single SoS and multi-level SoS.
- A multi-level SoS SeCurity Risk Analysis and Mitigation Framework that adopts a hybrid and scalable approach to secure and mitigate risks in multi-level SoS. This technique overcomes the limitations associated with complex SoS, providing an accurate means to measure, identify, and visualise security and vulnerabilities, to identify and quantify vulnerabilities and mitigate risks, and to measure the robustness of the entire multi-level SoS. This limits the multi-level SoS exposure to failures and attack vectors, with analysis undertaken on multi-level SoS that consist of up to twelve unique heterogeneous SoS. Early identification of high bridging nodes and those nodes with high eigenvector centrality scores, for example, means actions can be taken to reduce these dependencies and reduce potential SPoF, or the consequences that would be caused in the event these nodes failed or were removed from the SoS (i.e. potential repercussions and cascade failings). Additionally, to overcome the inaccuracies of security as new connections are made between connecting nodes, the method is capable of quantifying a new security grade based on the additional risks the new connections pose to devices. The technique is considered novel as the literature review has

shown no existing approach that provides a comparable level of analysis and visualisation when applying attack graph generation methods and combatting the data access control problem in a single solution. In addition, the solution does not only provide a single reported optimal solution, but reports alternative improved candidates for consideration to assist decision makers when having to consider improving security without unduly impacting budgeting restrictions, etc., without impacting upon the system resources on which the SCRAM solution operates.

Aspects of the research undertaken and presented in this thesis have been published in eight academic research journals and conferences, with a comprehensive list of publications being provided at the beginning of the thesis.

7.4 Limitations

The proposed solutions are impacted by several limitations, as discussed in this section.

- **Vulnerability Identification** – While the SCRAM framework can be programmed with adequate network discovery methods, the framework relies upon standardised vulnerability scoring metrics and databases, such as CVSS and NVD. Therefore, the algorithms that rely upon the accuracy of vulnerability identification, vulnerability scoring, node security grades, and network communication security level are vulnerable and limited by the associated issues of these external techniques. CVSS scoring can be inconsistent due to the principles being too theoretical and difficult to apply to real world identified vulnerabilities that rely on human administration to assign scores. With scores being assigned too high as administration is overly cautious or too low as they do not fully comprehend the threat severity. Discrepancies have increased among analysts over recent years, resulting in inconsistent scores. This scoring technique also fails to address misconfigurations for example, and the framework focuses upon software based vulnerabilities. NVD is synchronised to automatically update when new vulnerabilities are identified and published by CVE, however, it cannot be categorised as a real-time vulnerability and reporting mechanism. Often it can take as long as two full working days for NVD analysts to analyse the vulnerabilities and augment the vulnerability attributes. Additionally, NVD does not perform any vulnerability testing to identify new vulnerabilities, and relies upon CVE and other third-parties, thus is limited and vulnerable to their associated strengths and failings. Meaning the systems we are analysing potentially could be exposed to zero-day attacks, due to the slow confirmation of identified issues and assignment of risk probability scores. SoS communication levels and node security scores generated and visualised via the SCRAM framework may also be inaccurate due to identified vulnerabilities

initially being incorrectly analysed and reported. Hence, there is potential for devices to be inaccurately identified as insecure or secure, and this could result in a high rate of false security classifications.

- **Robustness Function Constants** – As previously stated in Section 4.6.2, the robustness function relies upon constants whose values are determined by analysis of an organisation's network, and would be assigned by their security experts and administration. Therefore, the assignments of scores to these constants are reliant upon the skills and knowledge of the administrators, their perception of risk, and removing their own personal bias. There is a possibility for constants to be over or under estimated, and while the main factor of the robustness function is the security level achieved, these constants which represent highest bridging centrality, centrality degree, minimum path average, and associated network cost are vital elements within the robustness function to establish an accurate robustness level and determine the optimal network. For SCRAM to be effective and report the most viable alternative secure configurations of SoS to enhance SoS and multi-level SoS security and robustness, this is something that could result in false negative reporting of candidates and impact the evolutionary process of security risk mitigation.
- **Identifying Behavioural Consequences** – Unfortunately, as we evolve the network and reconfigure connections between distinct devices and infrastructures, we do not have any means to identify the consequences or resulting negative behaviours that could arise due to the newly formed connections. Depending upon the type of infrastructure being evolved, dependencies could have formed between systems and when data communication paths are removed and replaced throughout the SoS, full or partial cascading failures could occur as a direct result. Similarly, as new connections are formed between devices, emergent behaviours could arise either immediately or in the future. Therefore, it is essential that this issue is seriously considered, with further research and development required in order to ensure the SCRAM framework is viable and can be applied to physical multi-level SoS without recommending erroneous network configurations that impede the SoS functionality or introduce failings.
- **Failure Tolerance** – The framework attempts to reconfigure and enhance an SoS topology, specifically communication paths between nodes and external SoS based on security grades, quantified vulnerability status, and data access levels, for example, while not unduly impacting network and node centralities, cost, and minimum path average, etc. Despite the fact the framework clearly will not tolerate a node being disconnected from its own SoS, and attempts to conform to strict parameters to ensure that there is a minimum of one secure route between all nodes, to guarantee that data does not traverse via insecure nodes or those that violate data access, on occasions SCRAM identifies nodes that while remaining connected

within their SoS, fail to be connected across the entire multi-level SoS infrastructure to any other secure nodes. This limitation is particularly prevalent in complex dynamic multi-level SoS that are composed of great numbers of quantified insecure nodes and nodes that violate data access policies. Hence, while the framework reports this issue further development is required in order to rectify this limitation, without unduly impacting SoS security and centralities by perhaps generating and reporting a solution that while it is not evaluated and considered the optimum, assures node security and secure communication routes between nodes as an alternative solution that in addition quantifies the impacts that will occur due to the forced essential connection(s).

- **Deployment Strategy** – Currently the SCRAM framework application is executed upon a single device, which is responsible for network discovery, vulnerability analysis, security risk mitigation process, and for generating undirected graphs and reports. These processes and reports are visualised and presented in a single SCRAM interface. In addition, the framework itself could be a SPoF or should the thresholds be targeted for example, the SCRAM framework would generate inaccurate results and leave systems exposed. To be effective in complex distributed multi-level SoS environments and in order to limit the impact on resources used for processing, this issue needs to be addressed. The complex distributed environments are formed using a variety of SoS, configured from diverse components. When deploying the framework it is vital that the responsibility for the SCRAM processing is assigned to devices with adequate resources, to ensure that we limit the impact of resources and in order to accurately determine the framework's true footprint. If the SCRAM framework is to be deployed within physical SoS which are managerially independent, we also need to address the issues associated with collaborative analysis and report production. For SoS to be able to understand the risks posed to them from their collaborative relations with other SoS, the framework will be required to distribute the results of the analysis and evaluation, to ensure all collaborative infrastructures are informed and have access to the same generated results and warnings. This limitation is particularly prevalent in large dynamic complex topologies, as it is difficult to ascertain how the framework will uphold when applying both vulnerability assessment methods and security risk mitigation techniques on such large distributed environments.

7.5 Future Work

This thesis has presented work that is relevant and could be applied to numerous differing areas. There are various means by which the work could be extended and developed further in order to address other challenges.

- The SCRAM framework could be further enhanced by researching deployment strategies in order to distribute the framework across multiple environments. Presently the framework is executed upon a single device, which is responsible for network discovery, analysis, risk mitigation, and evaluation. However, by deploying the application multiple devices would assist with the method's processing and prevent device resources from being strained. In order to achieve this, associated issues with deployment would need to be examined, along with connectivity, collaborative analysis reports and warning systems, securing globalised network view, accessibility between collaborating organisations, congestion avoidance and control, and limiting the impact of resources used for processing and issues with parallel processing.
- The conducted experiments have been programmed with particular assumptions, thus allowing us to conduct experiments on the SoS and evaluate their results. One such assumption was how much information was shared with SCRAM in order to reconfigure the network, and which for example, influences the removal and replacement of connections to external SoS during the risk mitigation process based on the shared security grades. The framework would be enhanced by establishing how topological information and security risks can be securely shared between unencrypted networked infrastructures. Further analysis would be required in order to determine the impact and potential risk that full disclosure would provide when compared against partial disclosure, along with establishing improvements to quantifying security risks and security grades between distinct SoS.
- The SCRAM framework could be enhanced by conducting research into vulnerability assessment and remediation. SCRAM has the functionality to quantify a node's security score and report the associated vulnerabilities, when nodes are identified as vulnerable it impacts the number of nodes within the SoS that can be utilised for secure communication. However, if low level vulnerabilities were identified and could be simply secured without unduly impacting the functionality of the node or increasing risks to the network, then by having an automated process that applies actions that secure the nodes, it will increase the number of secure nodes within the network, reduce vulnerabilities, directly increase security, and provide additional nodes for selection during the security risk mitigation process.
- The SCRAM framework could be further enhanced by conducting research into authentication, to ensure that the SoS that form collaborative relations do not contain unauthorised devices. The scale and complexity of SoS makes network discovery problematic in itself, as SoS are often managerially independent and have the ability to add and remove devices without informing or seeking permission, for its collaborative associates it can be difficult to establish the legitimacy of devices. Introducing an authentication method would ensure that only authorised devices could access the SCRAM data, and in turn SCRAM would have the functionality to identify unauthorised devices and report them.

- The constants used as part of the robustness function are based on the analysis of the SoS; however, these constants might be different for each SoS within the multi-level SoS. As administrators would be forced to prioritise their individual security based on their individual needs over that of the multi-level SoS. An enhancement to the framework would be to see if we can implement a multi-layered robustness function, which takes into account and reflects the distinct SoS environments. This would also assist to evaluate the use of a single set of constants agreed upon as part of a collaborative agreement, and prevent inaccurate assessments of the topologies.
- The SCRAM framework could be further enhanced by automating the monitoring tool. Currently the network is imported and analysed, with a detailed undirected graph and report being generated and presented, this then allows for the network to be further analysed by conducting the security risk mitigation process. Due to the dynamic nature of SoS, by automating SCRAM to continually monitor a network it would allow the framework to identify changes to the network as devices are removed and added to the network in real-time. This would allow the addition and removal of devices to be analysed and reported, quantifying the changes to security and identifying vulnerabilities that have been introduced or removed from the environment.
- As more organisations transition to the Cloud and take advantage of the many benefits it affords, and as more organisations outsource operations to third parties, monitoring these complex dynamic and geographically dispersed environments will become increasingly problematic. The SCRAM framework will need to be able to understand these complex systems and facilitate the visualisation of such topologies after it conducts the relevant network discovery, but will also have to factor in the additional risks that these third party infrastructures pose to the entire collaborative infrastructure.
- The SCRAM framework is able to generate and simulate a distinct SoS and multi-level SoS. The experiments conducted via the framework were designed to evaluate the success of the applied theoretical solutions proposed in the thesis against the simulated environments and their specific topological configurations, device types, and vulnerabilities. Additional analysis is required to evaluate the framework on larger physical multi-level SoS, to corroborate the appropriateness of the proposed solutions and evaluate the extent of the framework's limitations and benefits.

7.6 Concluding Remarks

Multi-level SoS are gaining prevalence as organisations, governments, and cities take advantage of the many benefits and automated processes ICT delivers, merging their physical assets and cyber

services forming vast complex and geographically dispersed collaborative infrastructures. Despite great investment, development, and research, SoS continue to fail with dire consequences, and as attacks against these types of infrastructure gain prevalence, finding new means of securing and mitigating associated risks becomes more urgent. As tightly coupled bonds form between systems and components, dependencies are generated; these interdependencies contribute to increases with system complexity, and in addition can introduce SPoF and be responsible for additional failings rippling across the SoS causing both partial and full cascade failure.

The proposed SeCurity Risk Analysis and Mitigation Framework endeavours to overcome the associated issues and challenges that impede SoS security and risk analysis. Applying the presented algorithms and techniques within the framework, we can identify and examine all vulnerabilities identified during the risk assessment and quantify a security score for each node, thus, quantifying the communication security level for the entire multi-level SoS, and using the robustness function measure the overall appropriateness of the collaborative infrastructure.

Identified vulnerabilities, security scores, data access levels, and robustness scores support the framework's security risk mitigation process, to enhance the overall multi-level SoS communication security and robustness, without introducing additional security resources into the collaborative infrastructure. Additionally, its quantification of network centralities allows us to consider the problematic relational states between nodes, and identifies nodes that have the ability to expose the entire multi-level SoS to risks. Including, for example, nodes that are influential and can cause dependencies within the infrastructure to form, which increase the risks of cascading failure, and nodes with high bridging centralities that are relied upon to maintain communications across the SoS between nodes.

The SCRAM framework generates detailed reports and graphs, allowing for the multi-level SoS topologies to be visualised in a series of undirected graphs. The use of evolutionary evolution combined with the robustness algorithm, means the security risk mitigation process produces a series of alternative security enhanced solutions for consideration. Meaning the framework provides a diverse number of recommend solutions that all mitigate risk and enhance security, and that support decision making processes when having to balance cyber security, risk levels, identification of topological vulnerabilities (centrality scores), and financial restrictions.

Analysis of the conducted experiments, corroborate that the framework and proposed techniques can succeed in enhancing multi-level SoS security and mitigate risks, and that they can overcome the challenges and issues associated with SoS and assuring their security. The SCRAM framework provides a practical means for individual networked components and the entire multi-level SoS to be monitored, using vulnerability analysis, node property aspects, topology data, and consider other factors including risks associated with high-centrality nodes, and the likelihood of violating access

control requirements, in order to identify risks, vulnerabilities, and interdependent links, thus, providing a means to prevent security issues with future multi-level SoS developments and infrastructures, and enhance SoS security by providing the means to improve security and mitigate risks without adding additional resources into the multi-level SoS.

References

- [1] Cabinet Office, 21 October 2011, "Keeping the Country Running: Natural Hazards and Infrastructure," [online] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/78901/natural-hazards-infrastructure.pdf [accessed January 2014].
- [2] Lever, K.E., Kifayat, K., Merabti, M., "Identifying interdependencies using attack graph generation methods," 2015 11th International Conference on Innovations in Information Technology (IIT), Dubai, pp. 80-85, 1-3 November 2015.
- [3] Ncube, C., "On the engineering of systems of systems: Key challenges for the Requirements Engineering community," 2011 Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), pp. 70-73, August 2011.
- [4] Badger, M., Bushmitch, D., Agnish, V., Cozby, R., Fikus, J., Halloran, F., Chang, K., McCabe, P., Erramilli, S., "Laboratory-based end-to-end network System of Systems Integration, design and risk reduction: Critical activity for System of Systems Integration Directorate and the Army," MILCOM 2012 - 2012 IEEE Military Communications Conference, pp. 1-6, 2012.
- [5] Commons Select Committee, 25 January 2016, "RBS, HSBC and Barclays IT system failures," [online] <https://www.parliament.uk/business/committees/committees-a-z/commons-select/treasury-committee/news-parliament-2015/it-system-failures-correspondence-15-16/>, [accessed January 2017].
- [6] Kurtz, A., 8 August 2016, "Delta Malfunction on Land Keeps a Fleet of Planes From the Sky," [online] http://www.nytimes.com/2016/08/09/business/delta-air-lines-delays-computer-failure.html?_r=0 [accessed: January 2017].
- [7] Delta News Hub, 2016, "Operations," [online] <http://news.delta.com/operations/see-all?page=4> [accessed January 2016].
- [8] Starr, S., 31 October 2012, "Costs and Consequences of the Fukushima Daiichi Disaster," [online] <http://www.psr.org/environment-and-health/environmental-health-policy-institute/responses/costs-and-consequences-of-fukushima.html> [accessed February 2014].
- [9] Hilton, S., 26 October 2016, "Dyn Analysis Summary of Friday October 21 Attack," [online] <http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/> [accessed March 2017].
- [10] Yahoo, 14 December 2016, "Yahoo Security Notice December 14, 2016," [online] <https://help.yahoo.com/kb/SLN27925.html> [accessed March 2017].

- [11] Martin, B., Titcomb, J., 7 November 2016, “Regulators could fine Tesco Bank over cyber attack,” [online] <http://www.telegraph.co.uk/business/2016/11/07/tesco-bank-to-freeze-customer-transactions-after-hacking-attack/> [accessed March 2017].
- [12] Knapp, E.D., Langill, J., “Industrial Network Security,” (Second Edition), Securing Critical Infrastructure Networks for Smart Grid, SCADA, and other Industrial Control Systems, 2015.
- [13] Sanchez, J., Claire, R., Hadjsaid, N., “ICT and Electric Power Systems Interdependencies Modeling,” Proceedings of International ETG-Congress 2013 Security in Critical Infrastructures Today, Symposium 1, 5-6 November 2013.
- [14] Pacheco, L.A.B., Gondim, J.J.C., Barreto, P.A.S., Alchieri, E., “Evaluation of Distributed Denial of Service threat in the Internet of Things,” 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, pp. 89, 2016.
- [15] Wang, P., Lin, W.H., Kuo, P.T., Lin, H.T, Wang, T.C., “Threat Risk Analysis for Cloud Security Based on Attack-Defense Trees,” 2012 8th International Conference on Computing Technology and Information Management (ICCM), vol. 1, pp. 106-11, 24-26 April, 2012.
- [16] Waller, A., Craddock, R., “Managing runtime re-engineering of a System-of-Systems for cyber security,” 2011 6th International Conference on System of Systems Engineering (SoSE), pp. 13-18, 27-30 June 2011.
- [17] Zahra, B.F., Abdelhamid, B., “Risk analysis in Internet of Things using EBIOS,” 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 1-7, 2017.
- [18] Montibeller, G., von Winterfeldt, D., “Cognitive and Motivational Biases in Decision and Risk Analysis,” Risk Analysis, vol. 35, no. 7, 2015.
- [19] International Standards Organisation, ISO/TR 31004:2013, Risk Management – Guidance for the implementation of ISO 31000, 2013.
- [20] Ijure, V.M, Laughter, S.A., Williams,R.D., “Security issues in SCADA networks,” Computers & Security, Volume 25, Issue 7, pp. 498-506, October 2006.
- [21] Lu, Z., Lu, X., Wang, W., Wang, C., “Review and evaluation of security threats on the communication networks in the smart grid,” MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, San Jose, CA, pp. 1830-1835, 2010.
- [22] M. Jamshidi, Ed., System of Systems Engineering: Innovations for the 21st Century. Hoboken, NJ: Wiley, 2009.
- [23] Kotov, V., “Systems of Systems as Communicating Structures,” Technical Report HPL-97-124, HP Labs, October 1997.
- [24] Maier, M., “Architecting Principles for Systems-of-Systems,” Systems Engineering, vol. 1, pp. 267-284, 1998.

- [25] Dahmann, J. S., Baldwin, K.J., “Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering,” 2008 2nd Annual IEEE Systems Conference, pp. 1-7, 7-10 April 2008.
- [26] Boardman, J., Sauser, B., “Systems of Systems – the meaning of of,” 2006 IEEE/SMC International Conference on System of Systems Engineering, pp. 6, 24-26 April 2006.
- [27] Shenhar, A., “A New Systems Engineering Taxonomy,” 4th International Symposium of the National Council on System Engineering, 1994.
- [28] BAE Systems, “Radar Upgrade Service,” [online] <http://www.baesystems.com/en-uk/product/radar-upgrade-service> [accessed August 2017].
- [29] US Department of Homeland Security, 2003, “The National Strategy For The Physical Protection of Critical Infrastructures and Key Assets,” [online] http://www.dhs.gov/xlibrary/assets/Physical_Strategy.pdf [accessed December 2013].
- [30] Department of Homeland Security, Industrial Control Systems Cyber Emergency Response Team, “Overview of Cyber Vulnerabilities,” [online] <http://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities> [accessed December 2013].
- [31] DeLaurentis, D.A., “A taxonomy-based perspective for systems of systems design methods,” 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 86-91, 10-12 October 2005.
- [32] U.S. Government Publishing Office, Public Law 107-56-Oct. 26, 2001, Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism (U.S.A. Patriot) Act of 2001. P.L. 107-56, [online] <https://www.gpo.gov/fdsys/pkg/PLAW-107publ56/pdf/PLAW-107publ56.pdf> [accessed March 2017].
- [33] The Intelligence and Security Committee of Parliament, June 2013, “Foreign involvement in the Critical National Infrastructure: The implications for national security,” [online] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/205680/ISC-Report-Foreign-Investment-in-the-Critical-National-Infrastructure.pdf [accessed March 2017].
- [34] Foster Jr., J.S., Gjelde, E., Graham, W.R., Hermann, R.J., Kluepfel, H.M., Lawson, R.L., Soper, G.K., Wood, L.L., Woodard, J.B., “Report of the commission to assess the threat to the United States from electromagnetic pulse (EMP) attack,” vol. 1, Executive Report 2004.
- [35] Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T.A., Scholl, H.J., “Understanding Smart Cities: An Integrative Framework,” 2012 45th Hawaii International Conference on System Sciences, Maui, HI, pp. 2289-2297, 2012.
- [36] Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., Williams, P., “Foundations for Smarter Cities,” in IBM Journal of Research and Development, vol. 54, no. 4, 2010.

- [37] Hall, R.E., "The vision of a smart city," in Proceedings of the 2nd International Life Extension Technology Workshop, Paris, France, 28 September, 2000.
- [38] Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M., "Internet of Things for Smart Cities," IEEE Internet of Things Journal, vol. 1, no. 1, February 2014.
- [39] Khali, N., Abid, M.R., Benhaddou, D., Gerndt, M., "Wireless Sensors Networks for Internet of Things," 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, pp. 1-6, 2014.
- [40] Siu, F.W., Brodwin, E.R., "Human Body Systems," Medical, Psychosocial and Vocational Aspects of Disability, Third Edition 2009, pp. 17-38, 2009.
- [41] Karcianas, N., Hessami, A.G., "System of Systems and Emergence Part 1: Principles and Framework," 2011 4th International Conference on Emerging Trends in Engineering & Technology, pp. 27-32, 18-20 November 2011.
- [42] Naqvi, A., Chitchyan, R., Zschaler, S., Rashid, A., Südholt, M., "Cross-Document Dependency Analysis for System-of-System Integration," C Choppy & O Sokolsky (eds), in: Foundations of Computer Software, Future Trends and Techniques for Development: 15th Monterey Workshop 2008, Budapest, Hungary, September 24-26, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6028, Springer, Berlin, pp. 201-226, Monterey Workshop 2008, 1 January.
- [43] National Aeronautics and Space Administration, "Mars Climate Orbiter Mishap Investigation Board," Phase I Report, 10 November 1999.
- [44] House of Commons Transport Committee, "The opening of Heathrow Terminal 5," Twelfth Report of Session 2007-2008, 22 October 2008.
- [45] Barrett, T.J., "Deepwater methods to address system of system risks," 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 92- 96, 10-12 October 2005.
- [46] Chacko, S., 2012, "Coast Guard acquisition chief: Deepwater dead," [online] <http://www.navytimes.com/news/2012/01/coast-guard-deepwater-dead-says-acquisition-chief-010512w/> [accessed 1st April, 2013].
- [47] Yang, K.W., Chen, Y.W., Lu, Y.J., Zhao, Q.S., "The study of guided emergent behavior in system of systems requirement analysis," 2010 5th International Conference on System of Systems Engineering (SoSE), pp. 1-5, 2010.
- [48] Wilkes, D., 2013 "Chaos at cash machines: Computer fault halts RBS withdrawals... While the boss picks up £700,000 bonus," [online] <http://www.dailymail.co.uk/news/article-2289358/Computer-fault-halts-RBS-withdrawals--While-boss-picks-700k-bonus.html>, [accessed 1st April, 2013].
- [49] National Transport Safety Board, "Air Florida, Inc., Boeing 737-222, N62AF, Collision with 14th Street Bridge, Near Washington National Airport, Washington, D.C., January 13, 1982," [online]

- https://www.faa.gov/about/initiatives/maintenance_hf/library/documents/media/aviation_maintenance/airflorida_inc.pdf [accessed August 2017].
- [50] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, December 2009, [online] <https://cloudsecurityalliance.org/csaguide.pdf> [accessed 26th April 2013].
- [51] Kumar, P.A., Selvakumar, S., “Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms,” 2009 IEEE International Advance Computing Conference IACC 2009, pp. 1275-1280, 6-7 March 2009.
- [52] Dantu, R., Loper, K., Kolan, P., “Risk management using behavior based attack graphs,” International Conference on Information Technology: Coding and Computing, ITCC 2004, vol. 1, pp. 445-449, 2004.
- [53] FireEye, 2013, “Next-Generation Threats,” [online] <http://www.fireeye.com/threat-protection/> [accessed 3rd April, 2013].
- [54] Yu, Y., Fry, M., Schaeffer-Filho, A., Smith, P., Hutchison, D., “An adaptive approach to network resilience: Evolving challenge detection and mitigation,” 2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN), pp. 172-179, 2011.
- [55] Verizon, 2013, “2013 Data Breach Investigations Report,” [online] <http://www.verizonenterprise.com/DBIR/2013/> [accessed 23rd April 2013].
- [56] Abdallah, A., Feron, E.M., Hellestrand, G., Koopman, P., Wolf, M., “Hardware/Software Codesign of Aerospace and Automotive Systems,” Proceedings of the IEEE, vol. 98, no. 4, pp. 584-602, April 2010.
- [57] Kumar, A., Lee, B.G., Lee, H., Kumari, A., “Secure storage and access of data in cloud computing,” 2012 International Conference on ICT Convergence (ICTC), pp. 336-339, 2012.
- [58] Cass, S., “Anatomy of malice [computer viruses],” in IEEE Spectrum, vol. 38, no. 11, pp. 56-60, November 2001.
- [59] Cert.org, “Melissa_FAQ,” [online] http://www.cert.org/historical/tech_tips/Melissa_FAQ.cfm? [accessed September 2017].
- [60] Provos, N., McClain, J., Wang, K., “Search Worms,” in Proceedings of the 4th ACM workshop on Recurring malcode (WORM '06), November 2006.
- [61] Kushner, D., “The real story of Stuxnet,” IEEE Spectrum, vol. 50, no. 3, pp. 48-53, March 2013.
- [62] Mohurle, S., Patil, M., "A brief study of Wannacry Threat: Ransomware Attack 2017," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, May - June 2017.

- [63] Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K., "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems*, vol. 21, no. 6, pp. 11-25, December 2001.
- [64] Rinaldi, S.M., "Modeling and Simulating Critical Infrastructures and Their Interdependencies," *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 5-8 January 2004.
- [65] Lee, E.E., Mitchell, J.E, Wallace, W.A., "Restoration of Services in Interdependent Infrastructure Systems: A Network Flows Approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1303-1317, 2007.
- [66] Engineering the Future, The Royal Academy of Engineering, "Infrastructure Interdependencies Timelines Report," May 2013, [online] http://www.engineeringthefuture.co.uk/government/pdf/EtF_Infrastructure_Interdependencies_Report.pdf [accessed 9th December 2013].
- [67] IEEE Standard Glossary of Software Engineering Terminology, IEEE 729-1983, IEEE Computer Society, 1983.
- [68] IEEE Standard Systems and Software Engineering – Vocabulary, IEEE 610.12, ISO/IEC/IEEE 24765:2010(E), IEEE Computer Society, 15 December 2010.
- [69] Kopetz, H., "System-of-Systems Complexity," In *Proceedings AiSoS 2013*, arXiv:1311.3195, EPTCS 133, 2013, pp. 35-39, November 2013.
- [70] Asprou, M., Hadjiantonis, A.M., Ciornei, I., Milis, G., Kyriakides, E., "On the complexities of interdependent infrastructures for wide area monitoring systems," *Complexity in Engineering (COMPENG)*, pp. 1-6, 2012.
- [71] Jovel, J., Jain, R., "Impact of Identified Causal Factors to "System of Systems" Integration Complexity from a Defense Industry Perspective," *Global Journal of Flexible Systems Management* 2009, vol. 10, no. 4, pp. 45-54, 2009.
- [72] Jain, R., Chandrasekaran, A., Elias, G., Cloutier, R., "Exploring the Impact of Systems Architecture and Systems Requirements on Systems Integration Complexity," *IEEE Systems Journal*, vol. 2, no. 2, pp. 209-223, June 2008.
- [73] Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., McDermid, J., Paige, R., "Large-Scale Complex IT Systems," *Communication of the ACM*, vol. 55, no. 7, pp. 71-77, July 2012.
- [74] Oskin, B., 2013, "Japan Earthquake & Tsunami of 2011: Facts and Information," [online] <http://www.livescience.com/39110-japan-2011-earthquake-tsunami-facts.html> [accessed January 2014].
- [75] Ng, T., Jing, L., 2013 "China 'shocked' by water leak at Fukushima nuclear plant," [online] <http://www.scmp.com/news/china/article/1298508/china-shocked-water-leak-fukushima->

- nuclear-plant [accessed January 2014].
- [76] Awodele, O., Onuri, E.E., Okolie, S.O., “Vulnerabilities in Network Infrastructures and Prevention/Containment Measures,” Proceedings of Information Science & IT Education Conference (InSITE) 2012, pp. 53-67, 22-27 June 2012.
- [77] Stamp, J., Dillinger, J., Young, W., DePoy, J., “Common Vulnerabilities in Critical Infrastructure Control Systems,” Sandia National Laboratories, 22 May 2003 (2nd edition, revised 11 November 2003), [online] <http://energy.sandia.gov/wp/wp-content/gallery/uploads/031172C.pdf> [accessed 18th November 2013].
- [78] Kumar, R., Arun, P., Selvakumar, S., “Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment – A survey on DDOS Attack Tools and Traceback Mechanisms,” 2009 IEEE International Advance Computing Conference (IACC 2009), pp. 1275- 1280,, 6-7 March 2009.
- [79] Wu, Z., Ou, Y., Liu, Y., “A Taxonomy of Network and Computer Attacks Based on Responses,” 2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), pp. 26-29, 25-25 September 2011.
- [80] International Standards Organisation, ISO 31000:2009, Risk Management – Principles and Guidelines, 2009.
- [81] Purdy, G., “ISO 31000:2009 – Setting a New Standard for Risk Management,” Risk Analysis, vol. 30, no. 6, pp. 881-886, June 2010.
- [82] Lever, K.E., Kifayat, K., Merabti, M., “Identifying interdependencies using attack graph generation methods,” 2015 11th International Conference on Innovations in Information Technology (IIT), Dubai, pp. 80-85, 1-3 Nov. 2015.
- [83] Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J.M., “Automated Generation and Analysis of Attack Graphs,” 2002 IEEE Symposium on Security and Privacy, pp. 273-284, 2002.
- [84] NIST, 2012, National Institute of Standards and Technology Special Publication 800-30 Revision 1, Guide for Conducting Risk Assessments, September [online] http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf, [accessed March 2015].
- [85] Risksafe.co.uk, 2015, Welcome to RiskSafe Assessment [online] Available at <http://www.risksafe.co.uk/> [accessed May 2015].
- [86] Handbook, 1996, Handleiding Afhankelijkheids – en Kwetsbaarheidsanalyse: stappenplan voor de uitvoering van een A&K-analyse, Version 1.01, Ministry of Internal Affairs, The Hague, The Netherlands (in Dutch).
- [87] CLUSIF, 2015, MEHARI Methods [online] Available at http://www.clusif.asso.fr/en/clusif/present/#mehari_link [accessed March 2015].

- [88] Kumar, R.K.P., Selvakumar, S, “Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment – A survey on DDOS Attack Tools and Traceback Mechanisms,” 2009 IEEE International Advance Computing Conference (IACC 2009), pp. 1275- 1280,, 2009.
- [89] Papp, D., Ma, Z., Buttyan, L., “Embedded systems security: Threats, vulnerabilities, and attack taxonomy,” 2015 13th Annual Conference on Privacy, Security and Trust (PST), 2015.
- [90] Myerson, J.M., “Identifying enterprise network vulnerabilities,” International Journal of Network Management, vol. 12, no.3 pp.135-144, 2002.
- [91] acunetix, “cross-site Scripting (XSS) Attack,” [online] <https://www.acunetix.com/websitesecurity/cross-site-scripting/> [accessed September 2017].
- [92] Nvd.nist.gov, 2017, NVD - Home [online] <https://nvd.nist.gov/> [accessed March 2017].
- [93] Tenable Network Security, 2017, Nessus Vulnerability Scanner [online] <http://www.tenable.com/products/nessus-vulnerability-scanner> [accessed March 2017].
- [94] Beyondtrust.com, 2017, Network Security Scanner | Vulnerability Scanner & Scanning [online] <https://www.beyondtrust.com/products/retina/> [accessed March 2017].
- [95] NMAP.org, 2017, Nmap: the Network Mapper – Free Security Scanner [online] <http://nmap.org/> [accessed March 2017].
- [96] Ptsecurity.com, 2017, MaxPatrol [online] <http://www.ptsecurity.com/products/maxpatrol/> [accessed March 2017].
- [97] Rapid7, 2017, Vulnerability Management & Risk Management Software | Rapid7 [online] <https://www.rapid7.com/products/nexpose/> [accessed May 2017].
- [98] OpenVas.org, 2017, OpenVAS – Open Vulnerability Assessment System [online] <http://www.openvas.org/> [accessed March 2017].
- [99] Saintcorporation.com, 2017, SAINT Cybersecurity solutions: vulnerability assessment, penetration testing [online] Available at <http://www.saintcorporation.com/index.html> [accessed March 2017].
- [100] Pentest-Tools.com [online] <https://pentest-tools.com/home> [accessed September 2017].
- [101] Qualys [online] <https://www.qualys.com/> [accessed September 2017].
- [102] Mell, P., Scarfone, K., Romanosky, S., 2007, “CVSS A Complete Guide to the Common Vulnerability Scoring System Version 2.0,” [online] <https://www.first.org/cvss/cvss-v2-guide.pdf> [accessed March 2017].
- [103] CVE.mitre.org, 2017, CVE - Common Vulnerabilities and Exposures (CVE), [online] Available at <https://cve.mitre.org/> [accessed May 2017].

- [104] Securityfocus.com, 2017, SecurityFocus, [online] <http://www.securityfocus.com/> [accessed October 2017].
- [105] OSVDB.org, 2017, Open Source Vulnerability Database, [online] Available at <http://osvdb.org/> [accessed March 2017].
- [106] Alhomidi, M., Reed, M., “Risk Assessment and Analysis Through Population-Based Attack Graph Modeling,” World Congress on Internet Security (WorldCIS-2013), pp. 19-24, 2013.
- [107] Ou, X., Govindadajhala, S., Appel, A.W., “MulVAL: A Logical-based Network Security Analyzer,” In Proceedings of the 14th Conference on USENIX Security Symposium, 2005.
- [108] Ingols, K., Chu, M., Lippman, R., Webster, S., Boyer, S., “Modeling Modern Network Attacks and Countermeasures Using Attack Graphs,” 2009 Annual Computer Security Applications Conference (ACSAC '09), pp. 117-126, 2009.
- [109] Chu, M., Ingols, K., Lippmann, R., Webster, S., Boyer, S., “Visualizing Attack Graphs, Reachability, and Trust Relationships with NAVIGATOR,” Proceedings of the Seventh International Symposium on Visualization for Cyber Security, pp. 22-33, 2010.
- [110] Jajodia, S., Noel, S., O’Berry, B., “Topological Analysis of Network Attack Vulnerability,” In Managing Cyber Threats, Springer US, pp. 247-266, 2005.
- [111] Jajodia, S., Noel, S., “Topological Vulnerability Analysis,” In Cyber Situational Awareness, Advances in Information Security, vol. 46, pp. 139-154, 2010.
- [112] Mehra, P., “A brief study and comparison of snort and bro open source network intrusion detection systems,” International Journal of Advanced Research in Computing and Communication Engineering, Vol. 1, No. 6, pp.383–386, August 2012.
- [113] Snort.org, 2017, [online] <https://www.snort.org/> [accessed March 2017].
- [114] Bro.org, 2017, The Bro Network Security Monitor, [online] <https://www.bro.org/> [accessed March 2017].
- [115] Tcpcap.org, 2017, TCPDUMP/LIBPCAP public repository, [online] <http://www.tcpcap.org/> [accessed March 2017].
- [116] Roesch, M., “SNORT-Lightweight Intrusion Detection for Networks,” Proceedings of LISA '99: 13th Systems Administration Conference, Seattle, Washington, USA, 1999.
- [117] M-Ice, 2017, “Modular Intrusion Detection and Countermeasure Environment,” Admin Guide, Version 4, [online] Available at <http://m-ice.sourceforge.net/docs/admin-guide.pdf> [accessed May 2017].
- [118] Shoki.sourceforge.net, 2017, Shoki [online] <http://shoki.sourceforge.net/> [accessed March 2017].

- [119] Tzur-David, S., Avissar, H., Dolev., D., Anker, T., "SPADE: Statistical Packet Acceptance Defense Engine," 2010 International Conference on High Performance Switching and Routing, Richardson, TX, 2010, pp. 119-126, 2010.
- [120] Firestorm NIDS, [online] <http://www.scaramanga.co.uk/firestorm/> [accessed March 2017].
- [121] Aroua, M.K., Zouari, B., "A Distributed and Coordinated Massive DDOS Attack Detection and Response Approach," 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 230-235, 2012.
- [122] Singh, K., Dhindsa, K.S., Bhushan, B., "Distributed Defense: An Edge over Centralized Defense against DDos Attacks," International Journal of Computer Network and Information Security, vol. 9, no. 3, pp.36-44, March 2017.
- [123] Naseer, J., Iyengar, N.C.H.S.N., Kumar, M., "Agent Based Detection Mechanism to Outwit Distributed Denial of Service Attacks in Cloud Computing Environment," International journal of Software Engineering and its Applications, vol. 10, no. 10, pp. 149-164, 2016.
- [124] Houssaini, M.A.E., Aaroud, A., Hore, A.E., Ben-Othman, J., "Detection of Jamming Attacks in Mobile Ad Hoc Networks using Statistacal Process Control," 7th International Conference in Ambient Systems, Networks and Technologies (ANT 2016), vol. 83, pp. 26-33, 2016.
- [125] Sharah, A.A, Oyedare, T., Shettu, S., "Detecting and Mitigating Smart Insider Jamming Attacks in MANETs Using Reputation-Based Coalition Game," Journal of Computer Networks and Communicactions, vol. 2016, no. 2, 2016.
- [126] Yalu, N., Banerjee, S., "An efficient packet hiding method for preventing jamming attacks in wireless networks," 2016 international Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, pp. 2318-2321, 2016.
- [127] Zou, Y., Zhu, J., Zheng, B., "Defending against eavesdropping attack leveraging multiple antennas in wireless networks," 2013 8th International Conference on Communications and Networking in China (CHINACOM), Guilin, pp. 699-703, 2013.
- [128] Ma, D., Wang, L., Lei, C., Xu, Z., Zhang, H., Li, M., "Thwart eavesdropping attacks on network communication based on moving target defense," 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC), Las Vegas, NV, pp. 1-2, 2016.
- [129] Li, X., Wang, H., Dai, H.N., Wang, Y., Zhao, Q., "An Analytical Study on Eavesdropping Attacks in Wireless Nets of Things," in Mobile Information Systems, vol. 2016, January 2016.
- [130] Gander, M., Sauerwein, C., Brey, R., "Tracing masquerading attacks in distributed healthcare information systems," 2016 SAI Computing Conference (SAI), London, pp. 1107-1117, 2016.
- [131] Kholidy, H.A., Baiardi, F., Hariri, S., "DDSGA: A Data-Driven Semi-Global Alignment Approach for Detecting Masquerade Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 2, pp. 164-178, March-April 2015.

- [132] Pratik, P.J., Madhu, B.R., “Data mining based CIDS: Cloud intrusion detection system for masquerade attacks [DCIDSM],” 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, pp.1-5, 2013.
- [133] Marques, D., Muslukhov, I., Guerreiro, T., Beznosov, K., Carriço, L., “Snooping on Mobile Phones: Prevalence and Trends,” in Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), Denver, CO, USA, 2016.
- [134] Gao, M., Zhu, X., Su, Y., “Protecting router cache privacy in named data networking,” 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, pp. 1-5, 2015.
- [135] Chung, C., Khatkar, P., Xing, T., Lee, J., Huang, D., “NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems,” IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 198-211, 2013.
- [136] Sheikhan, M., Bostani, H., “A hybrid intrusion detection architecture for Internet of things,” 2016 8th International Symposium on Telecommunications (ISST), Tehran, Iran, pp. 601-606, 2016.
- [137] Gai, K., Qiu, M., Tao, L., Zhu, Y., “Intrusion detection techniques for mobile cloud computing in heterogeneous 5G,” Security and Communication Networks, vol. 9, iss. 16, pp. 3049-3058, November 2016.
- [138] Zhao, X., Zhang, W., “Hybrid Intrusion Detection Method Based on Improved Bisecting K-means in Cloud Computing,” 2016 13th Web Information Systems and Applications Conference (WISA), Wuhan, China, pp. 225-230, 2016.
- [139] Xing, T., Huang, D., Xu, L., Chung, C.J., Khatkar, P., “SnortFlow: A OpenFlow-based Intrusion Prevention System in Cloud Environment,” 2013 Second GENI Research and Educational Experiment Workshop, Salt Lake City, UT, pp. 89-92, 2013.
- [140] Rodas, O., To, M.A., Alvarez, J., Maag, S., “Protecting Wireless Mesh Networks through a distributed intrusion prevention framework,” 2015 7th IEEE Latin-American Conference on Communications (LATINCOM), Arequipa, pp. 1-6, 2015.
- [141] Sanjab, A., Saad, W., “On bounded rationality in cyber-physical systems security: Game-theoretic analysis with application to smart grid protection,” 2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Vienna, pp. 1-6, 2016.
- [142] Ruiz, J.F., Rudolph, C., Maña, A., Arjona, M., “A security engineering process for systems of systems using security patterns,” 2014 8th Annual IEEE International Systems Conference (SysCon), Ottawa, ON, pp. 8-11, 2014.
- [143] Mori, M., Ceccarelli, A., Lollini, P., Bondavalli, A., Frömel, B., “A Holistic Viewpoint-Based SysML Profile to Design Systems-of-Systems,” 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), Orlando, FL, pp. 276-283, 2016.

- [144] Trivellato, D., Zannone, N., Etalle, S., "A Security Framework for Systems of Systems," 2011 IEEE International Symposium on Policies for Distributed Systems and Networks, Pisa, pp. 182-183, 2011.
- [145] Brunner, M., Huber, M., Sauerwein, C., Breu, R., "Towards an Integrated model for Safety and Security Requirements of Cyber-Physical Systems," 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, pp. 334-340, 2017.
- [146] Salah, K., Calyam, P., Boutaba, R., "Analytical Model for Elastic Scaling of Cloud-Based Firewalls," in IEEE Transactions on Network and Service Management, vol. 14, no. 1, pp. 136-146, March 2017.
- [147] Chomsiri, T., He, X., Nanda, P., Tan, Z., "An Improvement of Tree-Rule Firewall for a Large Network: Supporting Large Rule Size and Low Delay," 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, pp. 178-184, 2016.
- [148] Cheminod, M., Durante, L., Valenzano, A., Zunino, C., "Performance impact of commercial industrial firewalls on networked control systems," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, pp. 1-8, 2016.
- [149] Cohen, T., Hendler, D., Poashnik, D., "Supervised Detection of Infected Machines Using Anti-virus Induced Labels," International Conference on Cyber Security Cryptography and Machine Learning (CSCML), vol. 10332, pp. 34-49, June 2017.
- [150] Dev, M., Gupta, H., Mehta, S., Balamurugan, B., "Cache Implementation using Collective Intelligence on Cloud Based Antivirus Architecture," 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 593-595, 2016.
- [151] Sheta, M.A., Zaki, M., Hadad, K.A.E.S.E., Aboelseoud M, H., "Anti-spyware Security Design Patterns," 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, pp. 465-470, 2016.
- [152] Nunez, Y., Gustavson, F., Grossman, F., Tappert, C., "Designing a distributed patch management security system," 2010 International Conference on Information Society, London, pp. 162-167, 2010.
- [153] Kim, J.H., Sohn, M.S., Won, Y.J., "An Automatic Patch Management System with Improved Security," in Advanced Multimedia and Ubiquitous Engineering, Springer, Singapore, pp. 74-80, 2017.
- [154] Benzid, D., Kadoch, M., "Virtual Private Network over Wireless Mesh Networks," 2014 International Conference on Future Internet of Things and Cloud, Barcelona, pp. 340-345, 2014.
- [155] Bhat, A.Z., Shuaibi, D.K.A., Singh, A.V., "Virtual private network as a service - A need for discrete cloud architecture," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, pp. 526-532,

- 2016.
- [156] Yang, L., Cui, X., Wang, C., Guo, S., Xu, X., “Risk Analysis of Exposed Methods to JavaScript in Hybrid Apps,” 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, pp. 458-464, 2016.
- [157] Peikert, T., Garbe, H., Potthast, S., “Fuzzy based risk analysis for IT-Systems and Their infrastructure,” 2016 IEEE International Symposium on Electromagnetic Compatibility (EMC), Ottawa, ON, pp. 51-56, 2016.
- [158] Liu, X., Shahidehpour, M., Cao, Y., Wu, L., Wei, W., Liu, X., “Microgrid Risk Analysis Considering the Impact of Cyber Attacks on Solar PV and ESS Control Systems,” in IEEE Transactions on Smart Grid, vol. 8, no. 3, May 2017.
- [159] Ketabdar, H., Rezaee, R., GhaemiBafghi, A., Khosravi-Farmad, M., “Network security risk analysis using attacker's behavioral parameters,” 2016 6th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, pp. 325-330, 2016.
- [160] Vegendla, A., Sogaard, T.M., Sindre, G., “Extending HARM to make Test Cases for Penetration Testing,” International Conference on Advanced Information Systems Engineering CAISE 2016: Advanced Information Systems Engineering Workshops, pp. 254-265, 2016.
- [161] Kadam, S.P., Mahajan, B., Patanwala, M., Sanas, P., Vidyarthi, S., “Automated Wi-Fi penetration testing,” 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, pp. 1092-1096, 2016.
- [162] Berger, H., Jones, A., “Cyber Security & Ethical Hacking For SMEs,” in Proceedings of The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society, no. 12, Hagen, Germany, July 2016.
- [163] Guarda, T., Orozco, W., Augusto, M.F., Morillo, G., Navarrete, S.A., Pinto, F.M., “Penetration Testing on Virtual Environments,” In Proceedings of the 4th International Conference on Information and Network Security, pp.9-12, Malaysia, 28-31 December, 2016.
- [164] Ageneau, P.L., Wu, C., Boukhatem, N., Gerla, M., “Redundancy Adaption for Multi-Path Intra-Flow Network Coding in Wireless Mesh Networks,” 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, pp. 1-7, 2016.
- [165] Goswami, P., Chitnis, K., Jadav, B., Kapania, A., Sivasankaran, S., “Software framework for runtime application monitoring of fail-safe multi-processor,” 2017 IEEE international Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, pp. 39-42, 2017.
- [166] Savas, S.S., Ma, C., Tornatore, M., Mukherjee, B., “Backup reprovisioning with partial protection for disaster-survivable software-defined optical networks,” Photonic Network Communications, vol. 31, iss. 2, pp. 186-195, April 2016.
- [167] Fiala, D., Mueller, F., Ferreira, K.B., “FlipSphere: A Software-Based DRAM Error Detection and Correction Library for HPC,” 2016 IEEE/ACM 20th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), London, United Kingdom, pp.

- 19-28, 2016.
- [168] Aydos, G., Fey, G., “Empirical results on parity-based soft error detection with software-based retry,” *Microprocessors and Microsystems*, vol. 48, pp. 62-68, February 2017.
- [169] Borchert, C., Schirmeir, H., Spinczyk, O., “Generic Soft-Error Detection and Correction for Concurrent Data Structures,” in *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 22-36, January-February 2017.
- [170] Ulbrich, P., Hoffmann, M., Kapitzka, R., Lohmann, D., Schroder-Preikschat, W., Schmid, R., “Eliminating Single Points of Failure in Software-Based Redundancy,” 2012 Ninth European Dependable Computing Conference (EDCC), pp. 49-60, 2012.
- [171] Wang, Y., Li, X., “Achieve high availability about point-single failures in OpenStack,” 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, pp. 45-48, 2015.
- [172] Lin, C.H., Shieh, C.K., Hwang, W.S., Wang, J.J., “R-SPOFTR: Relieving single point of failure in Tree Routing to prolong sensor system lifetime,” 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, pp. 283-288, 2015.
- [173] Kumberg, T., Schink, M., Reindl, L.M., Schindelbauer, C., “T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers,” *Ad Hoc Networks*, vol. 59, pp. 97-115, May 2017.
- [174] Yadav, A.K., Das, S.K., Tripathi, S., “EFMMRP: Design of efficient fuzzy based multi-constraint multicast routing protocol for wireless ad-hoc network,” *Computer Networks*, vol. 118, pp. 15-23, 8 May 2017.
- [175] Pham, T.M.T., Nguyen, T.T., Kim, D.S., “Geographical awareness hybrid routing protocol in Mobile Ad Hoc Networks,” *Wireless Networks*, vol. 23, iss. 1, pp. 1-13, January 2017.
- [176] Rahem, A.A.T., Ismail, M., Najm, I.A., “Topology sense and graph-based TSG: efficient wireless ad hoc routing protocol for WANET,” *Telecommunication Systems*, vol. 65, iss. 4, pp. 739-754, August 2017.
- [177] Cadini, F., Agliardi, G.L., Zio, E., “A modeling and simulation framework for the reliability/availability assessment of a power transmission grid subject to cascading failures under extreme weather conditions,” *Applied Energy*, vol. 185, part 1, pp. 267-279, 1 January 2017.
- [178] Zhang, D.X., Zhao, D., Guan, Z.H., Wu, Y., Chi, M., Zheng, G.L., “Probabilistic analysis of cascade failure dynamics in complex network,” *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 299-309, November 2016.
- [179] Feng, Y., Sun, B., Zeng, A., “Cascade of links in complex networks,” *Physics Letters A*, vol. 381, iss. 4, pp. 263-269, 30 January 2017.

- [180] Wang, J., Sun, E., Xu, B., Li, P., Ni, C., “Abnormal cascading failure spreading on complex networks,” *Chaos, Solitons & Fractals*, vol. 91, pp. 695-701, October 2016.
- [181] Brummitt, C.D., D'Souza, R.M., Leicht, E.A., “Suppressing cascades of load in interdependent networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 12, 21 February 2012.
- [182] Cai, Y., Cao, Y., Li, Y., Huang, T., Zhou, B., “Cascading Failure Analysis Considering Interaction Between Power Grids and Communications Networks,” in *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 530-538, January 2016.
- [183] Xue, F., Bompard, E., Huang, T., Jiang, L., Lu, S., “Interrelation of structure and operational states in cascading failure of overloading lines in power grids,” *Physica A: Statistical Mechanics and its Applications*, vol. 482, pp. 728-740, 15 September 2017.
- [184] Zhu, Y., Yan, J., Sun, Y., He, H., “Revealing Cascading Failure Vulnerability in Power Grids Using Risk-Graph,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3274-3284, December 2014.
- [185] Heracleous, C., “Micropolis Interdependency Modeling using Open Hybrid Automata,” *Systems and Control (cs.SY)*, arXiv:1609.09395v1 [cs.SY], September 2016.
- [186] Zhu, W., Milanović, J.V., “Interdependency Modeling of Cyber-physical Systems Using a Weighted Complex Network Approach,” 2017 IEEE Manchester powerTech, pp. 1-6, 2017.
- [187] Tøndel, I.A., Foros, J., Kilskar, S.S., Hokstad, P., Jaatun, M.G., “Interdependencies and reliability in the combined ICT and power systems: An overview of current research,” *Applied Computing and Informatics*, 2017.
- [188] Heracleous, C., Kolios, P., Panayiotou, C.G., Ellinas, G., Polycarpou, M.M., “Hybrid systems modeling for critical infrastructures interdependency analysis,” *Reliability Engineering and System Safety*, vol. 165, pp. 89-101, September 2017.
- [189] Mane, M., DeLaurentis, D., Frazho, A., “A Markov Perspective on Systems-of-Systems Complexity,” 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 1238-1243, 2011.
- [190] Liu, Z., Li, S., He, J., Xie, D., Deng, Z., “Complex Network Security Analysis based on Attack Graph Model,” 2012 Second International Conference on Instrumentation & Measurement, Computer, Communication and Control, pp. 183-186, 2012.
- [191] Meyerhenke, H., Sanders, P., Schulz, C., “Parallel Graph Partitioning for Complex Networks,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 9, pp. 2625-2638, September 2017.
- [192] Liao, H., Marani, M. S., Medo, M., Zhang, Y. C., Zhou, M. Y., “Ranking in evolving complex networks,” *Physics Reports*, vol. 689, pp. 1-54, 19 May 2017.

- [193] Li, W., Jia, Y., Du, J., "State estimation for stochastic complex networks with switching topology," in *IEEE Transactions on Automatic Control*, no. 99, 2017.
- [194] O'Toole, E., Nallur, V., Clarke, S., "Decentralised Detection of Emergence in Complex Adaptive Systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 12, iss. 1, no. 4, May 2017.
- [195] Zoppi, T., Ceccarelli, A., Bondavalli, A., "Exploring anomaly detection in systems of systems," *Proceedings of the Symposium on Applied Computing SAC '17*, pp. 1139 - 1146, April 2017.
- [196] Khan, T.A., Wang, J., "On formalization of emergent behaviours in multiagent systems with limited interactions," *2016 IEEE International Conference on Electro Information Technology (EIT)*, pp. 0553-0558, 2016.
- [197] Shi, G., Proutiere, A., Johansson, M., Baras, J.S., Johansson, K.H., "Emergent Behaviours Over Signed Random Dynamical Networks: Relative-State-Flipping Model," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, June 2017.
- [198] Singh, S., Lu, S., Kokar, M.M., Kogut, P.A., Lockheed Martin, "Detection and Classification of Emergent Behaviours using Multi-Agent Simulation Framework (WIP) ," *Proceedings of the Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems*, no. 3, April 2017.
- [199] Ahmad, M.A., Woodhead, S., Gan, D., "Early containment of fast network worm malware," *2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, Danang City, pp. 195-201, 2016.
- [200] US Department of Homeland Security, National Cybersecurity and Communications Integration Center, "Cyber Security Evaluation Tool," [online] https://ics-cert.us-cert.gov/sites/default/files/FactSheets/ICS-CERT_FactSheet_CSET_S508C.pdf [accessed July 2017].
- [201] Yao, J., Venkitasubramaniam, P., Kishore, S., Snyder, L.V., Blum, R.S., "Network topology risk assessment of stealthy cyber attacks on advanced metering infrastructure networks," *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pp. 1-6, 2017.
- [202] Tanimoto, S., Shiraki, S., Iwashita, M., Kobayashi, T., Sato, H., Kanai, A., "Risk Assessment Based on User's Viewpoint for Mobile Ad Hoc Network," *2016 19th International Conference on Network-Based Information Systems (NBIS)*, Ostrava, pp. 280-285, 2016.
- [203] Loutchkina, I., Jain, L.C., Nguyen, T., Nestervo, S., "Systems' Integration Technical Risks' Assessment Model (SITRAM)," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3, pp. 342-352, March 2014.
- [204] Guzman, A., Ishida, S., Choi, E., Aoyama, A., "Artificial intelligence improving safety and risk analysis: A comparative analysis for critical infrastructure," *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 471-475,

- 2016.
- [205] Cheng, Q., Kwait, K., Kamhoua, C.A., Njilla, L., “Attack Graph Based Network Risk Assessment: Exact Inference vs Region-Based Approximation,” 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), pp. 84-87, 2017.
- [206] Wang, L., Islam, T., Long, T., Singhal, A., Jajodia, S., “An Attack Graph-Based Probabilistic Security Metric,” Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pp. 283-296, 2008.
- [207] Feng, C., Jin-Shu, S., “A Flexible Approach to Measuring Network Security Using Attack Graphs,” 2008 International Symposium on Electronic Commerce and Security, pp. 426-431, 2008.
- [208] Kecskemeti, G., Casale, G., Jha, D.N., Lyon, J., Ranjan, R., “Modelling and Simulation Challenges in Internet of Things,” in IEEE Cloud Computing, vol. 4, no. 1. pp. 62-69, January-February 2017.
- [209] Sarigiannidis, P., Karapistoli, E., Economides, A., “Modelling the Internet of Things Under Attack: A G-network Approach,” in IEEE Internet of Things Journal, no. 99, 2017.
- [210] Milanovic, J.V., Zhu, W., “Modelling of Interconnected Critical Infrastructre Systems Using Complex Network Theroy,” in IEEE Transactions on Smart Grid, no. 99, 2017.
- [211] Kaynar, K., Sivrikaya, F., “Distributed Attack Graph Generations,” in IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 5, pp. 519-532, September-October 2016.
- [212] Li, H., Wang, Y., Cao, Y., “Searching Forward Complete Attack Graph Generation Algorithm Based on Hypergraph Partitioning,” Procedia Computer Science, vol. 107, pp. 27-38, 2017.
- [213] Nichols, W., Hawrylak, P., Hale, J., Papa, M., “Introducing priority into hybrid attack graphs,” CISRC '17 Proceedings of the 12th Annual Conference on Cyber and Information Security Research, no. 12, April 2017.
- [214] Chejara, P., Garg, U., Singh, G., “Vulnerability Analysis in Attack Graphs Using Conditional Probability,” International Journal of Soft Computing and Engineering (IJSCE), vol. 3. no. 2, December 2013.
- [215] Polad, H., Puzis, R., Shapira, B., “Attack Graph Obfuscation,” International Conference on Cyber Security Cryptography and Machine Learning, pp. 269-287, 2017.
- [216] Johnson, P., Vernotte, A., Gorton, D., Ekstedt, M., Lagerström, R., “Quantitative information Security Risk Estimation Using Probabilistic Attack Graphs,” Risk Assessment and Risk-Driven Quality Assurance: 4th International Workshop, RISK 2016, Held in Conjunction with ICTSS 2016, Graz, Austria, 18 October, 2016, Revised Selected Papers (Vol. 10224, p. 37). Springer, 2017.
- [217] Sun, X., Singhal, A., Liu, P., “Towards Actionable Mission Impact Assessment in the Context of Cloud Computing,” In IFIP Annual Conference on Data and Applications Security and

- Privacy, pp. 259-274, 2017.
- [218] Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., “Fuzziness based semi-supervised learning approach for intrusion detection system,” *Information Sciences*, vol. 378, pp. 484-497, 1 February 2017.
- [219] Won, J., Singla, A., Bertino, E., “CertificateLess Cryptography-Based Rule Management Protocol for Advanced Mission Delivery Networks,” 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, pp. 7-12, 2017.
- [220] Hong, H., Sun, Z., “Achieving secure data access control and efficient key updating in mobile multimedia sensor networks,” *Multimedia Tools and Applications*, pp. 1-14, May 2017.
- [221] Böse, B., Avasarala, B., Tirthapura, S., Chung, Y.Y., Steiner, D., “Detecting Insider Threats Using RADISH: A System for Real-Time Anomaly Detection in Heterogeneous Data Streams,” in *IEEE Systems Journal*, vol. 11, no. 2, pp. 471-482, June 2017.
- [222] Li, X., Tang, S., Xu, L., Wang, H., Chen, J., “Two-Factor Data Access Control With Efficient Revocation for Multi-Authority Cloud Storage Systems,” in *IEEE Access*, vol. 5, pp. 393-405, 2017.
- [223] Fugkeaw, S., Sato, H., “Scalable and secure access control policy update for outsourced big data,” *Future Generation Computer Systems*, June 2017.
- [224] Sampaio, L., Garcia, A., “Exploring context-sensitive data flow analysis for early vulnerability detection,” *Journal of Systems and Software*, vol. 113, pp. 337-361, March 2016.
- [225] Szabó, T., Alperovich, S., Erdweg, S., Voelter, M., “An extensible framework for variable-precision data-flow analyses in MPS,” 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE), Singapore, 2016, pp. 870-875, 2016.
- [226] Yan, X., Zhang, L., Wu, Y., Luo, Y., Zhang, X., “Secure smart grid communications and information integration based on digital watermarking in wireless sensor networks,” *Enterprise Information Systems*, vol. 11, iss. 2, February 2017.
- [227] Glissa, G., Rachedi, A., Meddeb, A., “A Secure Protocol Based on RPL for Internet of Things,” 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, pp. 1-7, 2016.
- [228] Wang, X., Cheng, H., Yao, Y., “Addressing-Based Routing Optimization for 6LoWPAN WSN in Vehicular Scenario,” in *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3939-3947, 15 May 2016.
- [229] Farooqi, A., H., Khan, F., A., “Securing wireless sensor networks for improve performance in cloud-based environments,” *Annals of Telecommunications*, iss. 5-6/ , pp. 1-18, 2017.
- [230] Kumrai, T., Ota, K., Dong, M., Kishigami, J., Sung, D.K., “Multi-objective Optimization in Cloud Brokering Systems for Connected Internet of Things,” in *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404-413, April 2017.

- [231] Yao, H., Yang, H., Zhang, A., Fang, C., Guo, Y., “WLAN interference self-optimization using som neural networks,” *Concurrency and Computation: Practice and Experience*, vol. 29, iss. 3, 10 February 2017.
- [232] Rullo, A., Midi, D., Serra, E., Bertino, E., “A Game of Things: Strategic Allocation of Security Resources for IoT,” in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pp. 185-190, 18-21 April 2017.
- [233] Zhao, P., Chen, X., Yu, P., Li, W., Qiu, X., Guo, S., “Risk assessment and optimization for key services in smart grid communication network,” *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, pp. 600-603, 2017.
- [234] Yun, D., Wu, C., Zhu, M.M., “Transport-Support Workflow Composition and Optimization for Big Data Movement in High-performance Networks,” in *IEEE Transactions on Parallel and Distributed Systems*, 2017.
- [235] Alfarhan, F., Alsohaily, A., “Self-organizing wireless network parameter optimization through mixed integer programming,” *2017 IEEE International Conference on Communications (ICC)*, Paris, France, 2017.
- [236] Li, Y., Chen, Z., Wang, Y., Jioa, L., Xue, Y., “A Novel Distributed Quantum-Behaved Particle Swarm Optimization,” *Journal of Optimization*, vol. 2017, 2017.
- [237] Zhou, B., Arabo, A., Drew, O., Llewellyn-Jones, D., Merabti, M., Shi, Q., Waller, A., Craddock, R., Jones, G., Arnold, K.L.Y., “Data Flow Security Analysis for System-of-Systems in a Public Security Incident,” in *The 3rd Conference on Advances in Computer Security and Forensics (ACSF 2008)*, Liverpool, UK, 10-11 July 2008.
- [238] Arabo, A., Kifayat, K., Shi, Q., Llewellyn-Jones, D., Merabti, M., “State-of-the-Art in System-of-Systems Security for Crisis Management,” In *Fourth Annual Layered Assurance Workshop (LAW 2010)*, 2010.
- [239] Gharaibeh, A., Salahuddin, M.A., Hussini, S.J., Khreishah, A., Khalil, I., Guizani, M., Al-Fuqaha, A., “Smart Cities: A Survey on Data Management, Security and Enabling Technologies,” in *IEEE Communications Surveys & Tutorials*, 2017.
- [240] Vishwakarma, P.K., “Optimizing and Analysing The Effectiveness of Security Hardening Measures Using Various Optimization Techniques as well as Network Management Models giving Special Emphasis to Attack Tree Model,” *International Journal of Network Security & Its Applications (IJNSA)*, vol. 3, no. 4, July 2011.
- [241] Lai, J.B., Wang, H.Q., Liu, X.W., Liang, Y., Zheng, R.J., Zhao, G.S., “WNN-Based Network Security Situation Quantitative Prediction Method and Its Optimization,” *Journal of Computer Science and Technology*, vol. 23, iss. 2, pp. 222-230, March 2008.
- [242] Saravanan, K., Karthik, S., “Packet Score Based Network Security and Traffic Optimization,” *Networking and Internet Architecture*, arXiv:1202.2024, 2012.

- [243] Zhang, X., Zhan, J., Jiang, W., Ma, Y., "A Vulnerability Optimization Method for Security-Critical Real-Time Systems," 2013 IEEE International Conference on Networking, Architecture and Storage, pp. 215-221, 2013.
- [244] Enaya, Y.A., Deb, K., "Network Path Optimization Under Dynamic Conditions," 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 2977-2984, 2014.
- [245] Panda, M., "Security in Wireless Sensor Networks using Cryptographic Techniques," American Journal of Engineering Research (AJER), vol. 3, no. 1, pp. 50-56, 2014.
- [246] Sanka, S., Hota, C., Rajarajan, M., "Secure Data Access in Cloud Computing," 2010 IEEE 4th International Conference on Internet Multimedia Services Architecture and Application, Bangalore, pp. 1-6, 2010.
- [247] Kim, H., Song, J., "Social network analysis of patent infringement lawsuits," Technological Forecasting and Social Change, vol. 80, iss. 5, pp. 944-955, June 2013.
- [248] Meghanathan, N., "Advanced Methods for Complex Network Analysis," IGI Global, 2016.
- [249] Hwang, W., Cho, Y.R., Zhang, A., Ramanathan, M., "Bridging centrality: Identifying Bridging Nodes in Scale-Free Networks," In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), August 2016.
- [250] Lever, K.E., Kifayat, K., "Risk Assessment and Attack Graph Generation for Collaborative Infrastructures: a Survey," International Journal of Critical Computer-Based Systems, vol. 6, no. 3, 2016.
- [251] FIRST.Org, "CVSS, Common Vulnerability Scoring System v3.0: Specification Document," [online] <https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf> [accessed November 2016].
- [252] Medhi, J., "Statistical methods: An Introductory Text," Second Edition, New Age International, 2013.
- [253] MarineBio, 29 December 2011, "The Deep Sea," [online] <http://marinebio.org/oceans/deep/> [accessed June 2017].
- [254] Monteiro, M.S.R., Fontes, D.B.M.M., Fontes, F.A.C.C., "Concave minimum cost network flow problems solved with a colony of ants," Journal of Heuristics, vol. 19, iss. 1, pp. 1-33, February 2013.
- [255] Grefenstette, J.J., "Optimization of Control Parameters for Genetic Algorithms," IEEE Transactions on Systems, man, and Cybernetics, vol. 16, no. 1, pp. 122-128, January 1986.
- [256] Kaur, G., "A preventive approach to mitigate the effects of gray hole attack using genetic algorithm," 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), pp. 1-8, Dehradun, 2016.
- [257] Blum, C., "Ant colony optimization: Introduction and recent trends," Physics of Life Reviews,

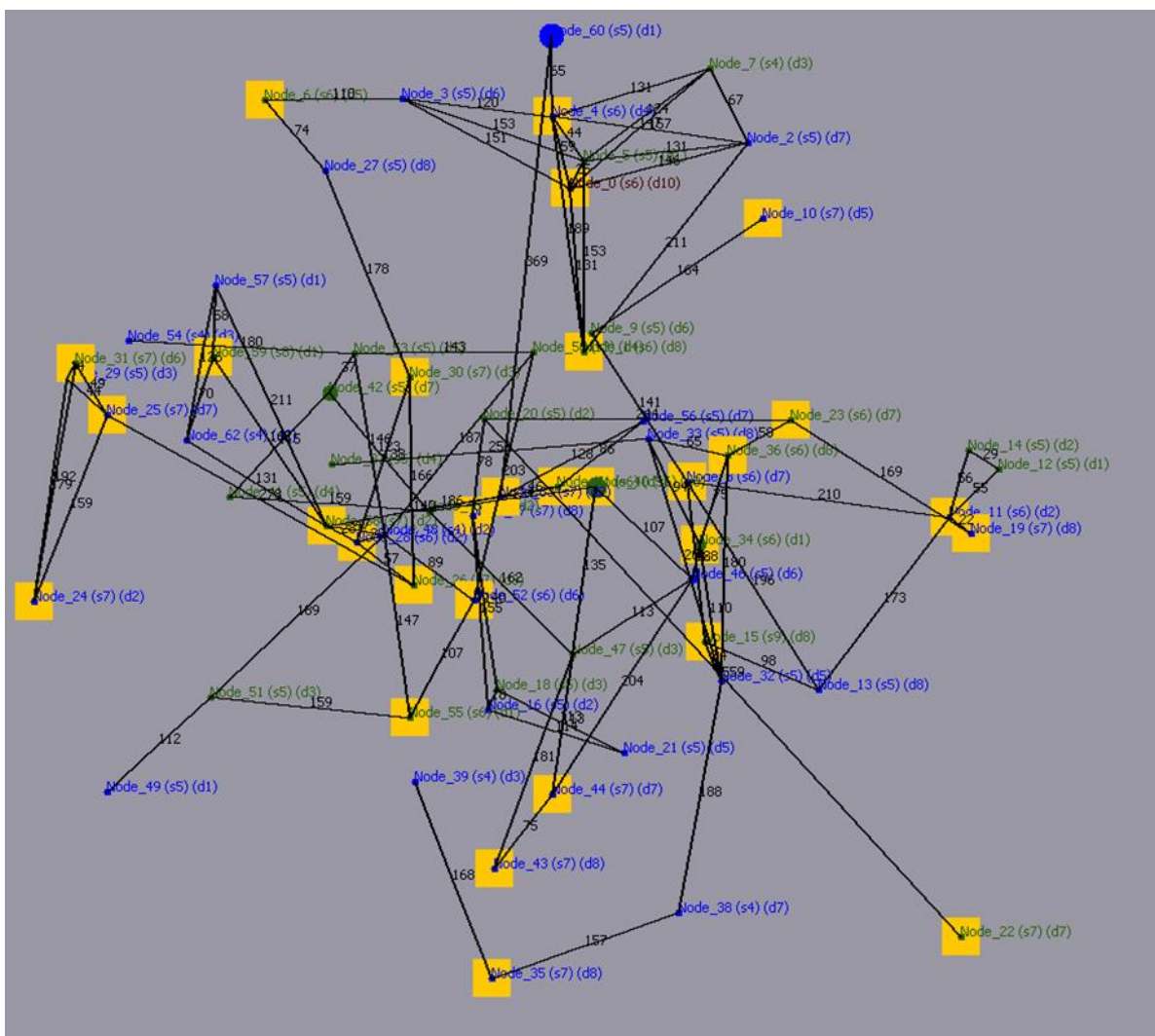
vol. 2, no. 4, pp. 353-373, December 2005.

- [258] Brownlee, J., *Clever Algorithms, Nature-Inspired Programming Recipes*, First Edition, 2011.
- [259] Gao, Q., Holding, D.J., Blow, K.J., “Energy Efficiency Design Challenge in Sensor Networks,” in *Proceedings of London Communications Symposium*, 2002.
- [260] Qui, T., Chen, N., Li, K., Qiao, D., Fu, Z., “Heterogeneous ad hoc networks: Architectures, advances and challenges,” *Ad Hoc Networks*, vol. 55, pp. 143-152, February 2017.

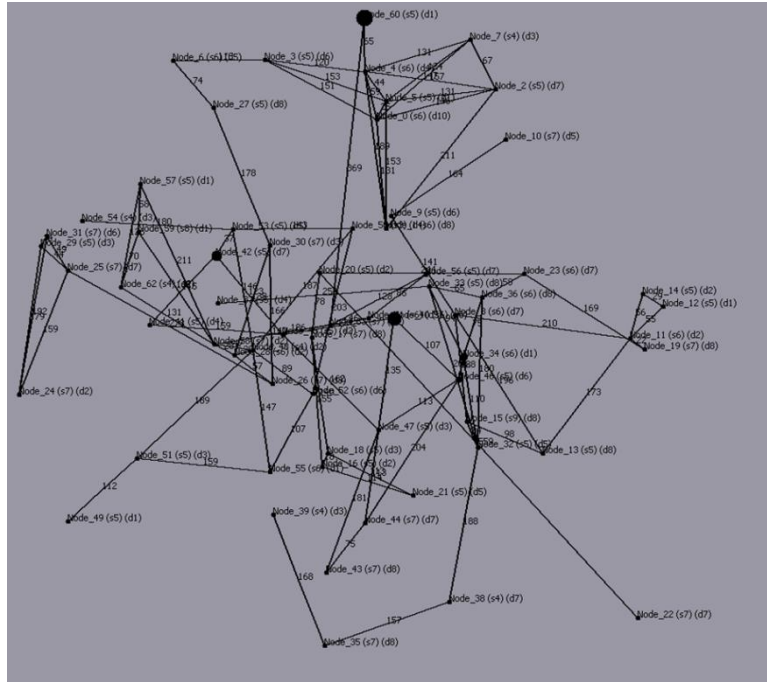
Appendix A

SCRAM Positive Multi-Level SoS Vulnerability Performance

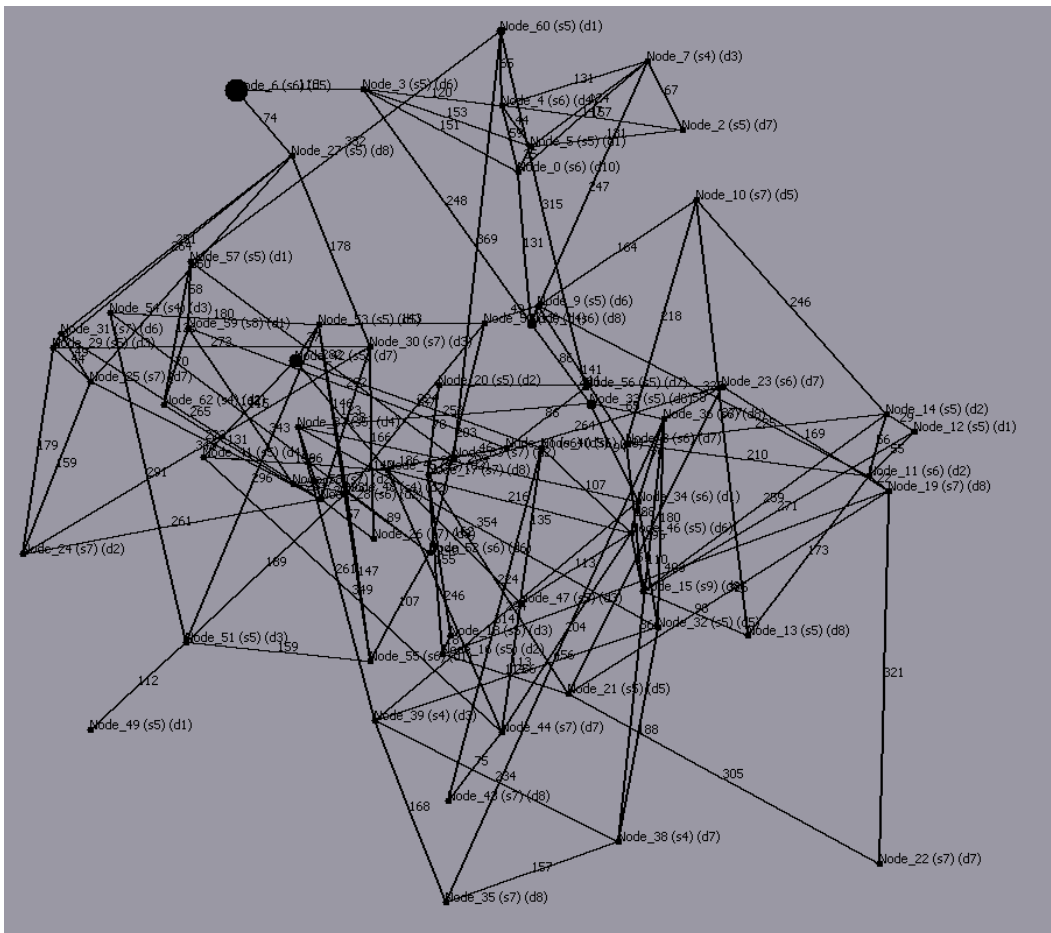
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.11-a, 6.11-b, and 6.11-c.



Appendix A Figure 1. Multi-Level SoS A with Node Status

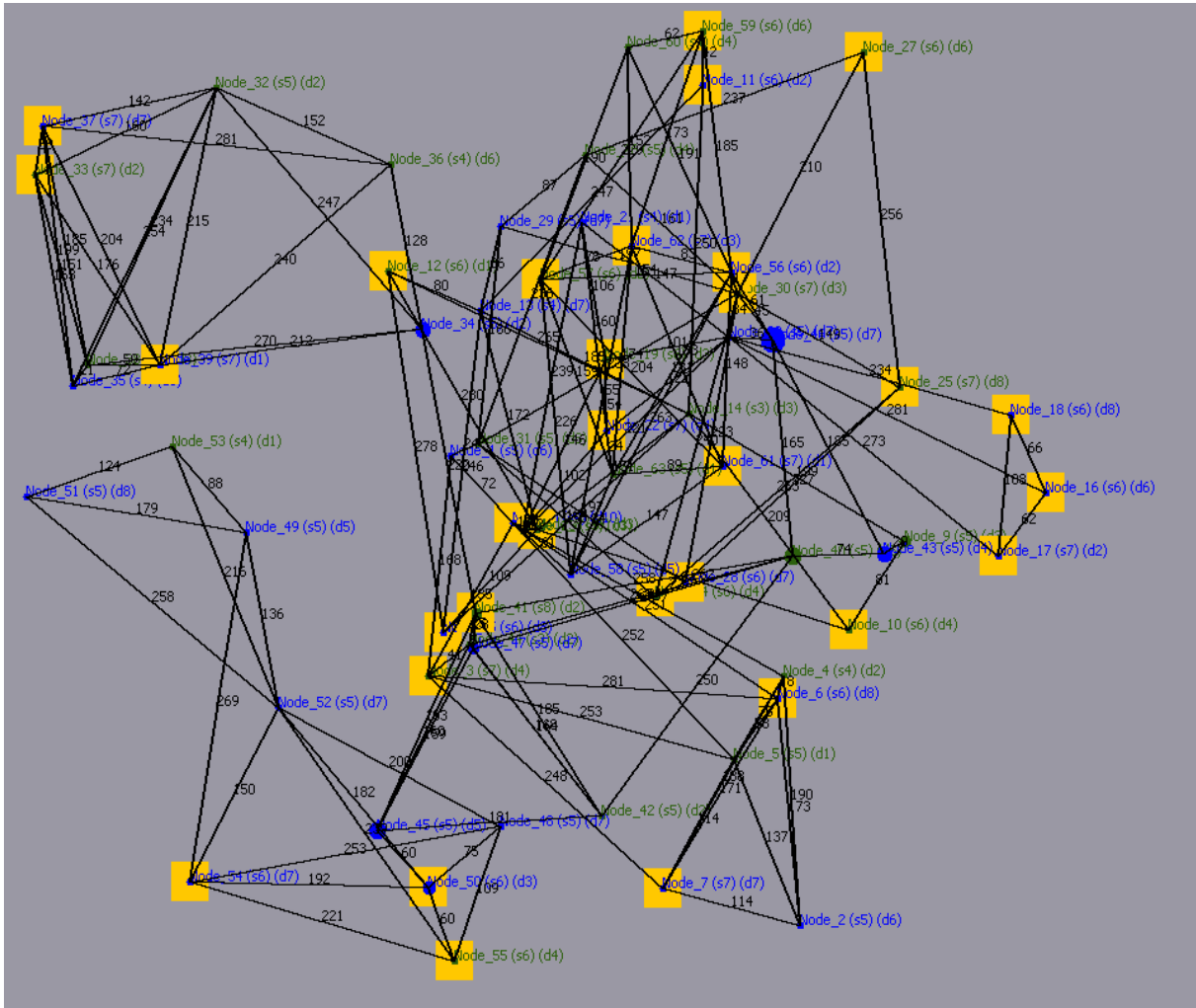


Appendix A Figure 2. Multi-Level SoS A Topology

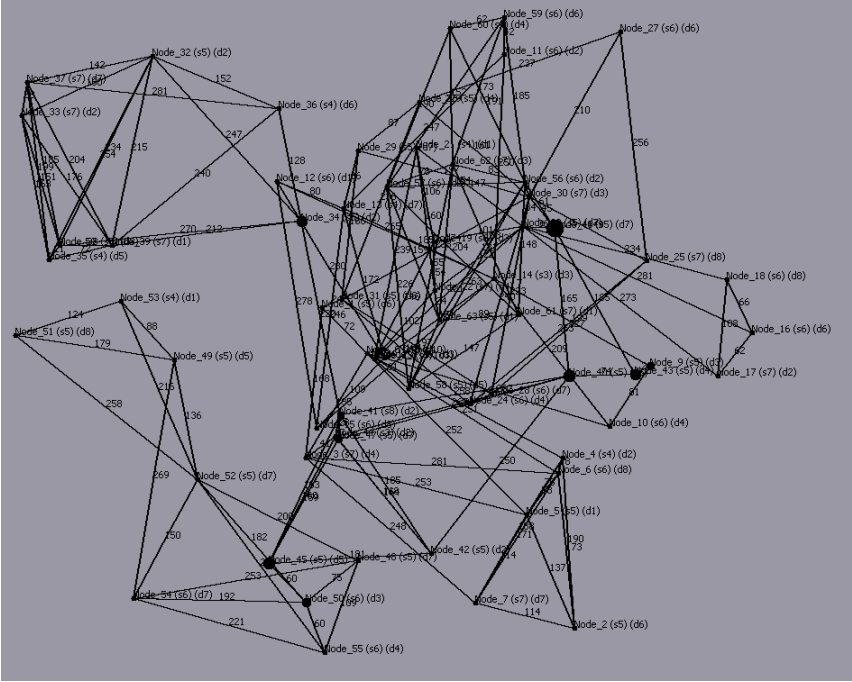


Appendix A Figure 3. Multi-Level SoS A Optimum Candidate

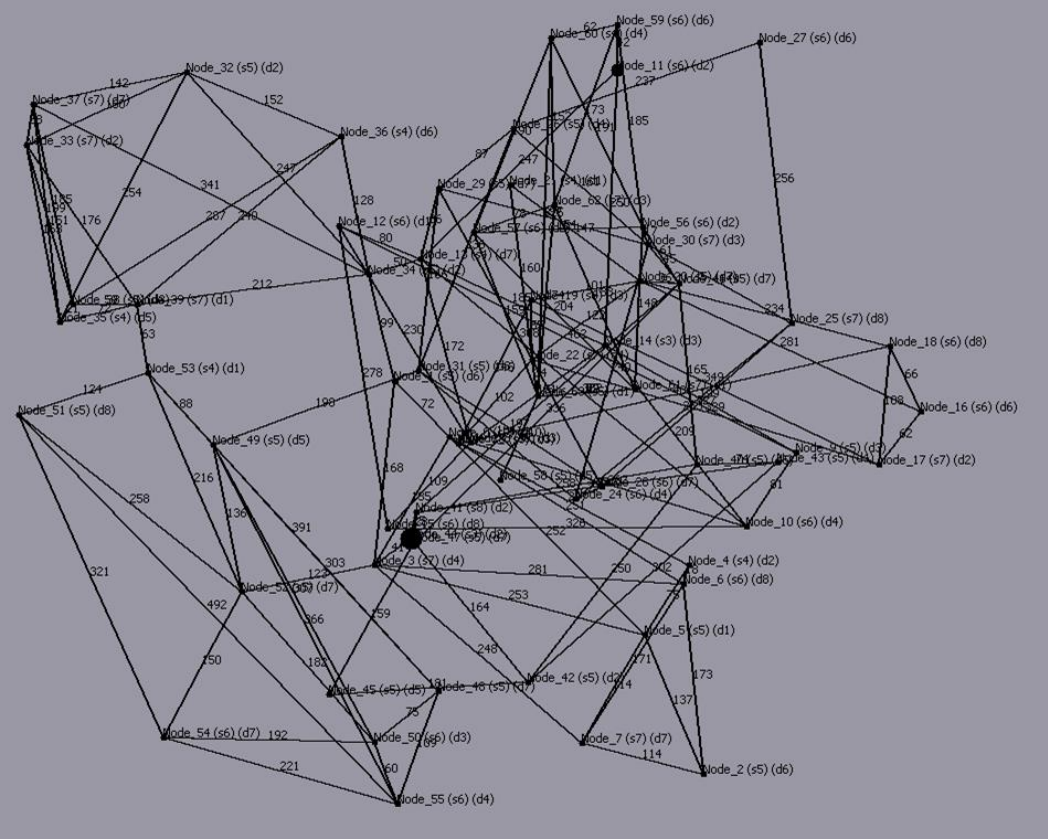
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.11-d, 6.11-e, and 6.11-f.



Appendix A Figure 4. Multi-Level SoS B with Node Status

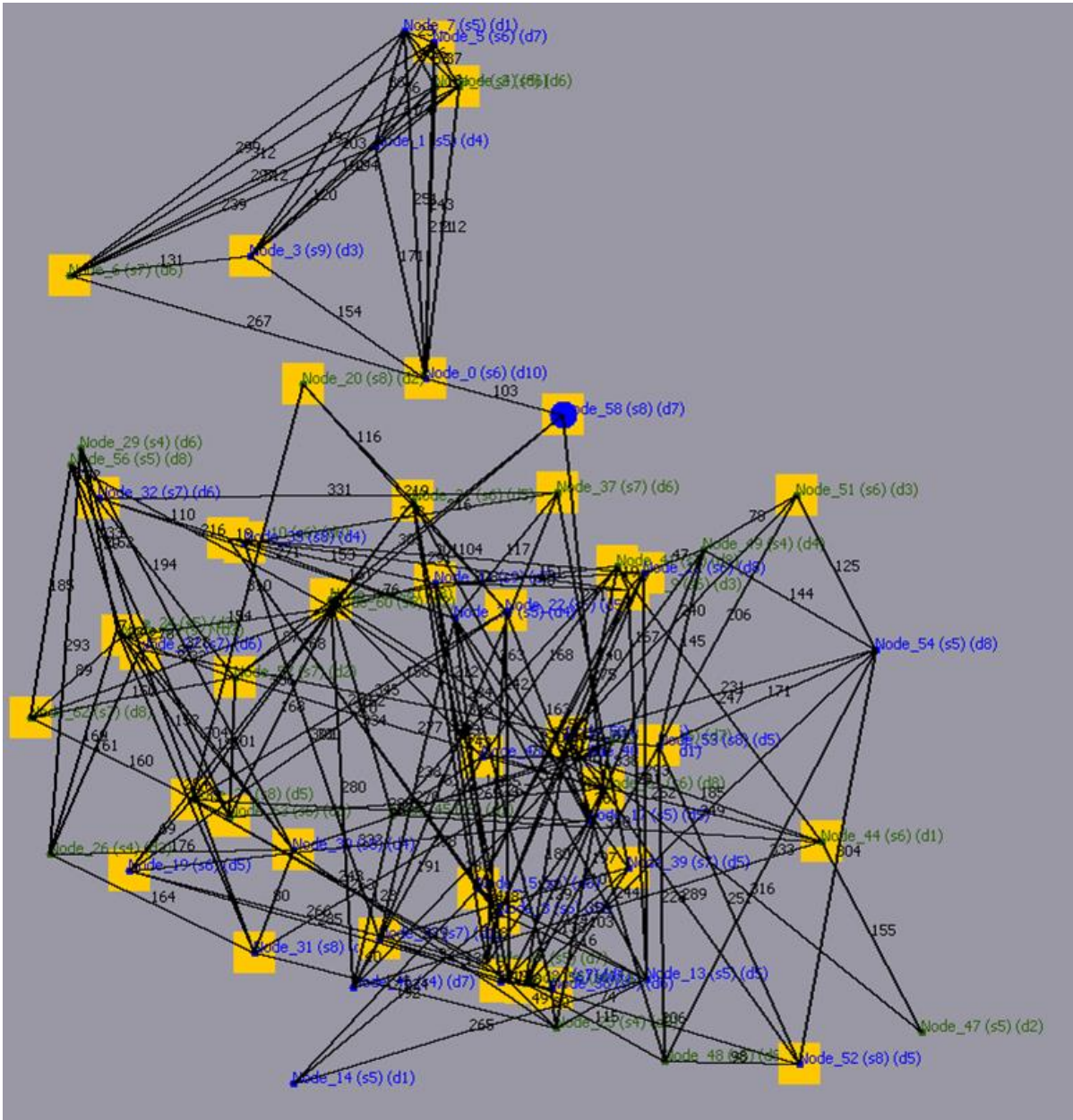


Appendix A Figure 5. Multi-Level SoS B Topology

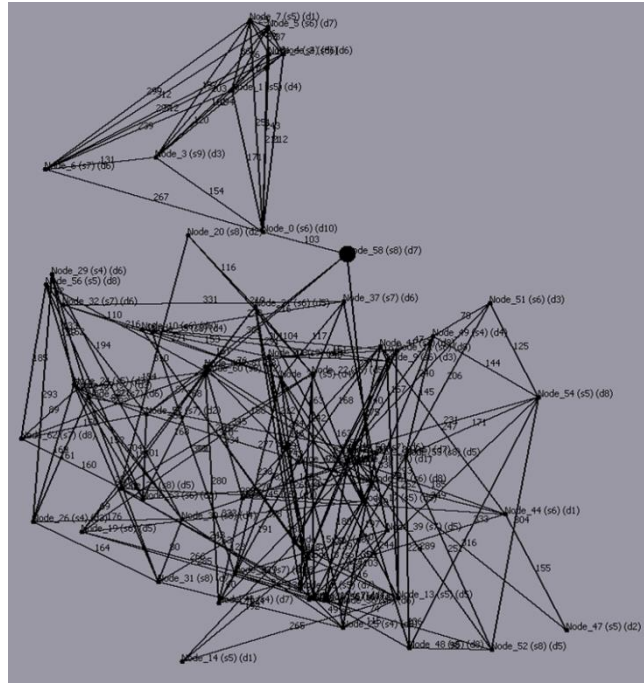


Appendix A Figure 6. Multi-Level SoS B Optimum Candidate

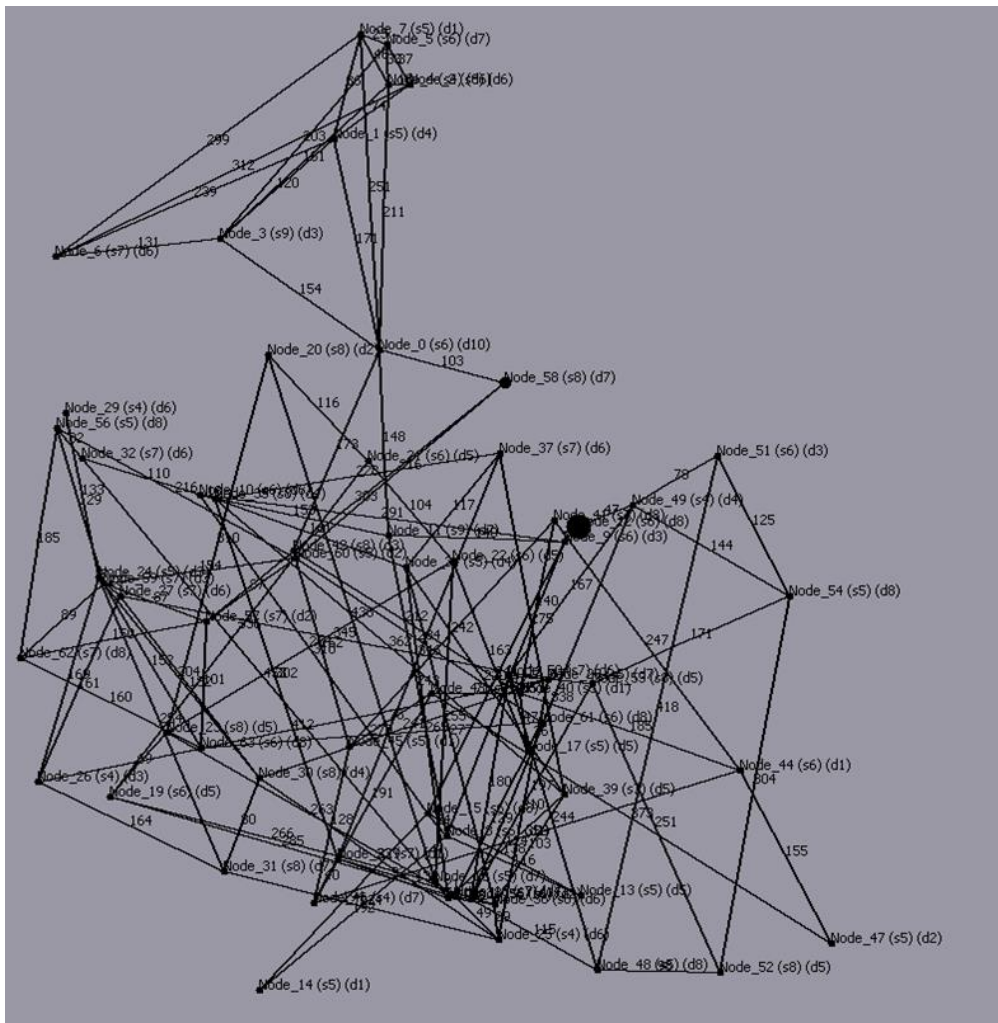
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.11-g, 6.11-h, and 6.11-i.



Appendix A Figure 7. Multi-Level SoS C with Node Status

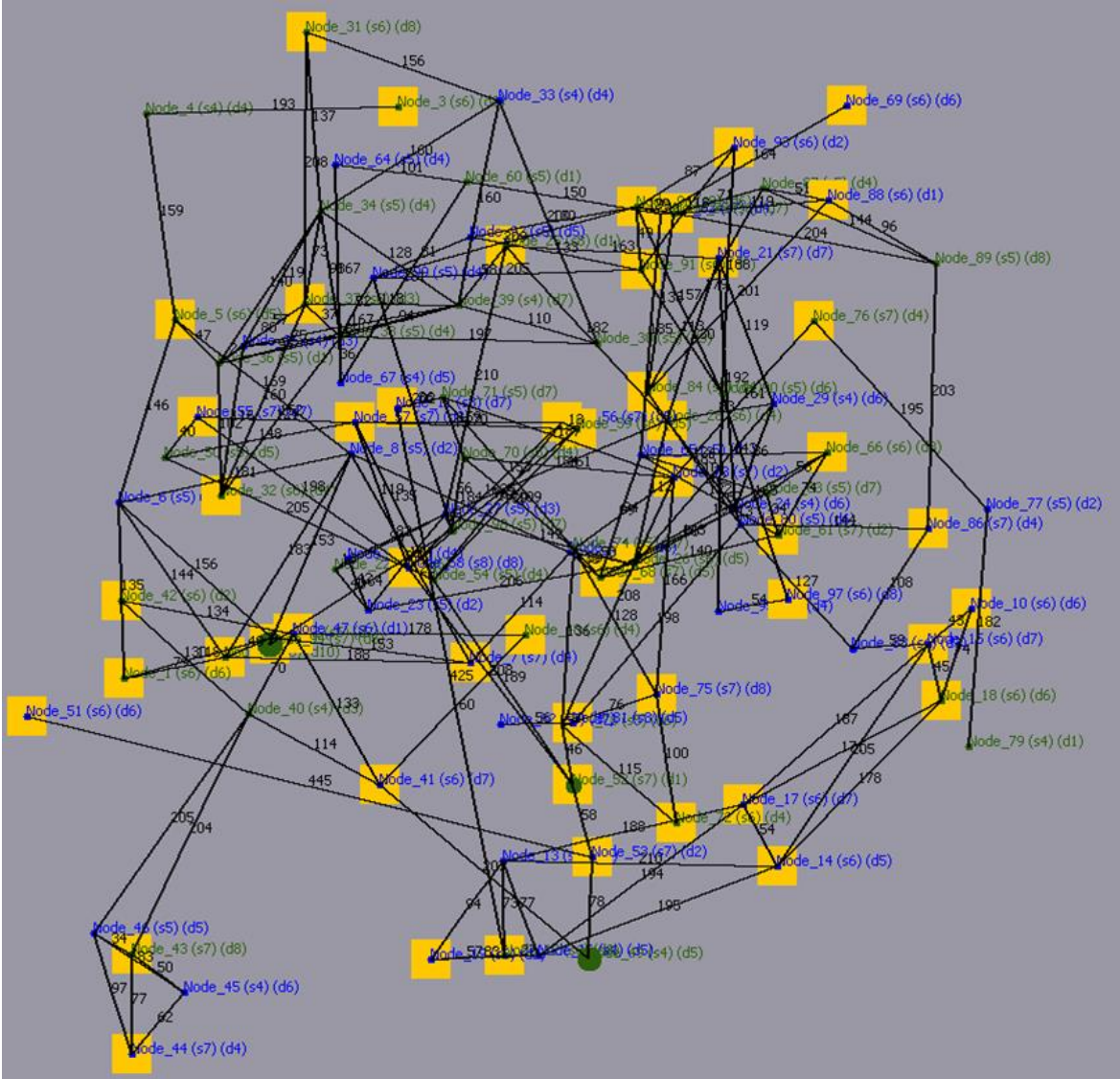


Appendix A Figure 8. Multi-Level SoS C Topology

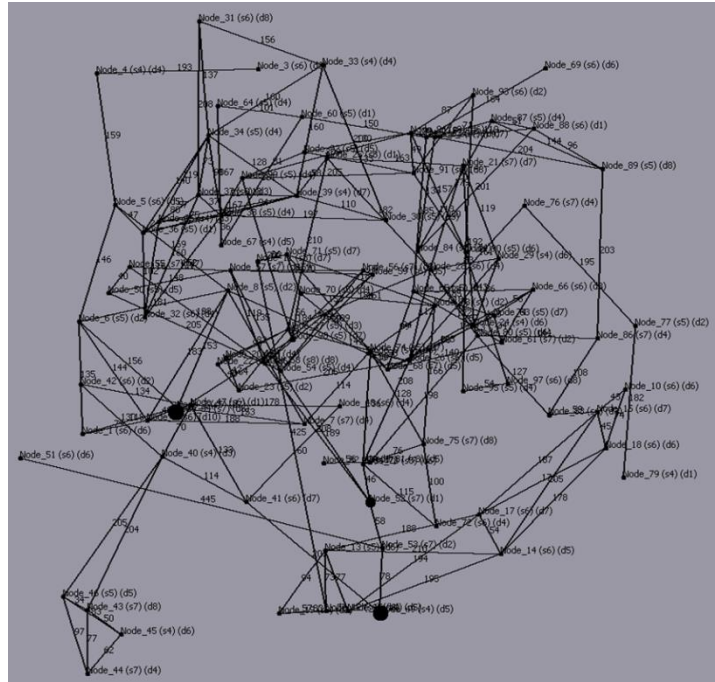


Appendix A Figure 9. Multi-Level SoS C Optimum Candidate

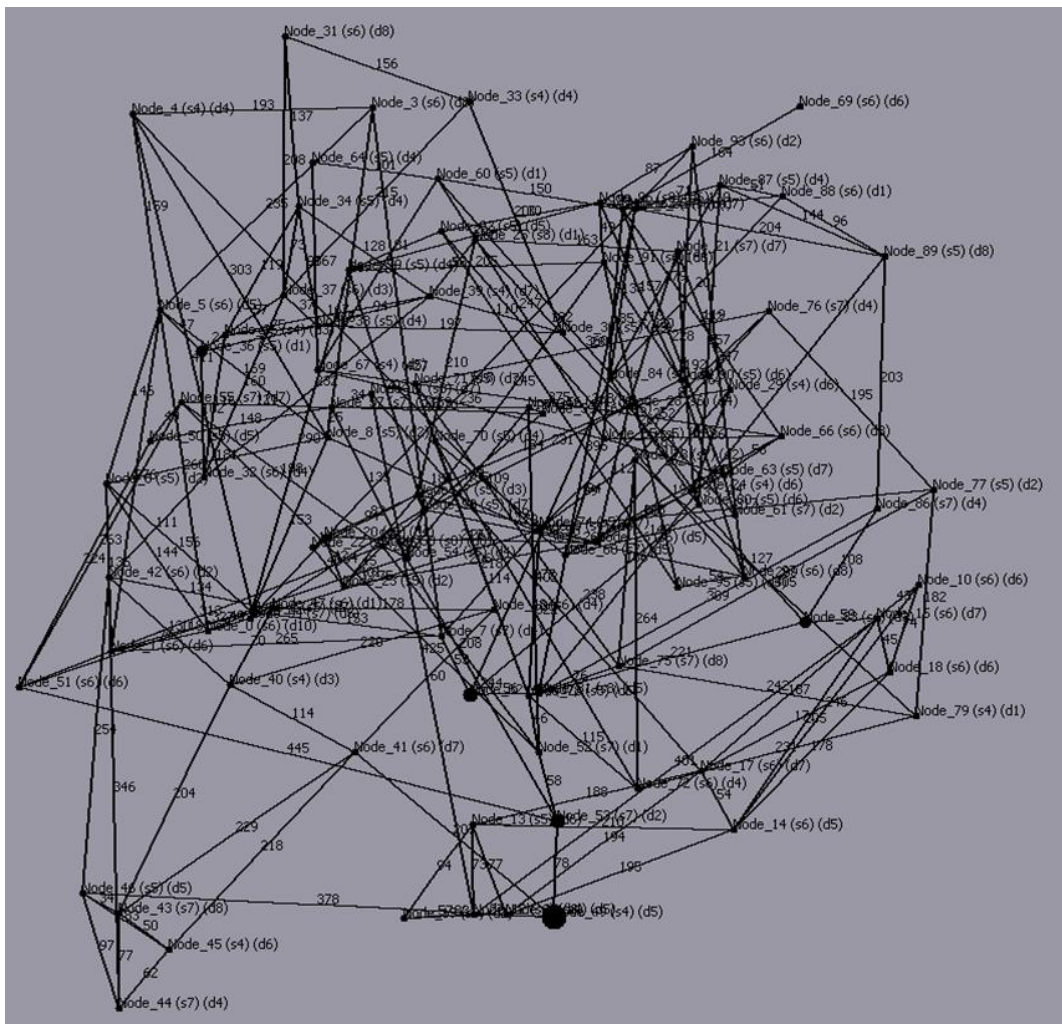
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.12-a, 6.12-b, and 6.12-c.



Appendix A Figure 10. Multi-Level SoS D with Node Status

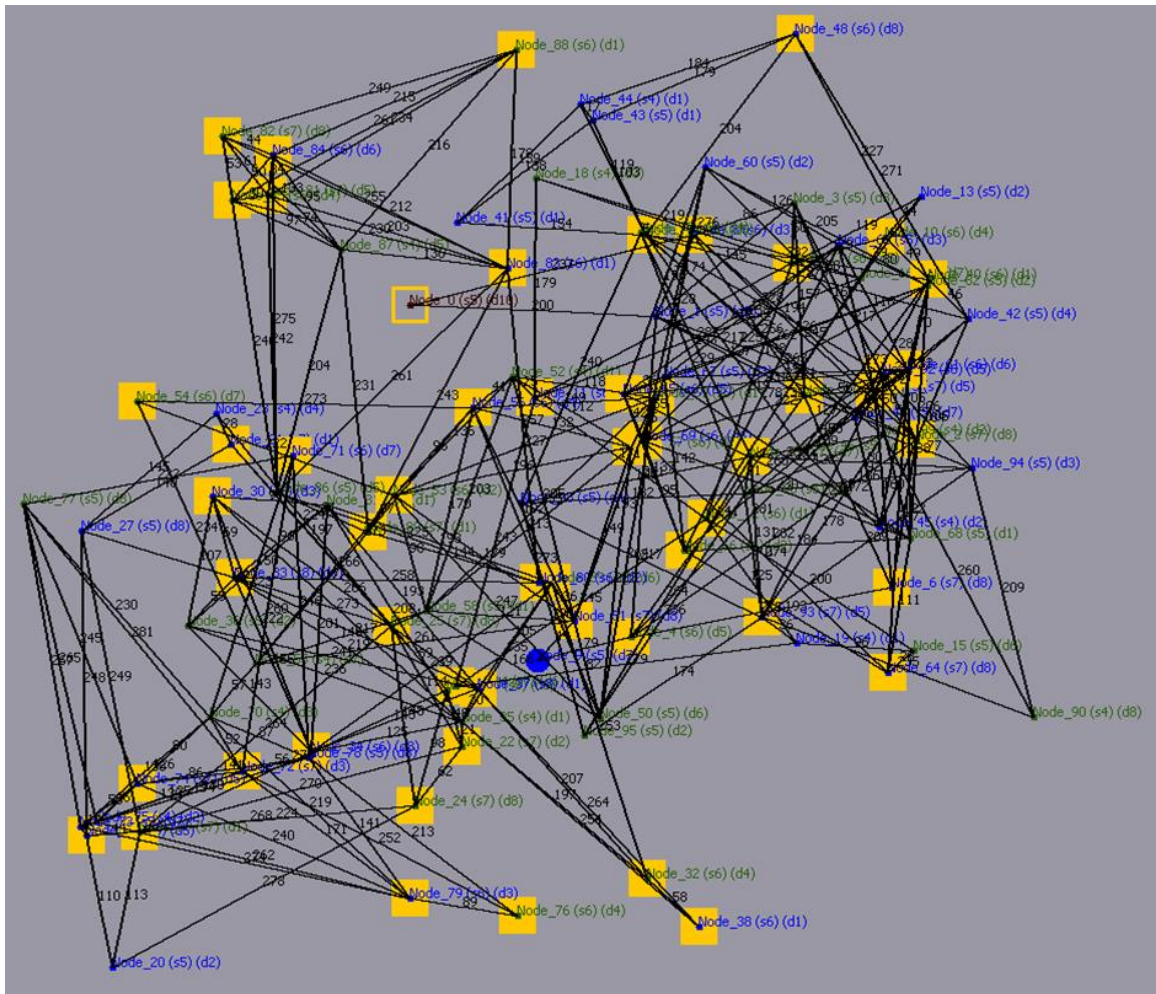


Appendix A Figure 11. Multi-Level SoS D Topology

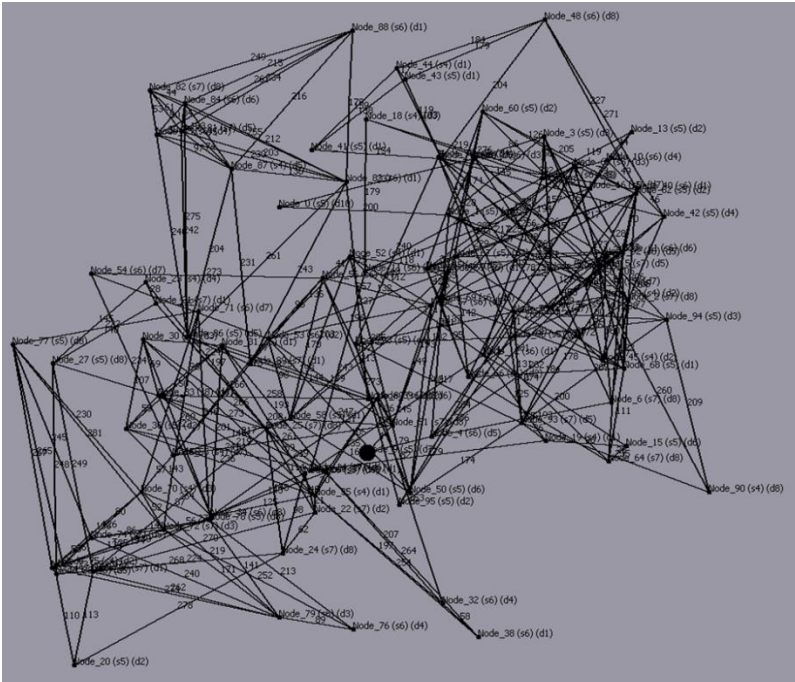


Appendix A Figure 12. Multi-Level SoS D Optimum Candidate

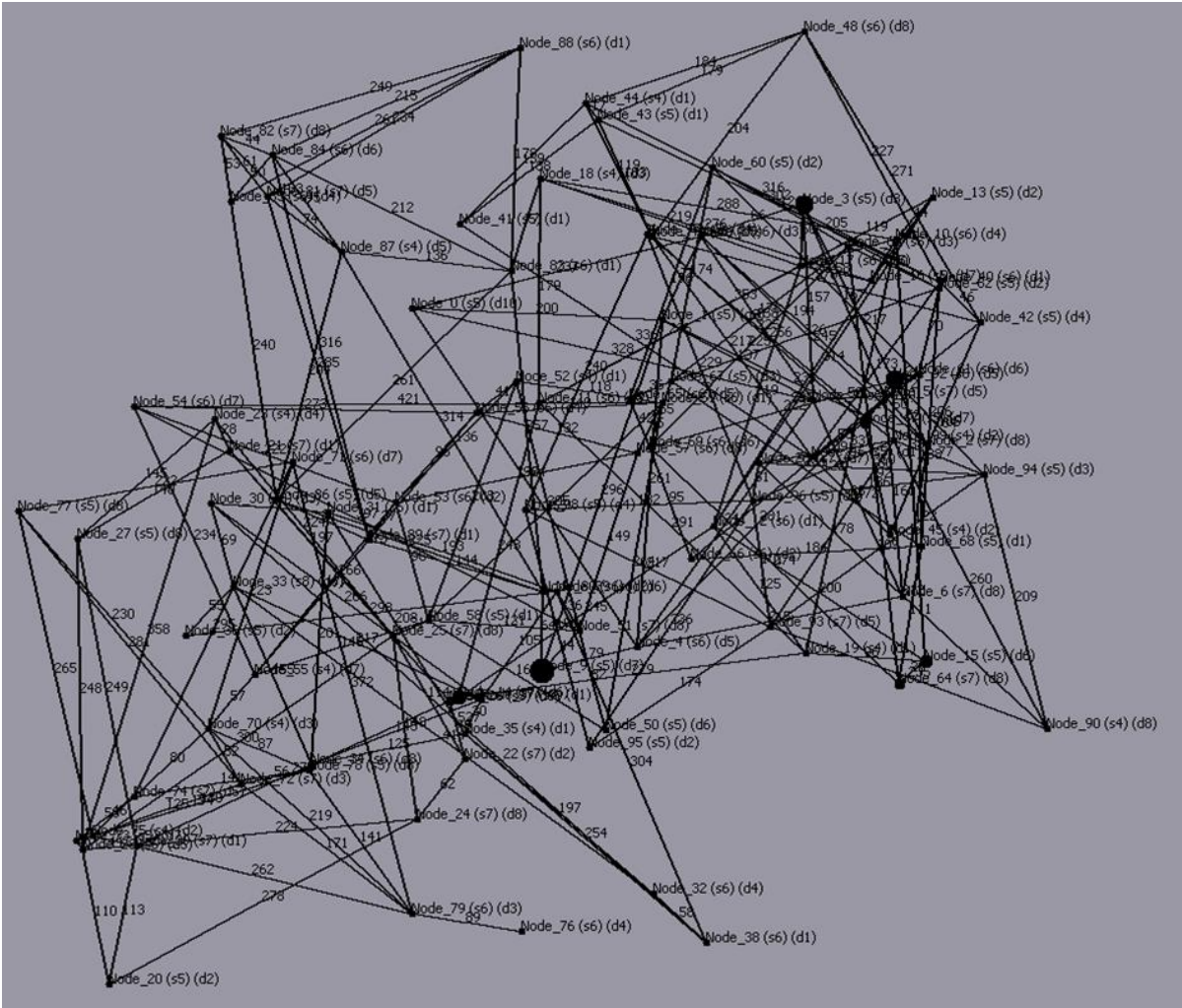
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.12-d, 6.12-e, and 6.12-f.



Appendix A Figure 13. Multi-Level SoS E with Node Status

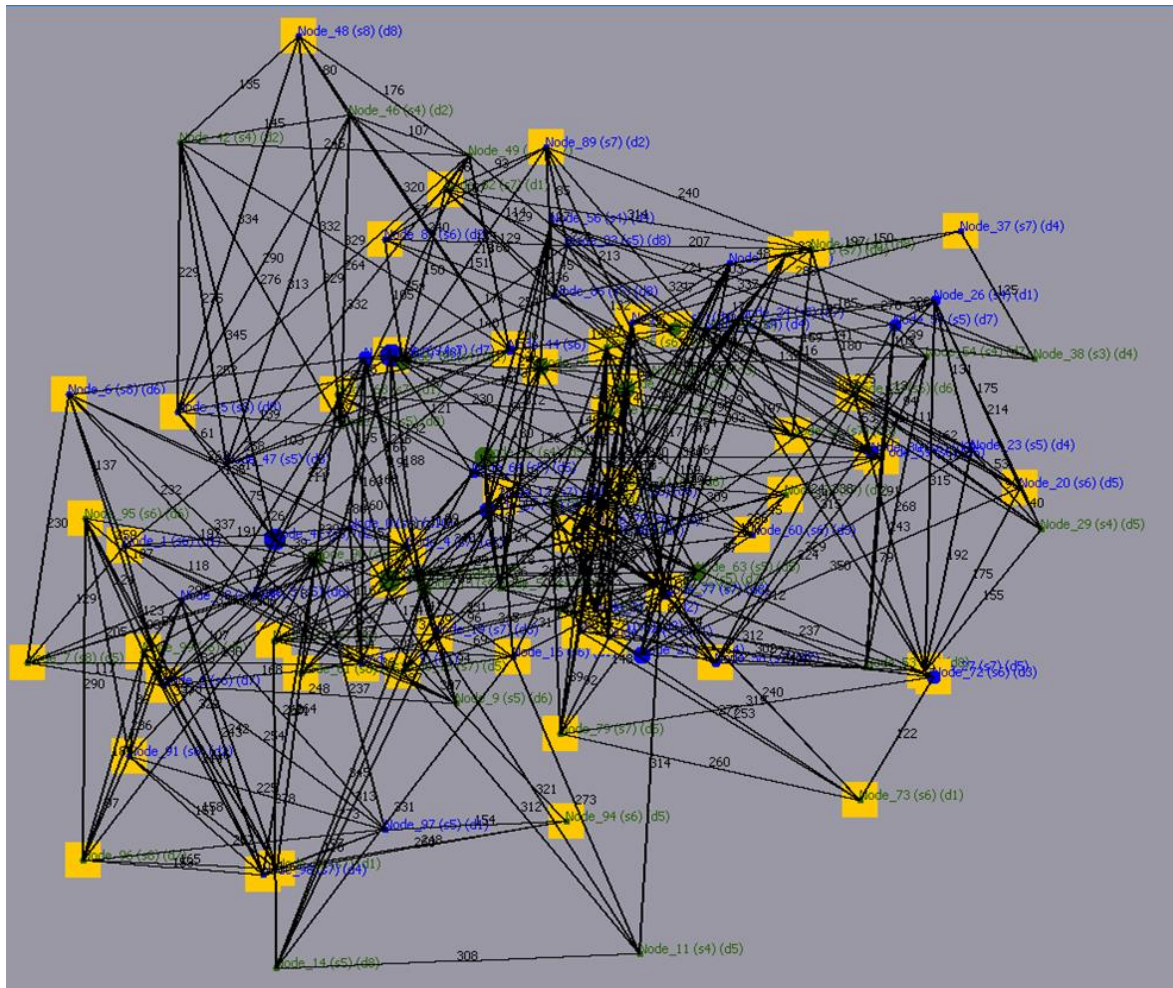


Appendix A Figure 14. Multi-Level SoS E Topology

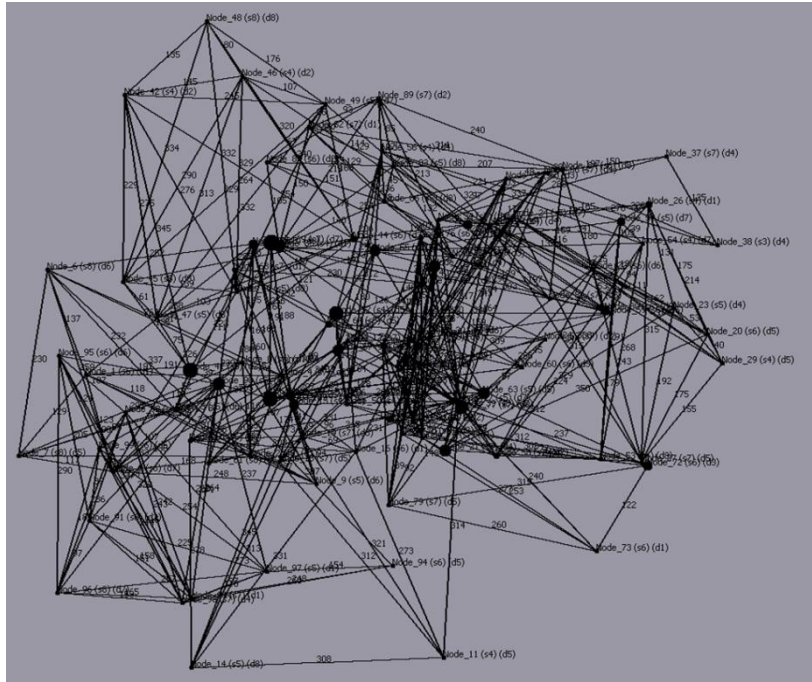


Appendix A Figure 15. Multi-Level SoS E Optimum Candidate

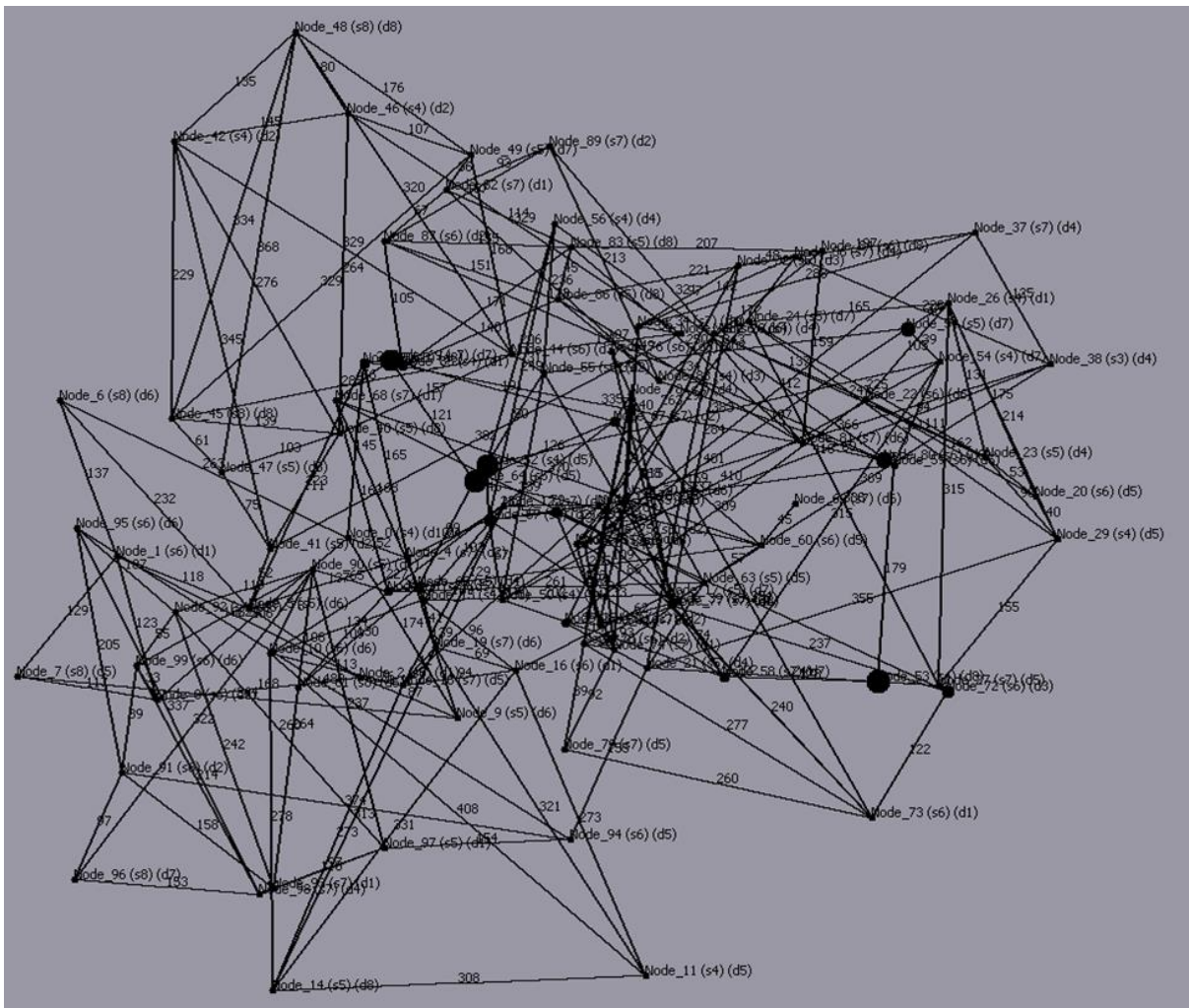
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.12-g, 6.12-h, and 6.12-i.



Appendix A Figure 16. Multi-Level SoS F with Node Status

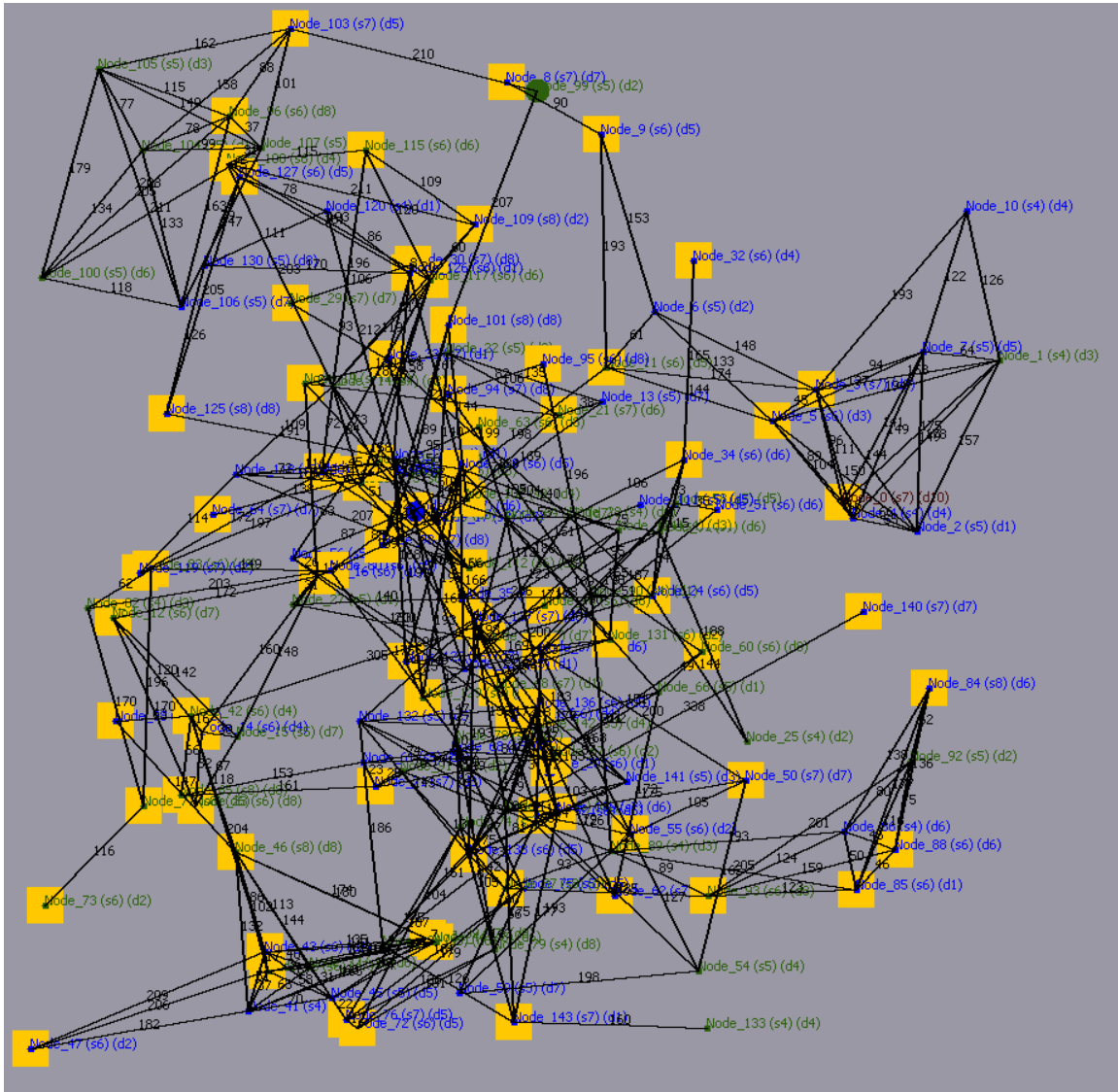


Appendix A Figure 17. Multi-Level SoS F Topology

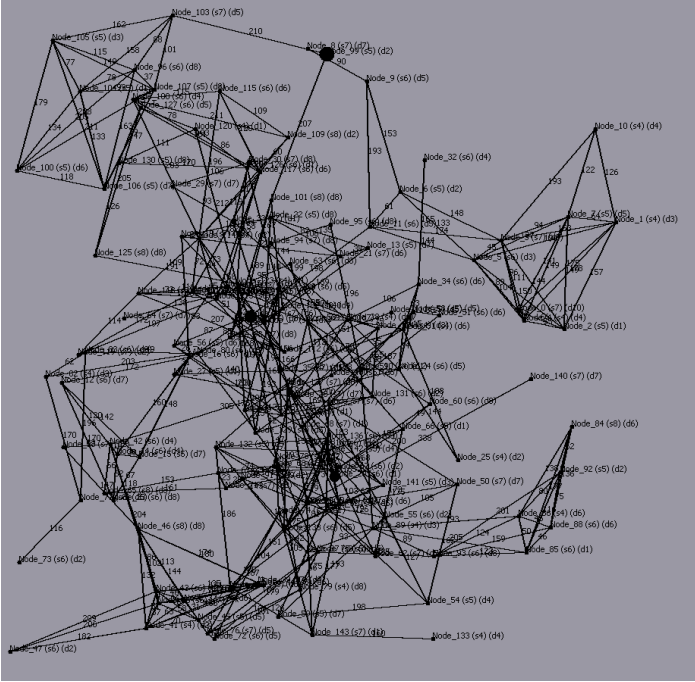


Appendix A Figure 18. Multi-Level SoS F Optimum Candidate

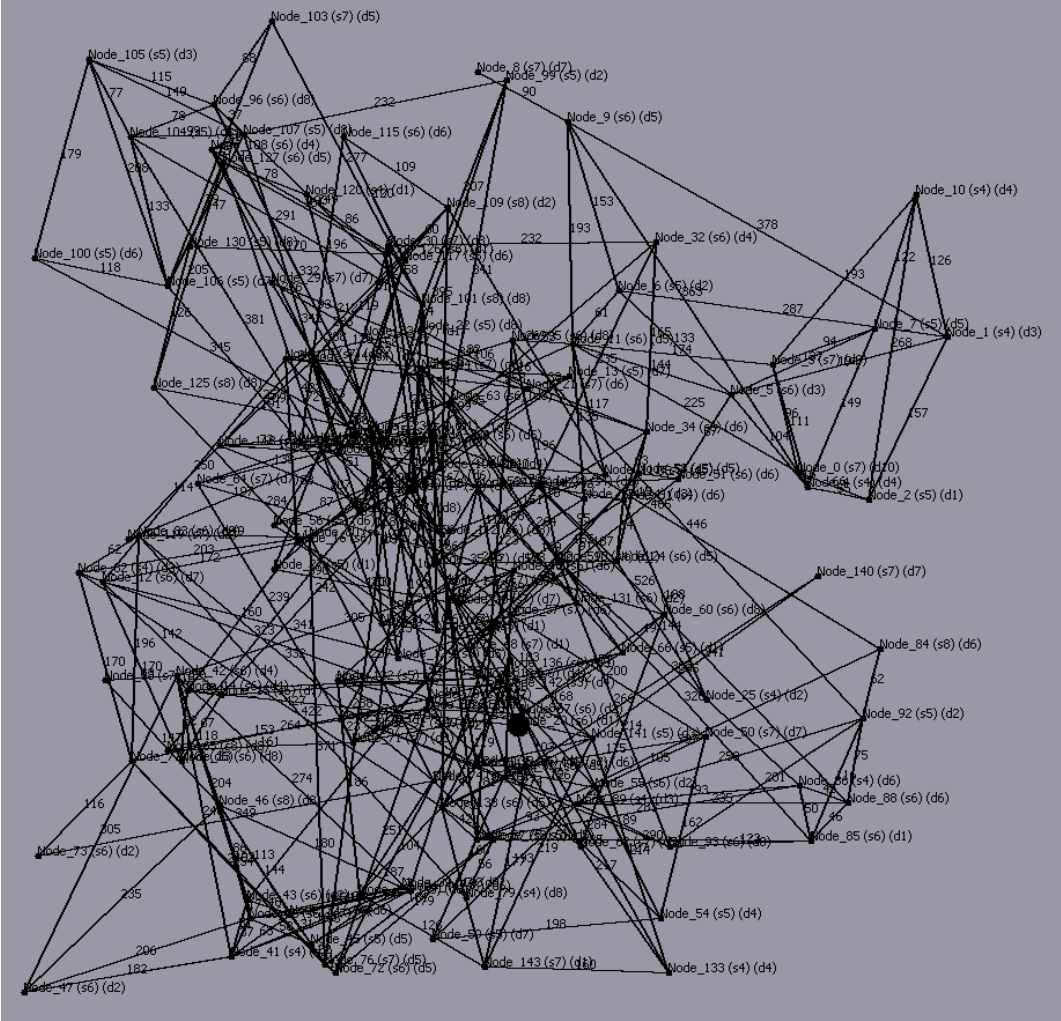
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.13-a, 6.13-b, and 6.13-c.



Appendix A Figure 19. Multi-Level SoS G with Node Status

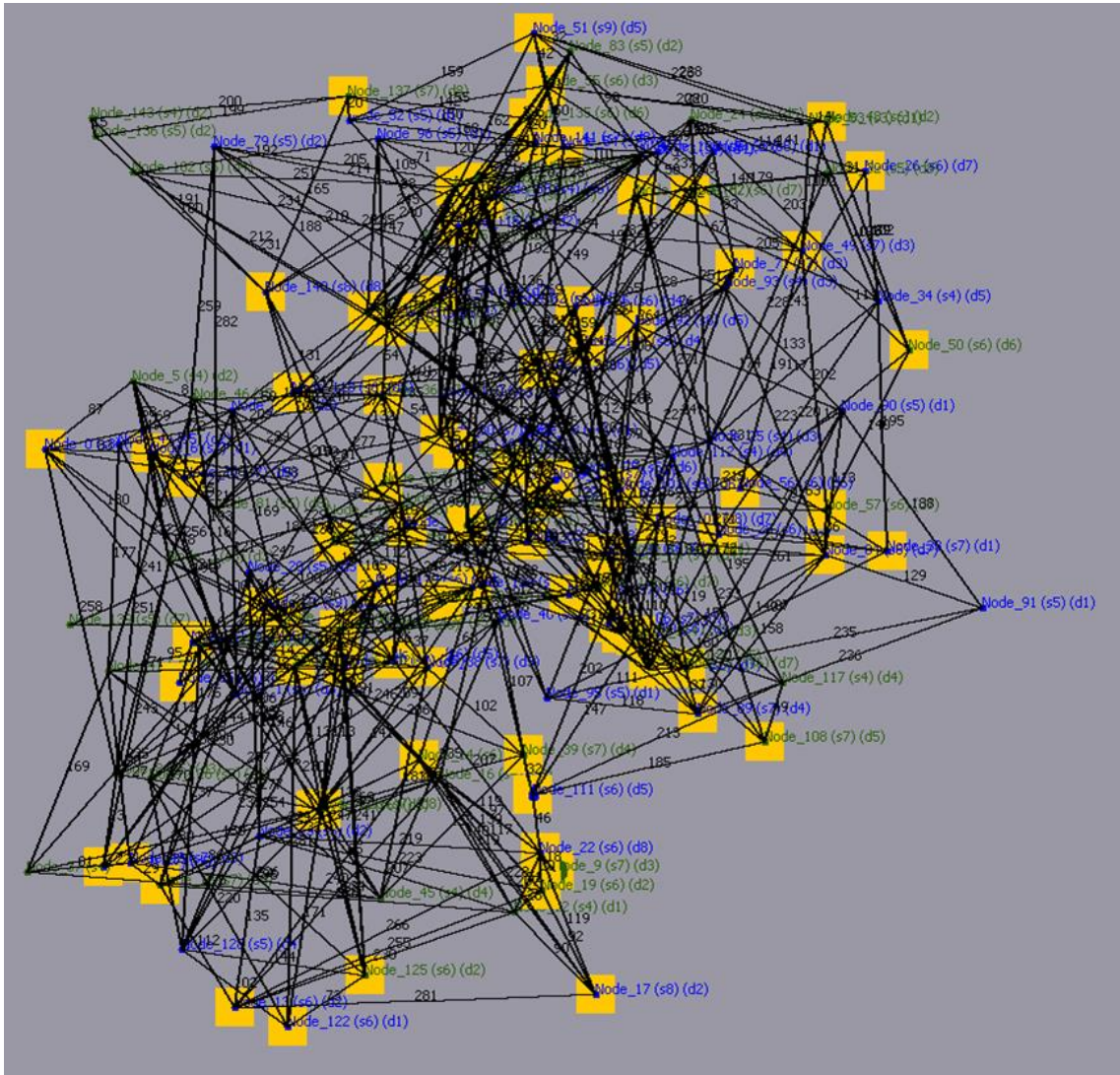


Appendix A Figure 20. Multi-Level SoS G Topology

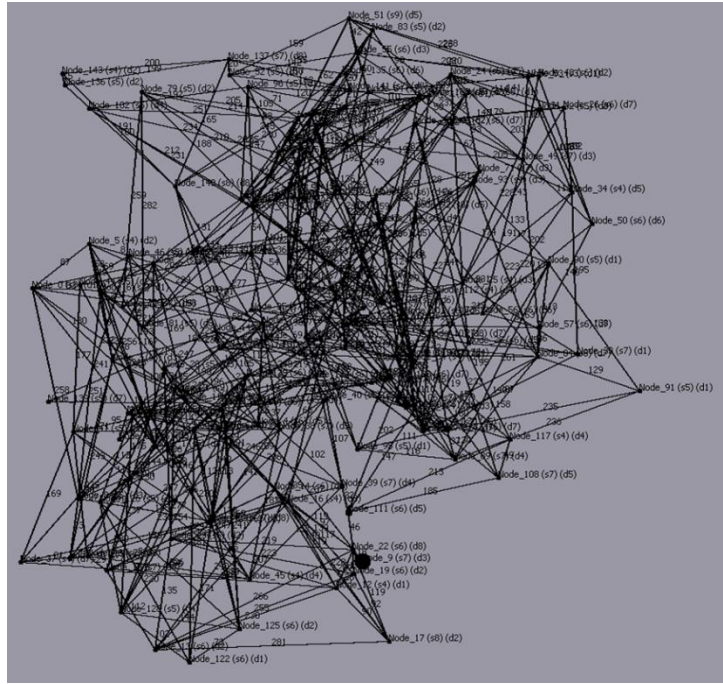


Appendix A Figure 21. Multi-Level SoS G Optimum Candidate

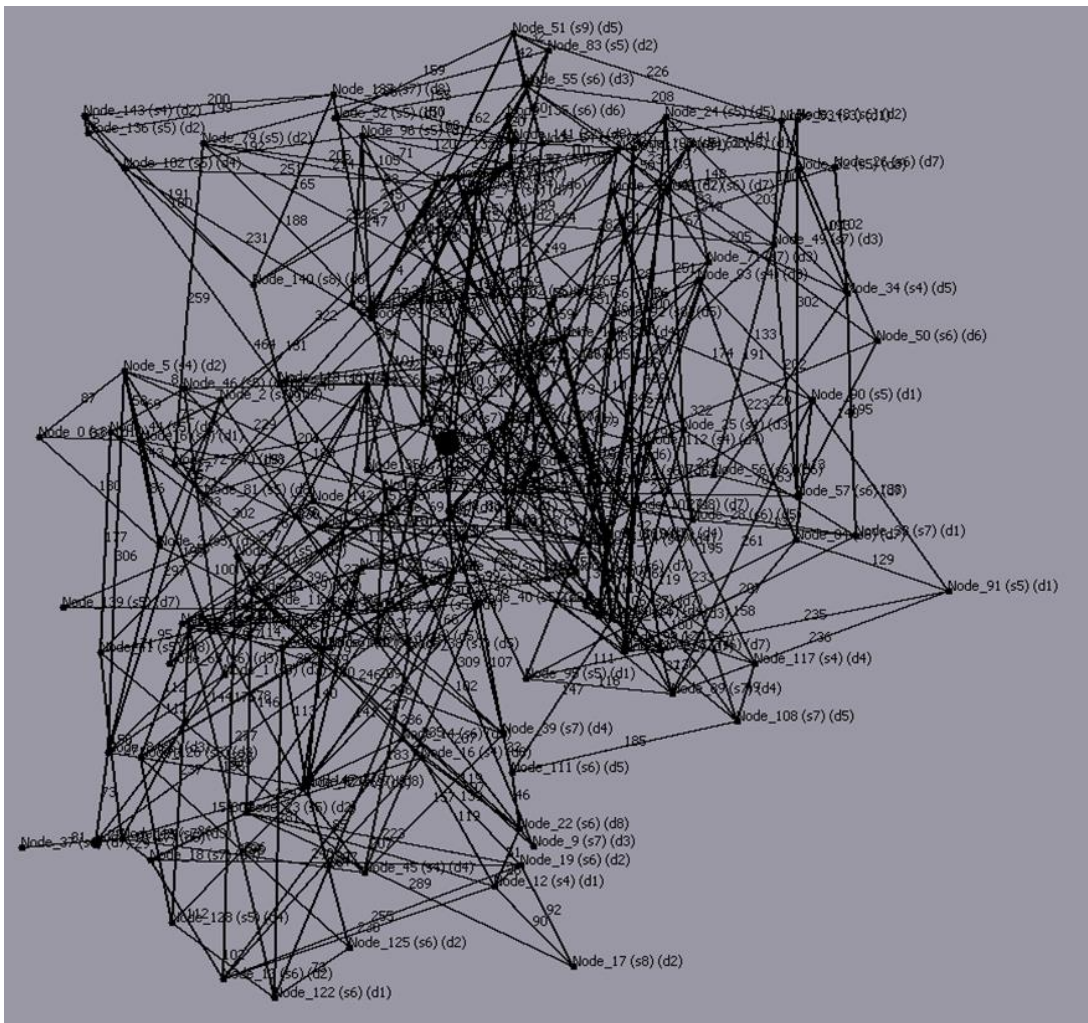
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.13-d, 6.13-e, and 6.13-f.



Appendix A Figure 22. Multi-Level SoS H with Node Status

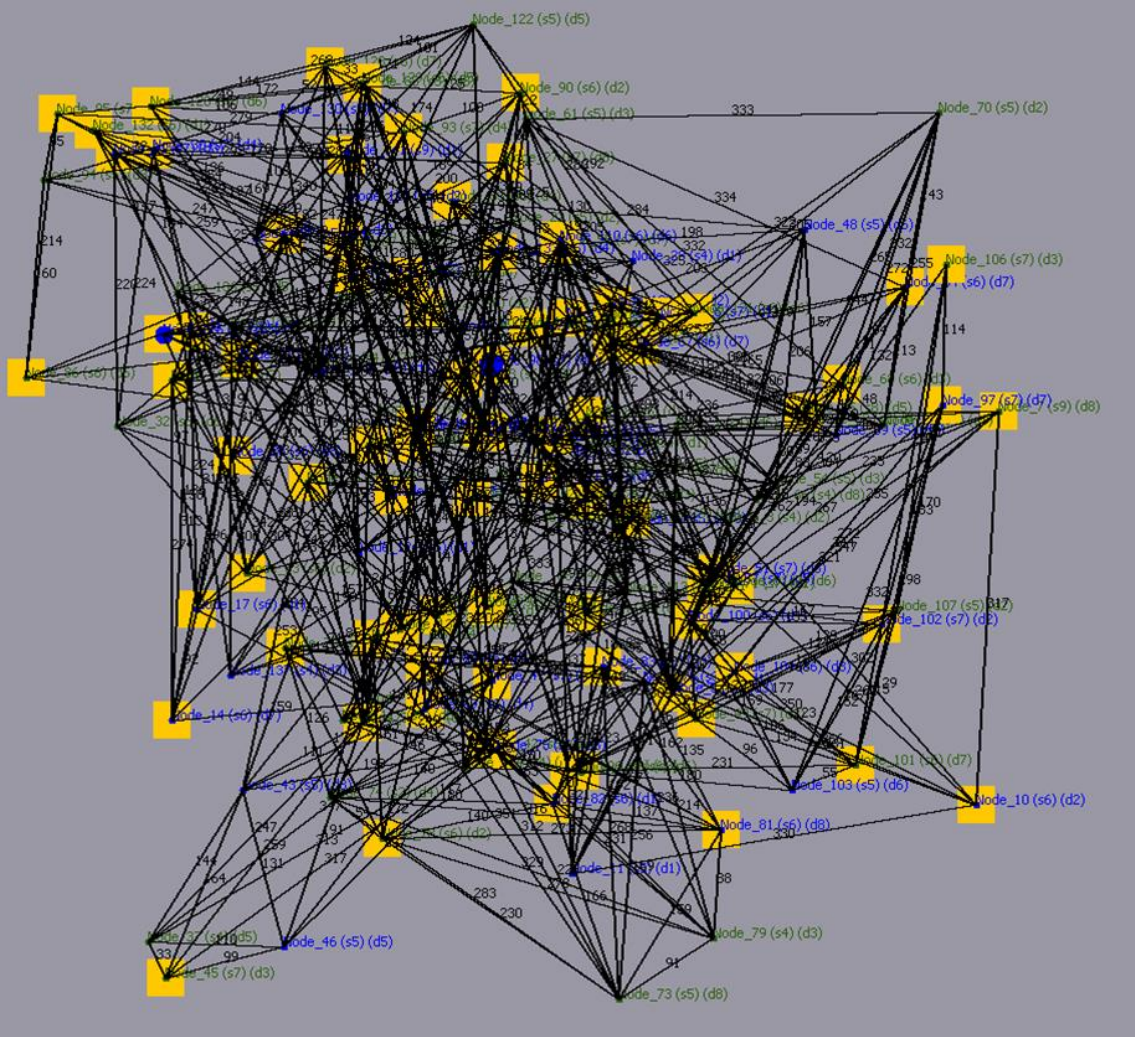


Appendix A Figure 23. Multi-Level SoS H Topology

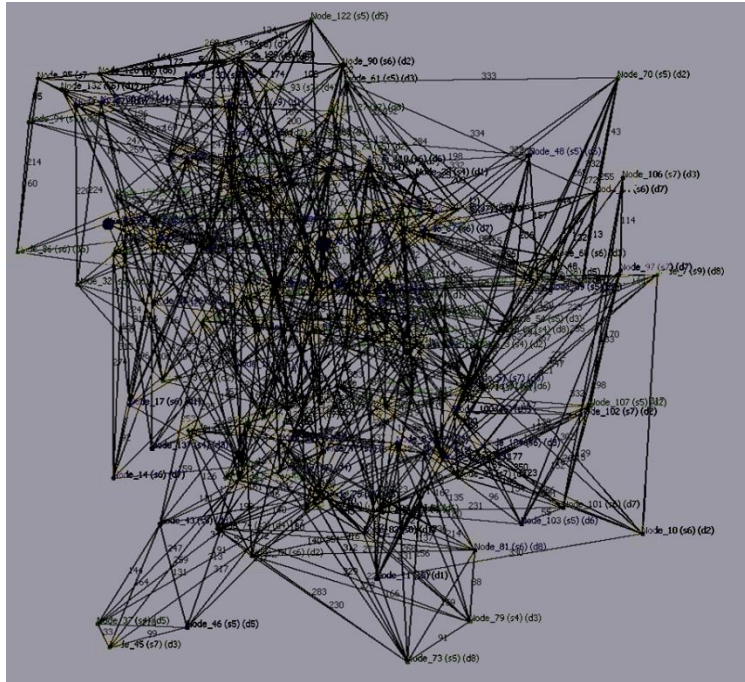


Appendix A Figure 24. Multi-Level SoS H Optimum Candidate

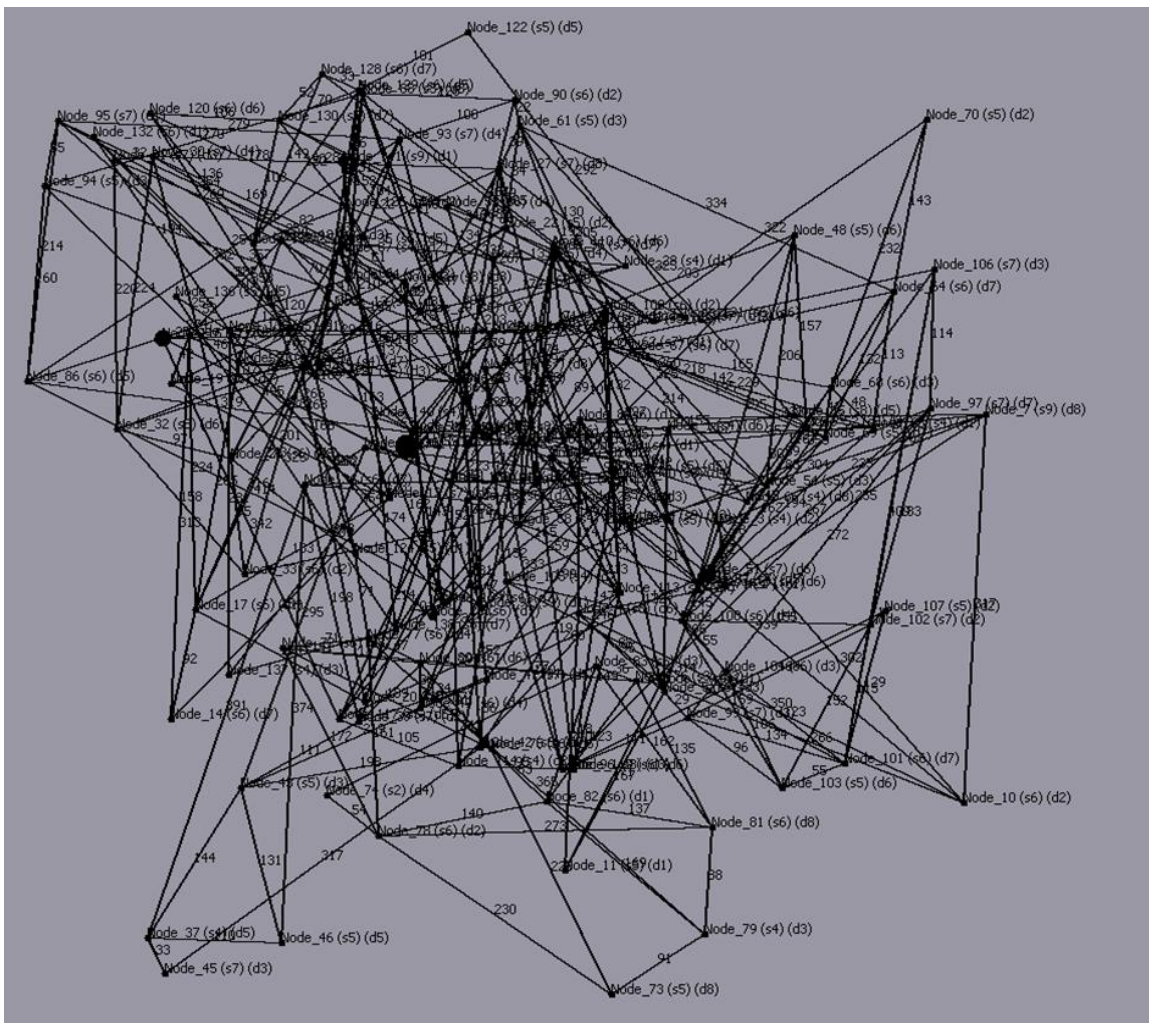
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.13-g, 6.13-h, and 6.13-i.



Appendix A Figure 25. Multi-Level SoS I with Node Status



Appendix A Figure 26. Multi-Level SoS I Topology

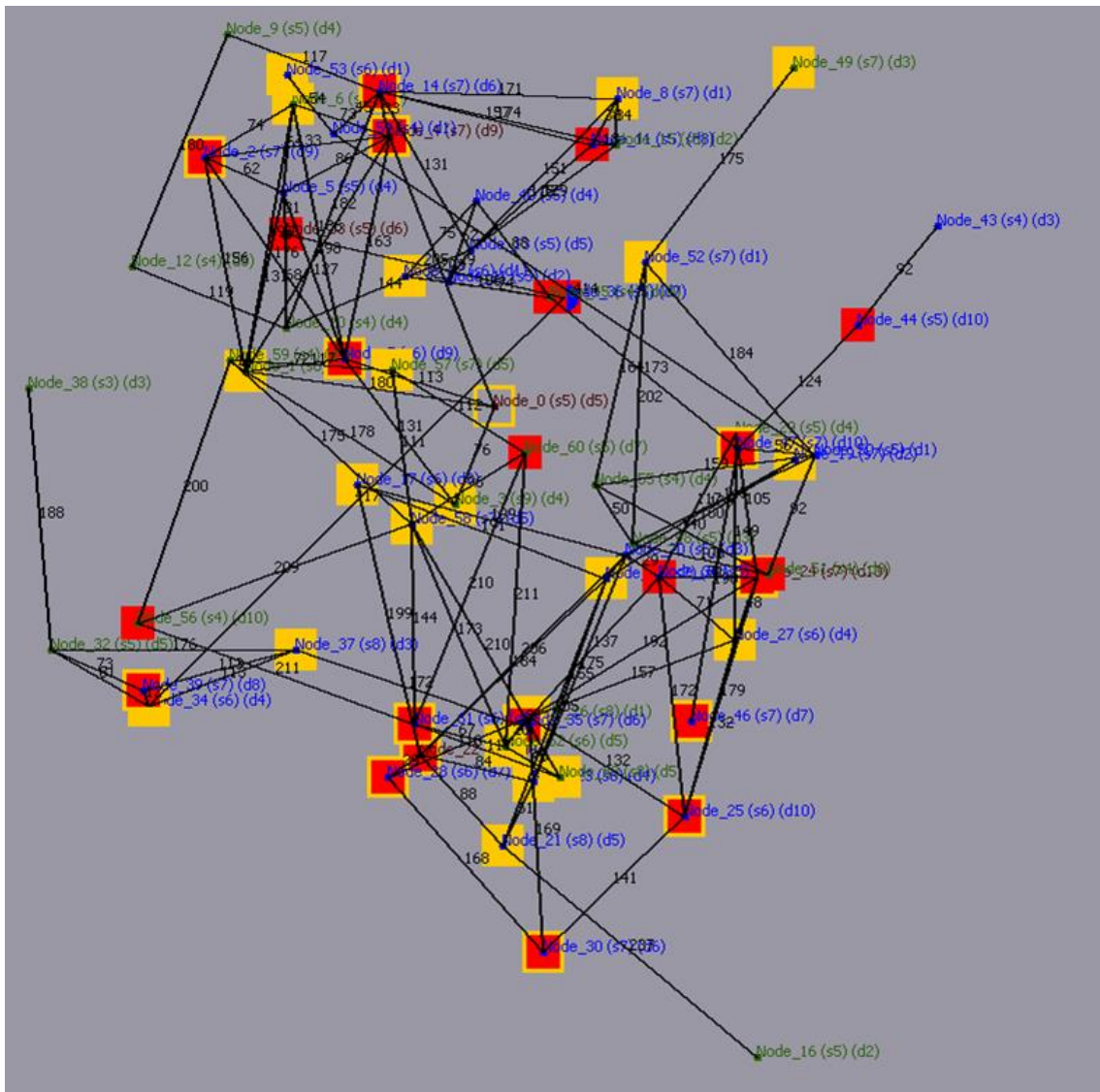


Appendix A Figure 27. Multi-Level SoS I Optimum Candidate

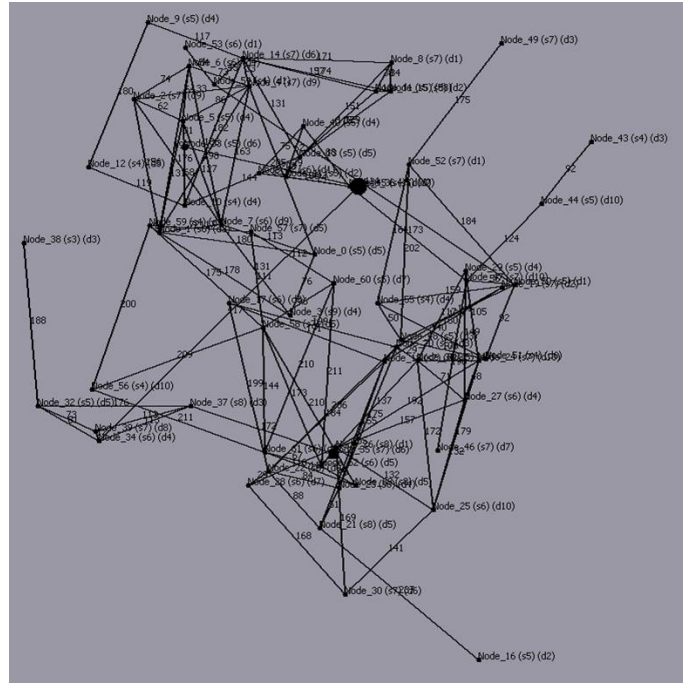
Appendix B

SCRAM Positive Multi-Level SoS Vulnerability and Data Access Performance

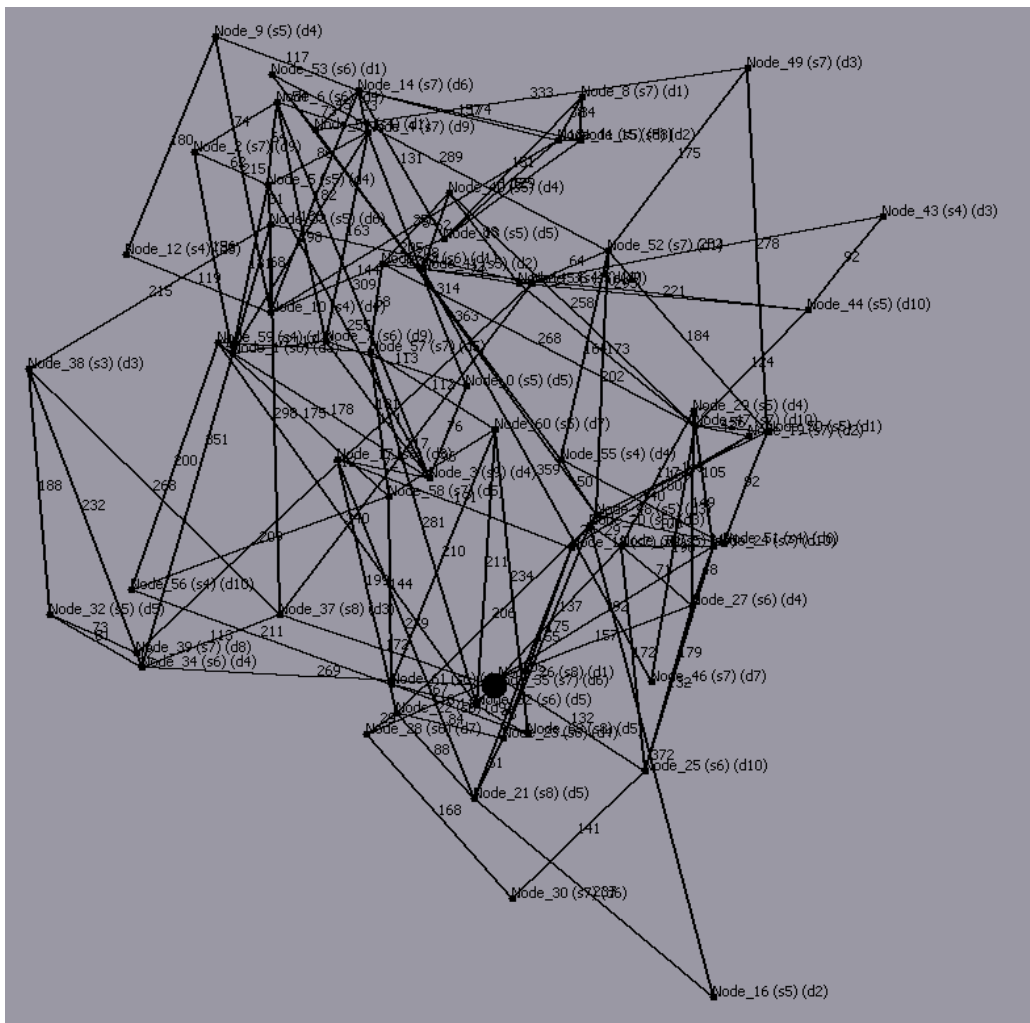
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.17-a, 6.17-b, and 6.17-c.



Appendix B Figure 1. Multi-Level SoS J with Node Status

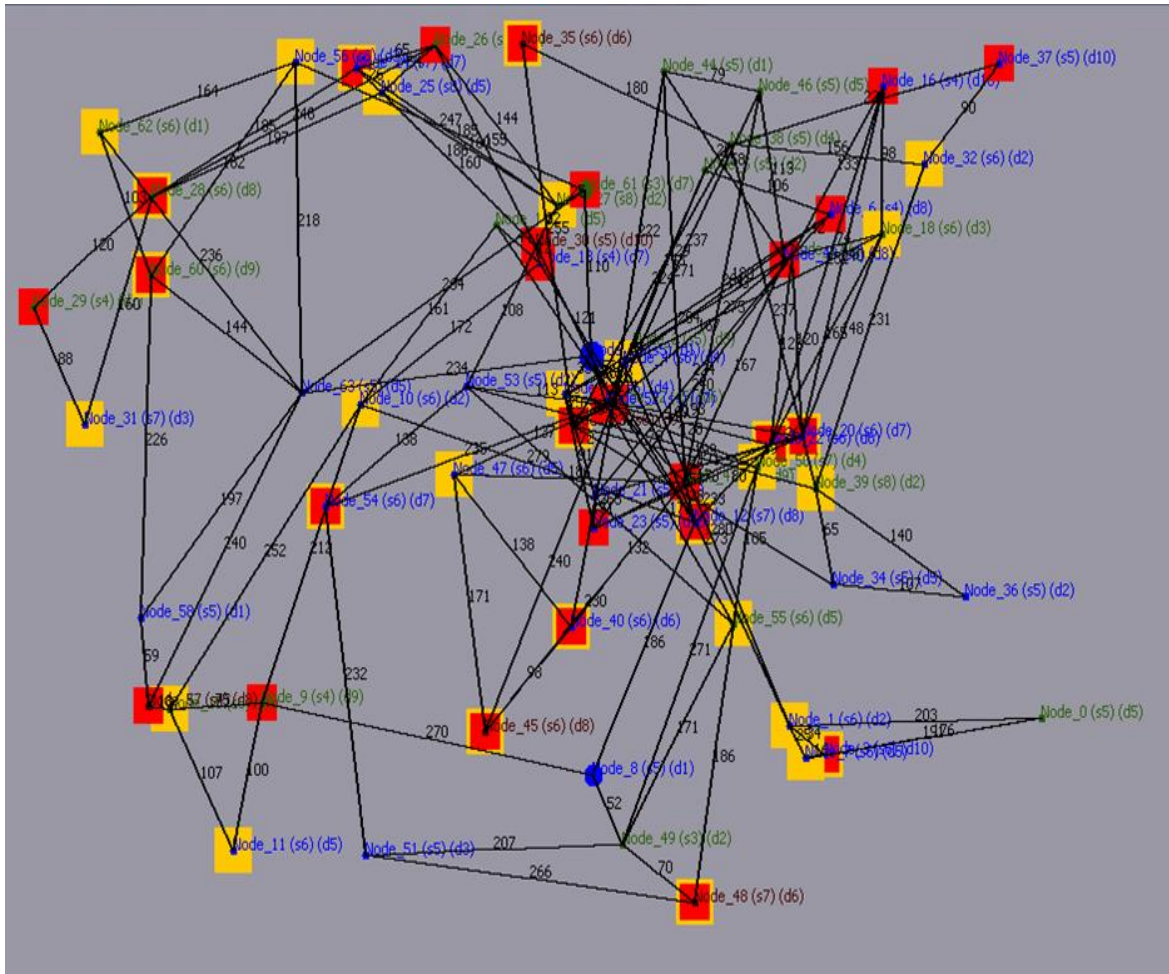


Appendix B Figure 2. Multi-Level SoS J Topology

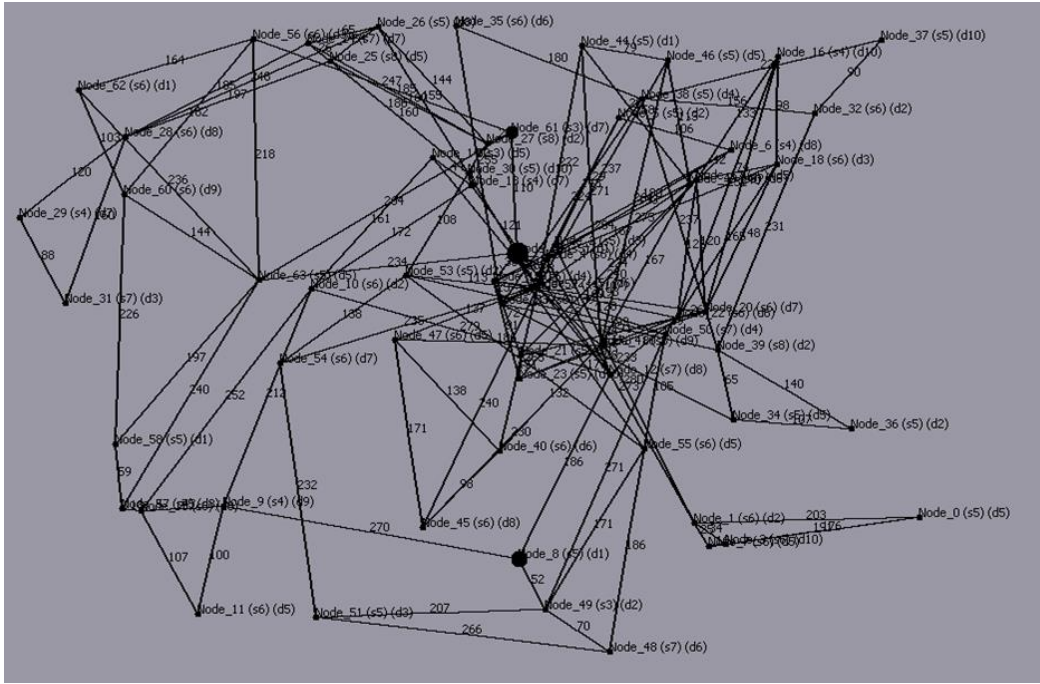


Appendix B Figure 3. Multi-Level SoS J Optimum Candidate

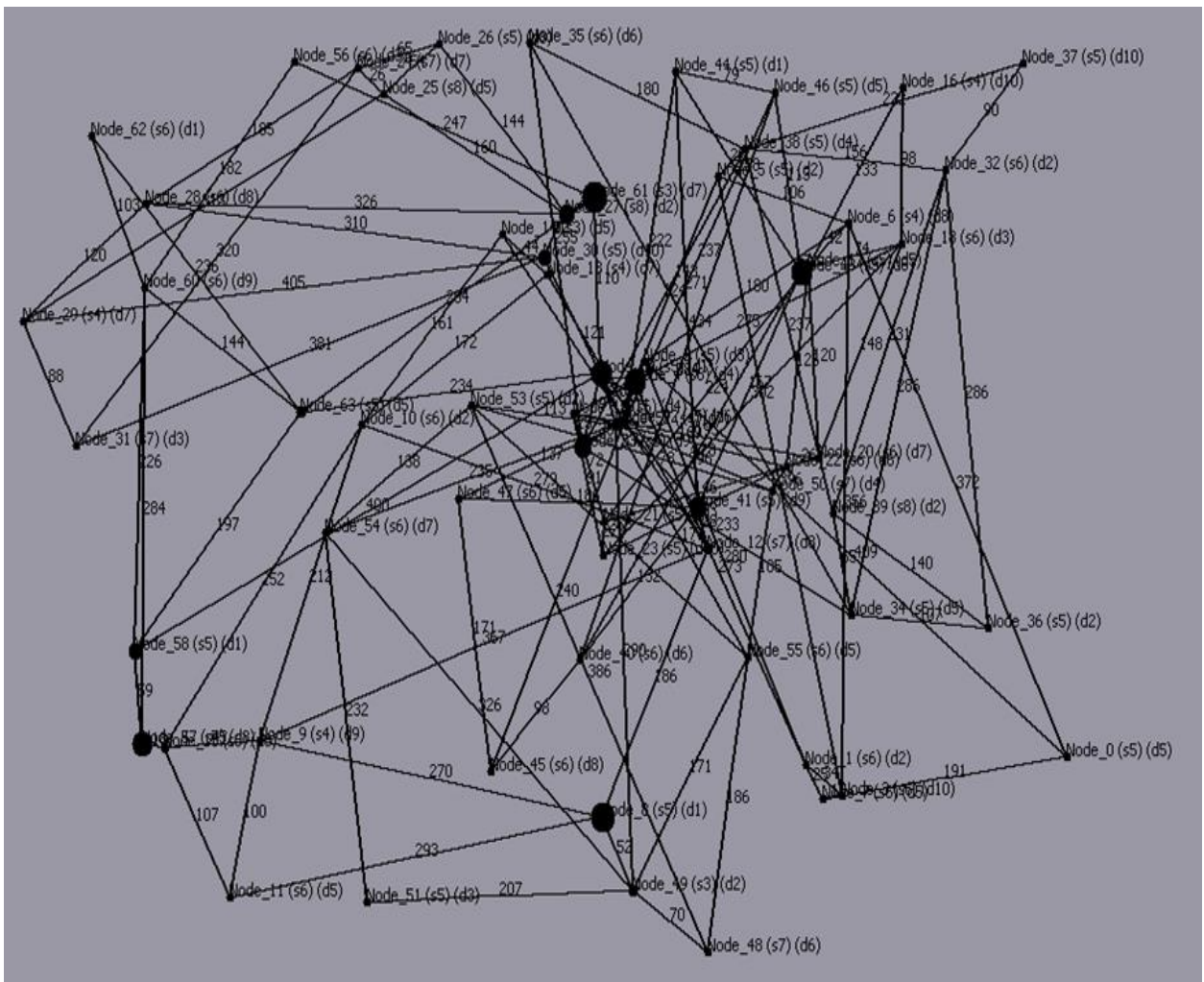
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.17-d, 6.17-e, and 6.17-f.



Appendix B Figure 4. Multi-Level SoS K with Node Status

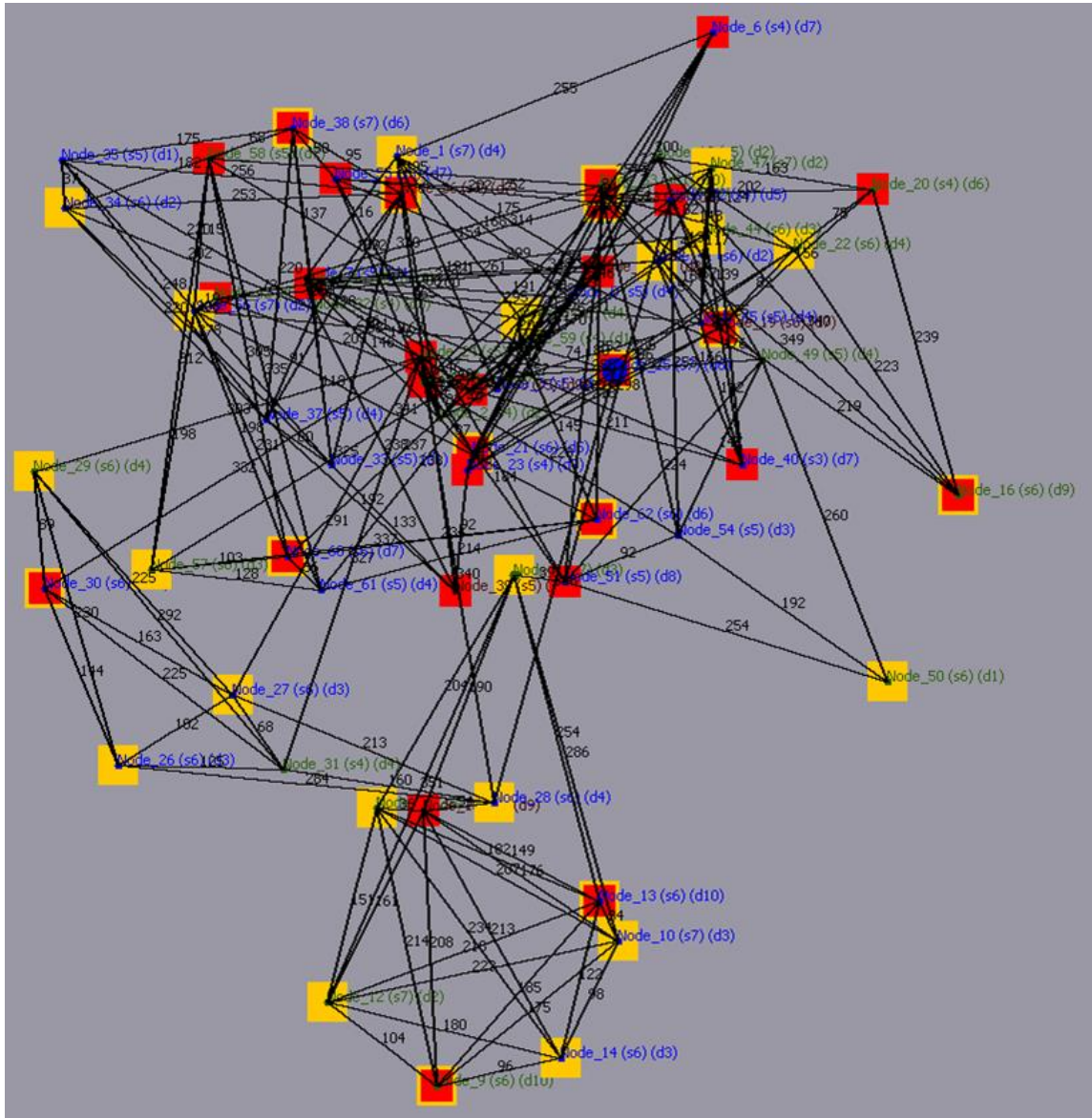


Appendix B Figure 5. Multi-Level SoS K Topology

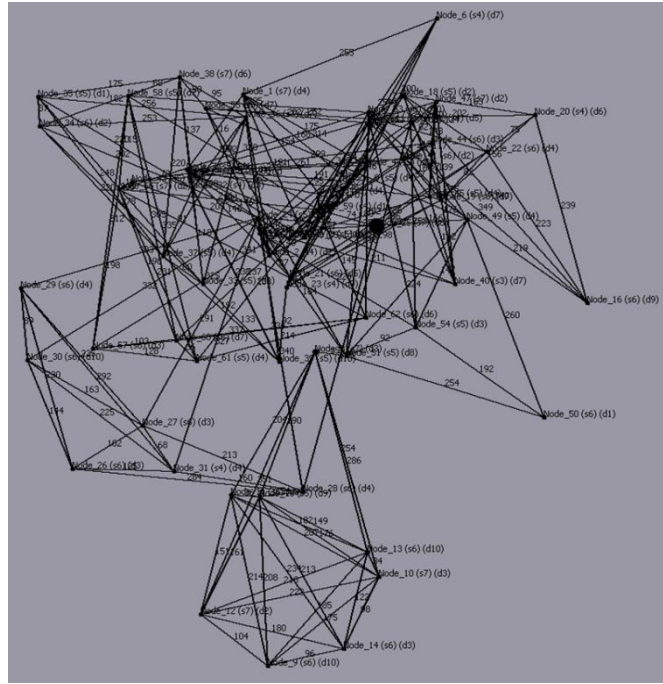


Appendix B Figure 6. Multi-Level SoS K Optimum Candidate

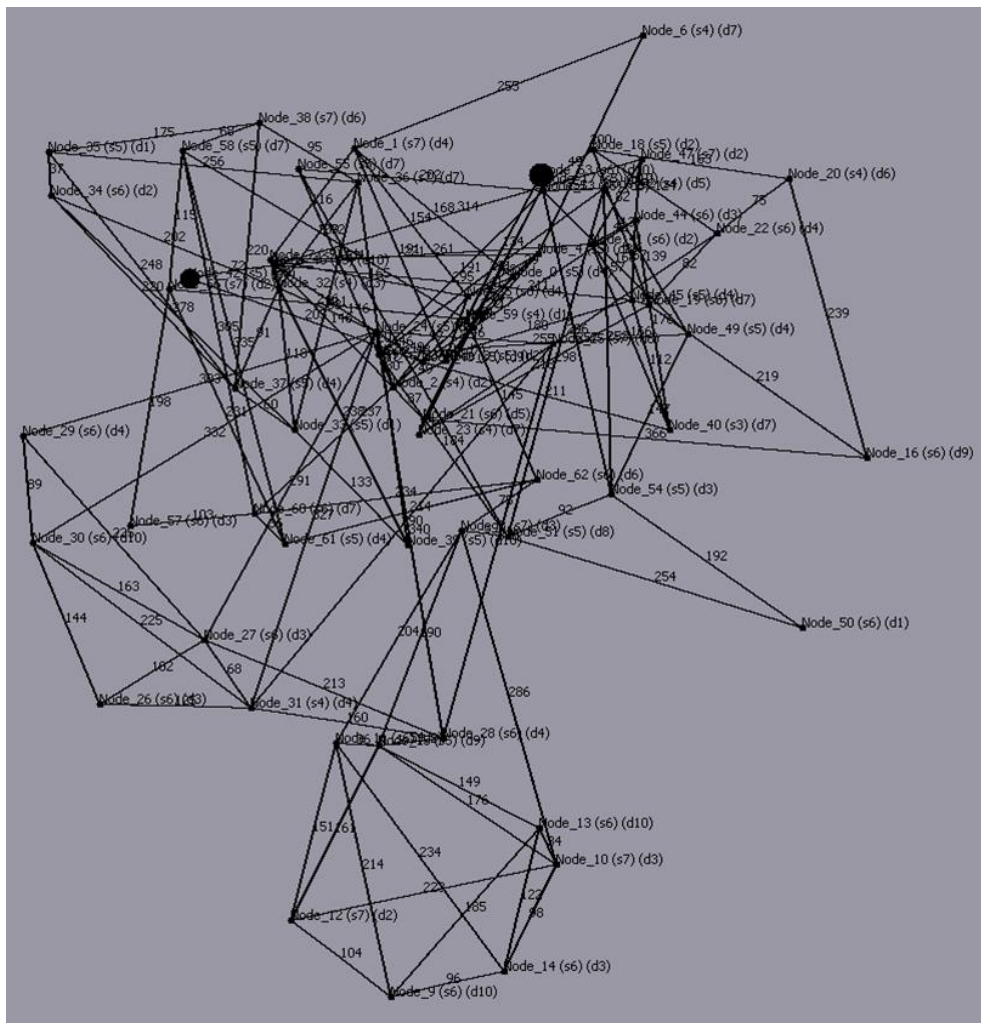
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.17-g, 6.17-h, and 6.17-i.



Appendix B Figure 7. Multi-Level SoS L with Node Status

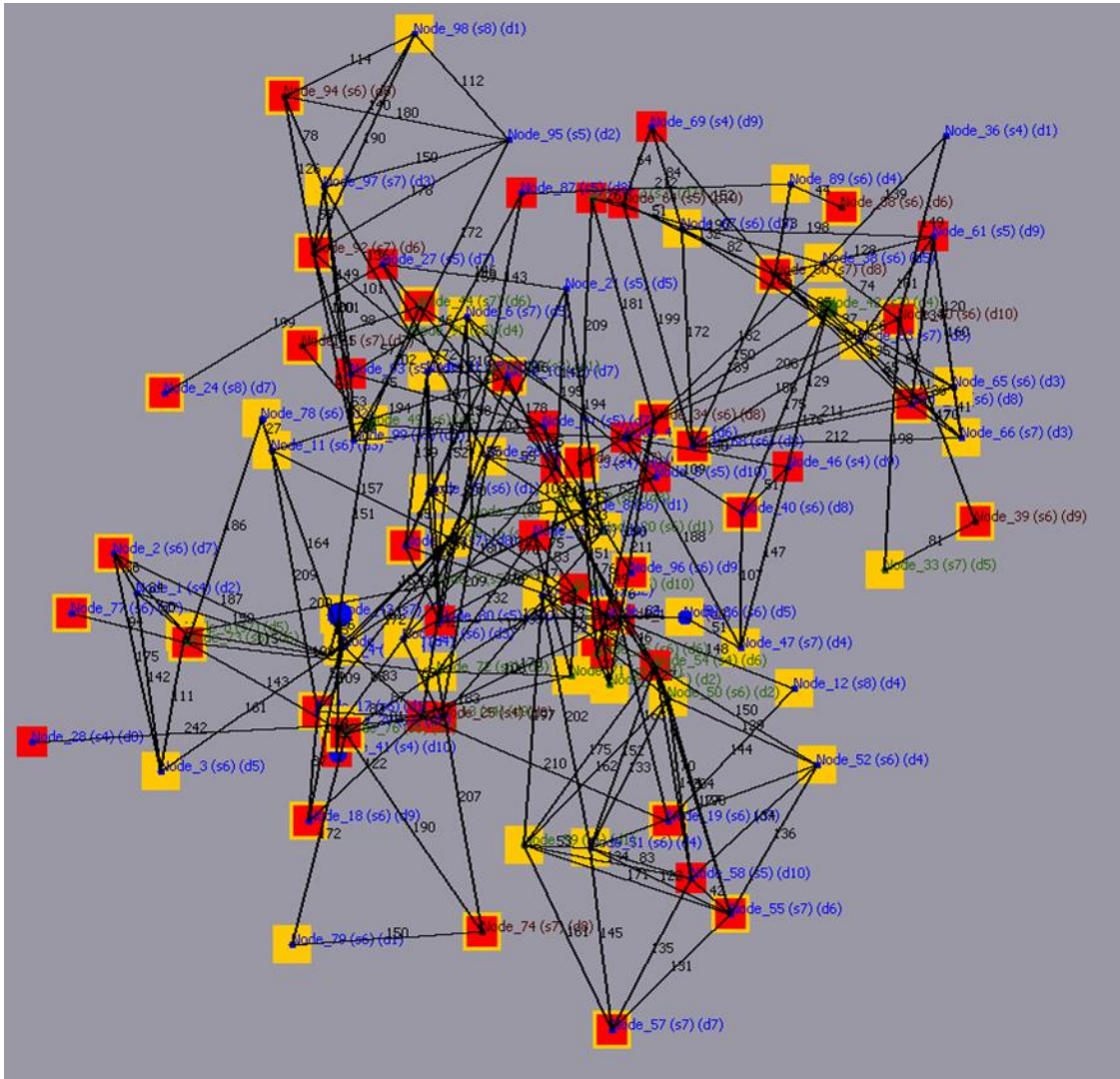


Appendix B Figure 8. Multi-Level SoS L Topology

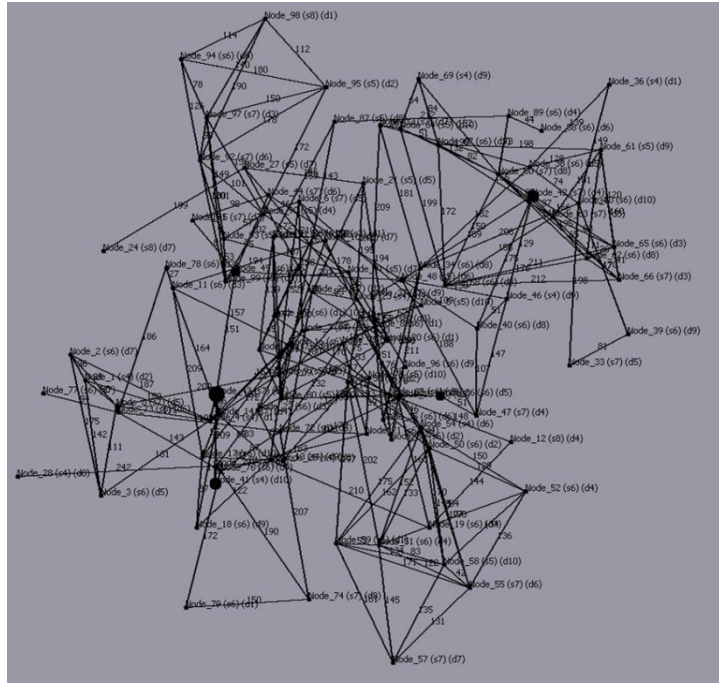


Appendix B Figure 9. Multi-Level SoS L Optimum Candidate

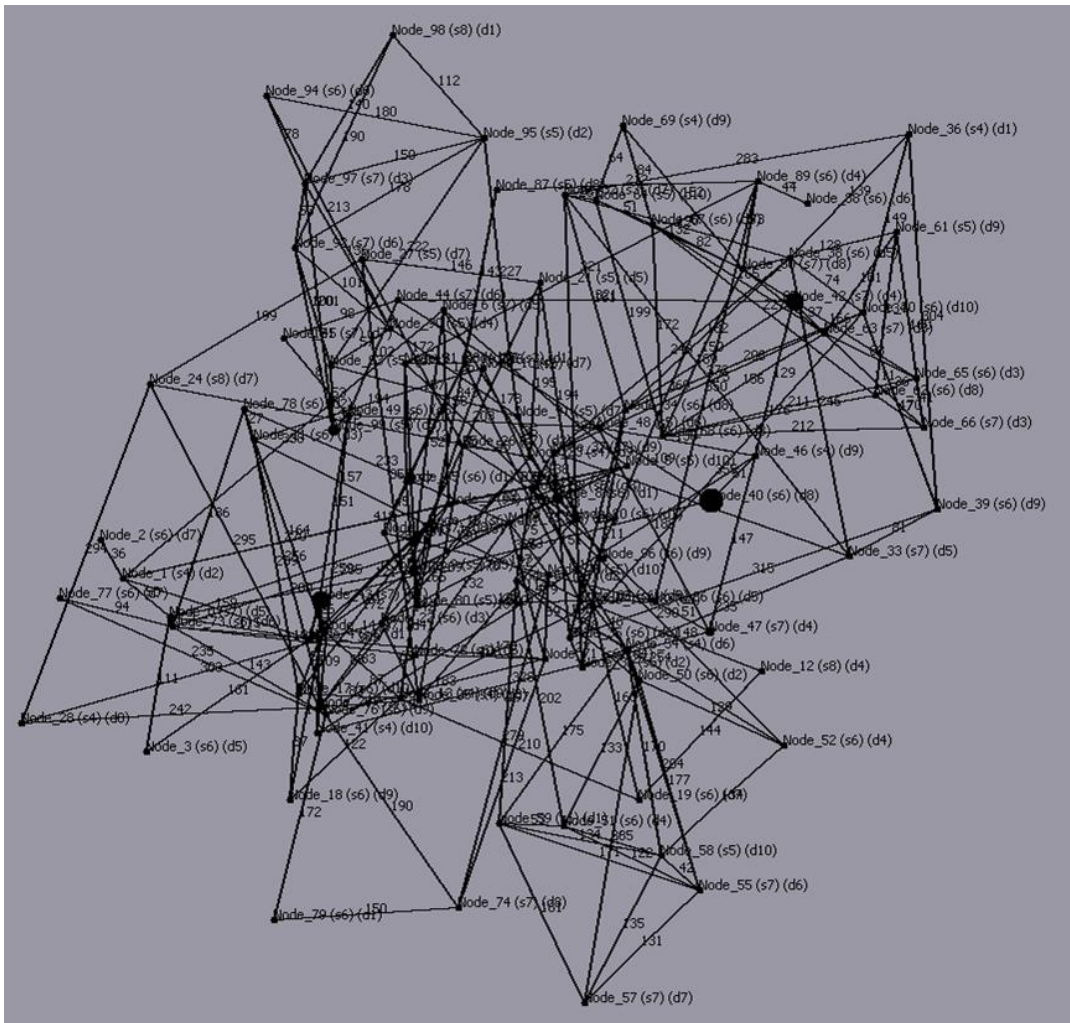
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.18-a, 6.18-b, and 6.18-c.



Appendix B Figure 10. Multi-Level SoS M with Node Status

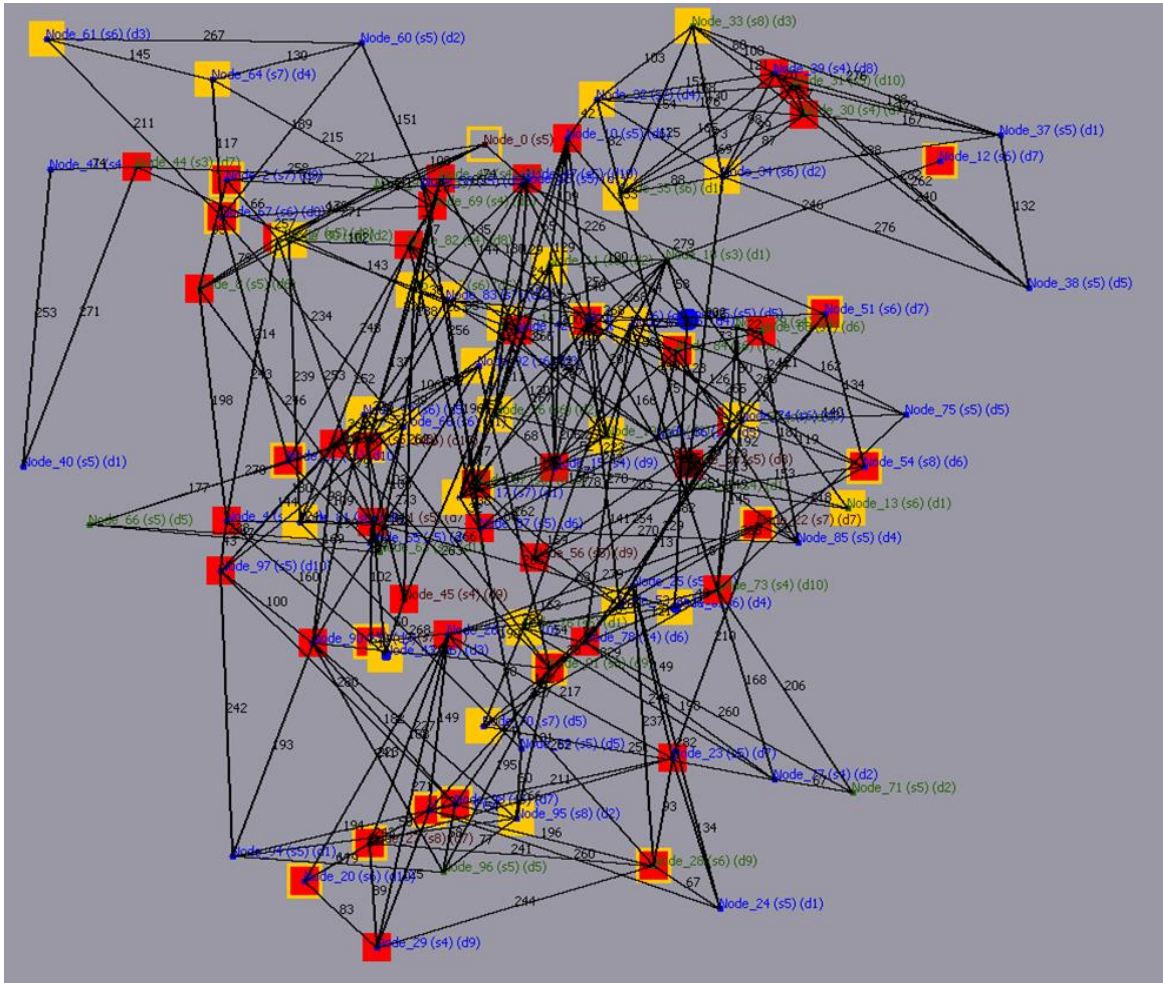


Appendix B Figure 11. Multi-Level SoS M Topology

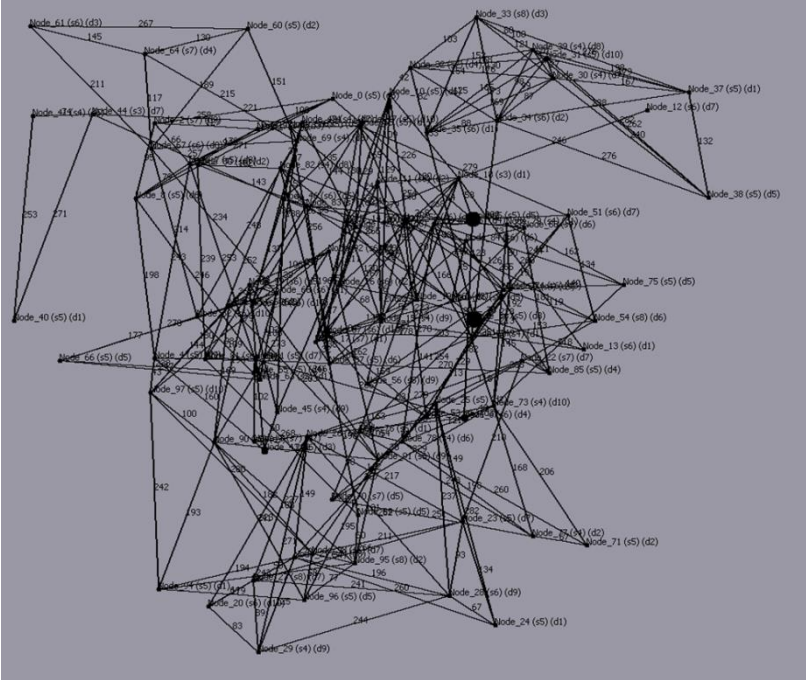


Appendix B Figure 12. Multi-Level SoS M Optimum Candidate

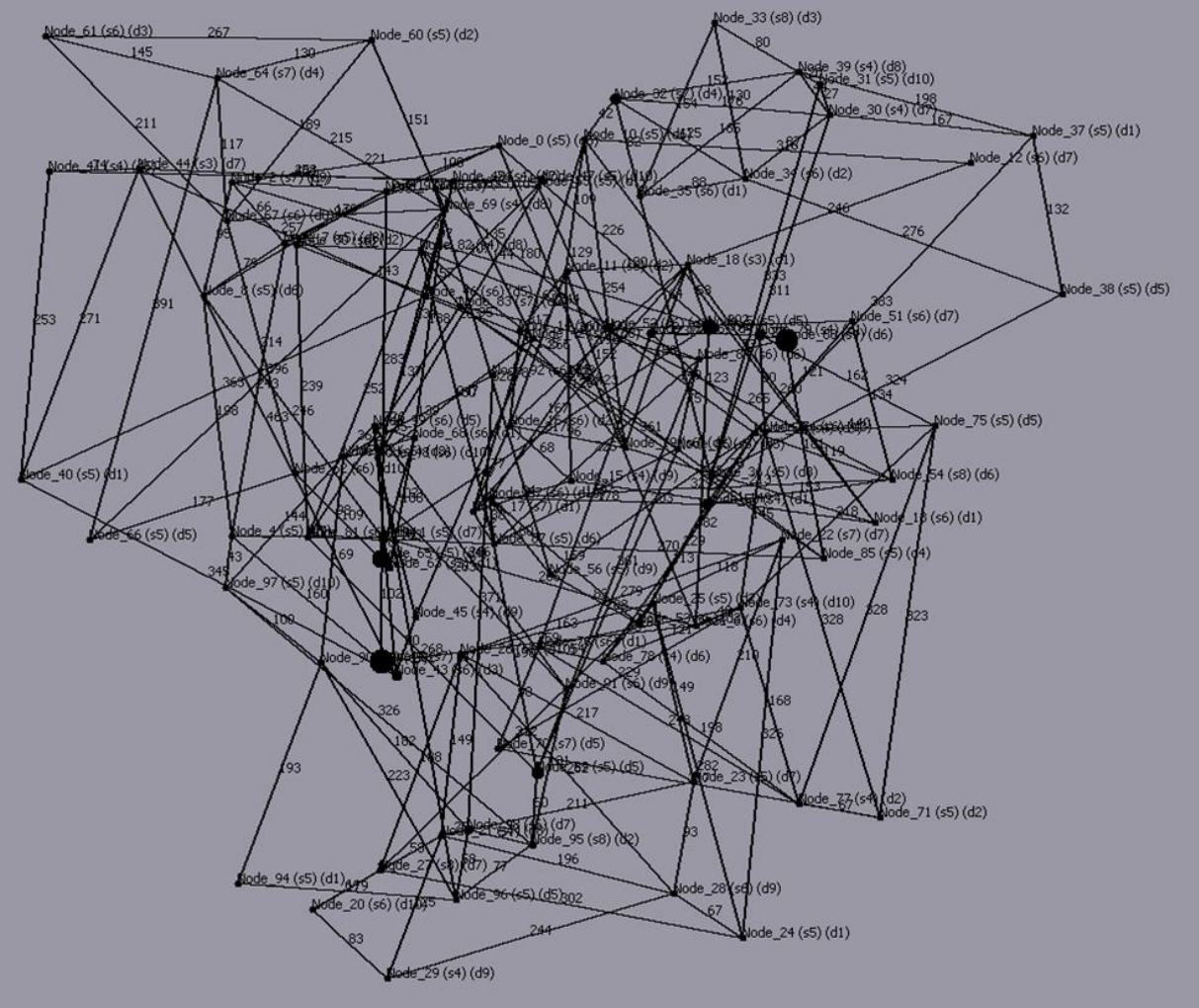
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.18-d, 6.18-e, and 6.18-f.



Appendix B Figure 13. Multi-Level SoS N with Node Status

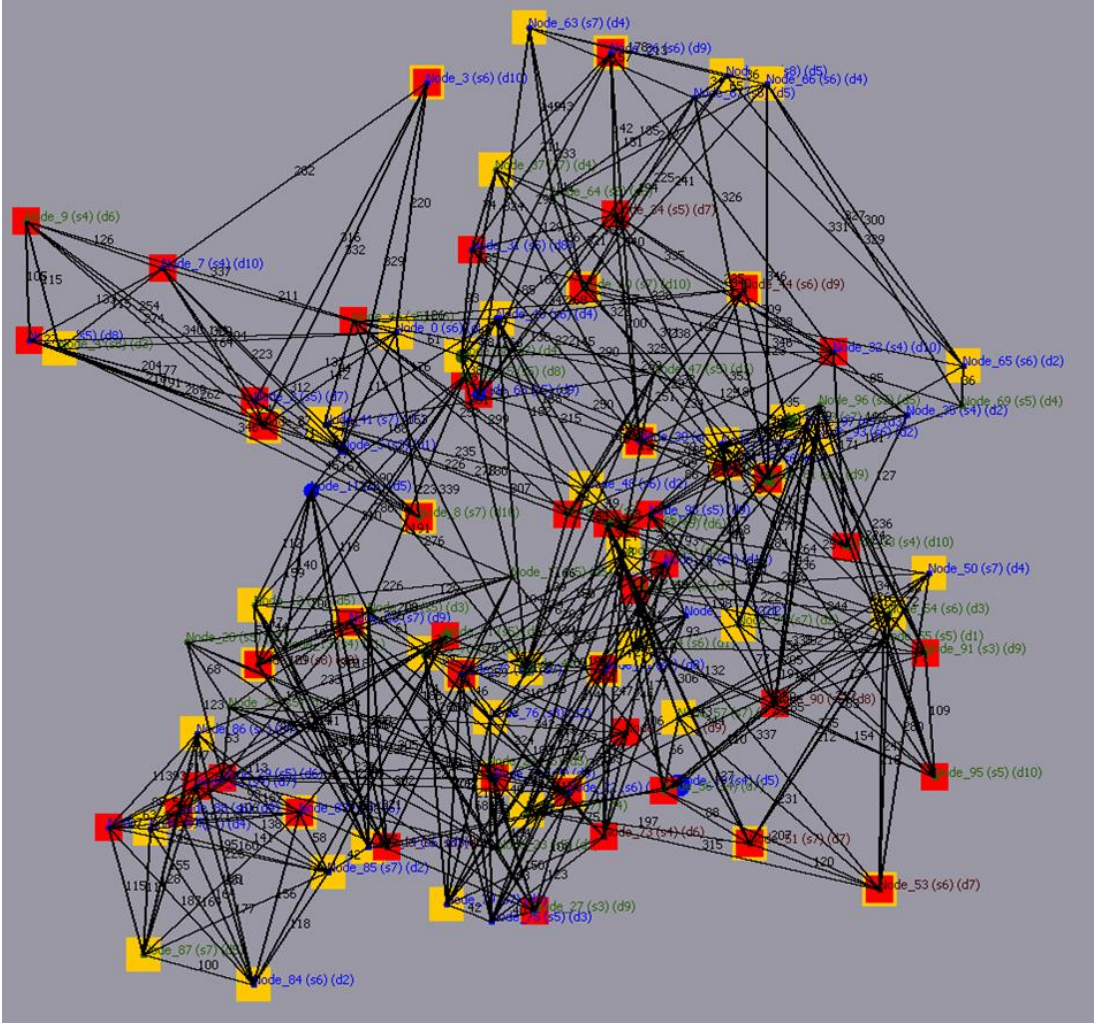


Appendix B Figure 14. Multi-Level SoS N Topology

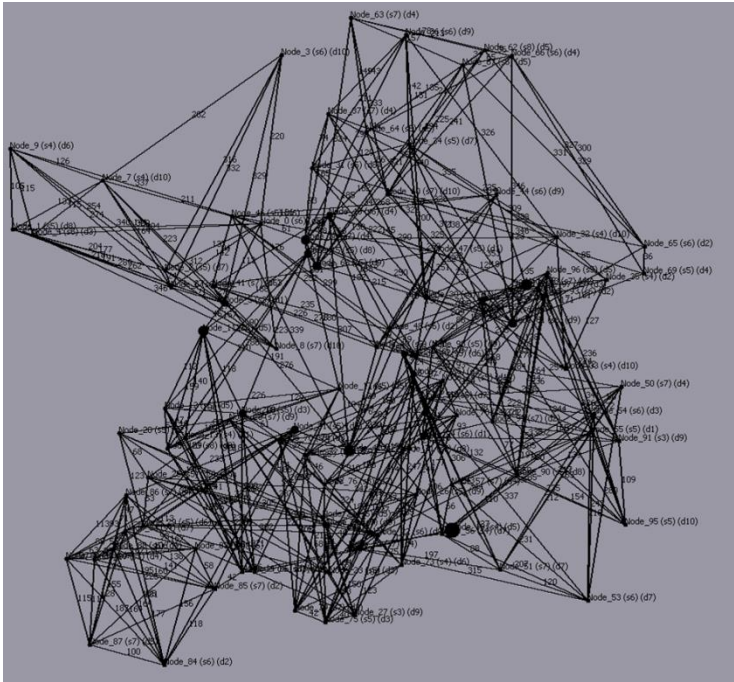


Appendix B Figure 15. Multi-Level SoS N Optimum Candidate

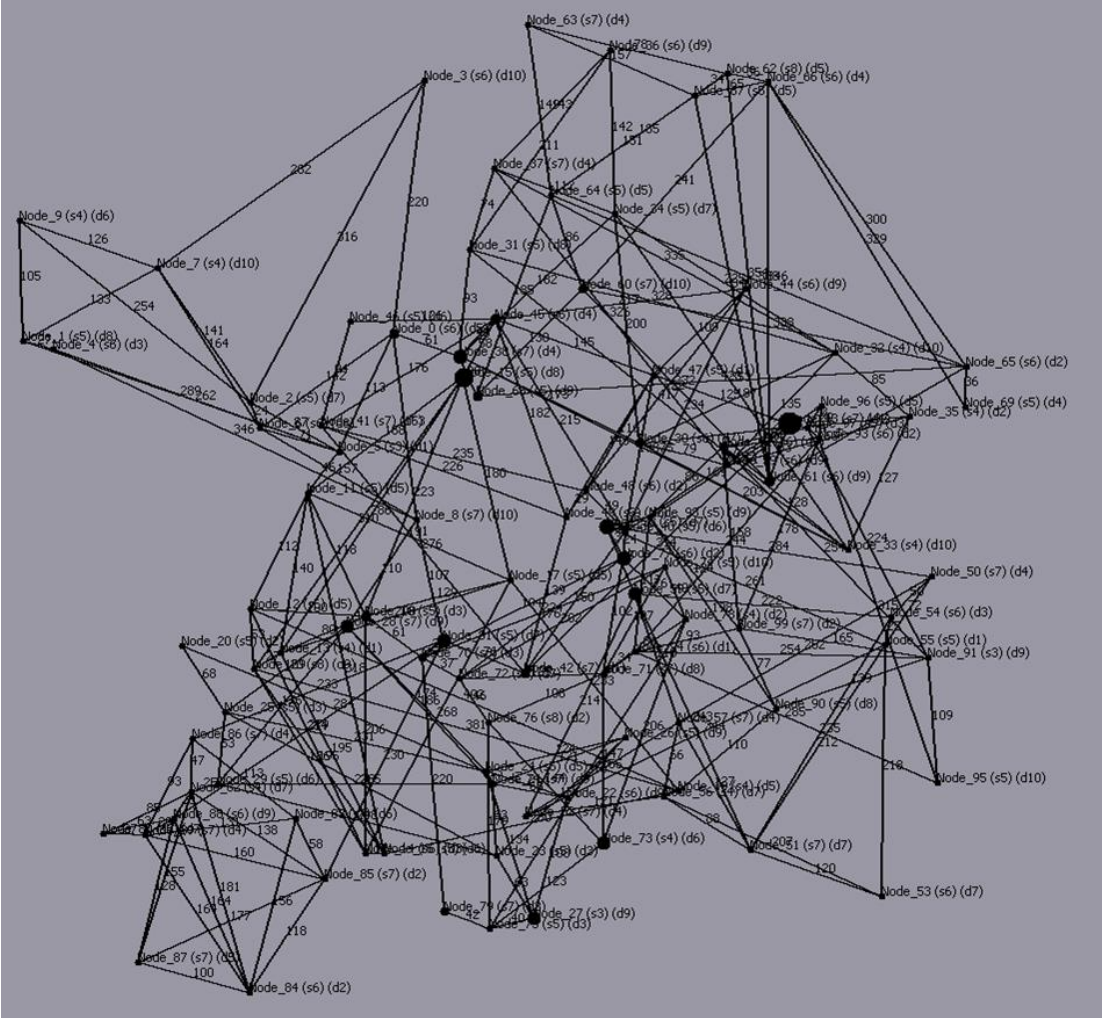
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.18-g, 6.18-h, and 6.18-i.



Appendix B Figure 16. Multi-Level SoS O with Node Status

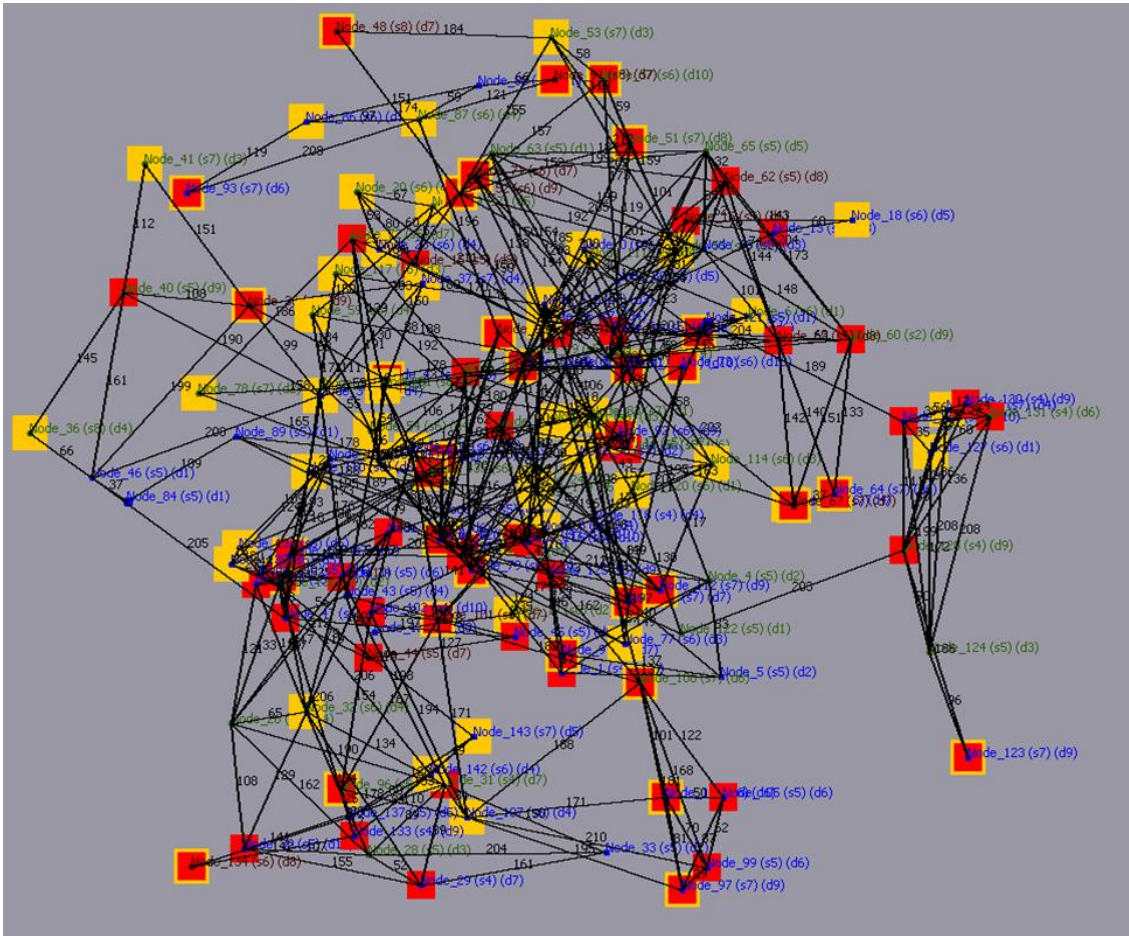


Appendix B Figure 17. Multi-Level SoS O Topology

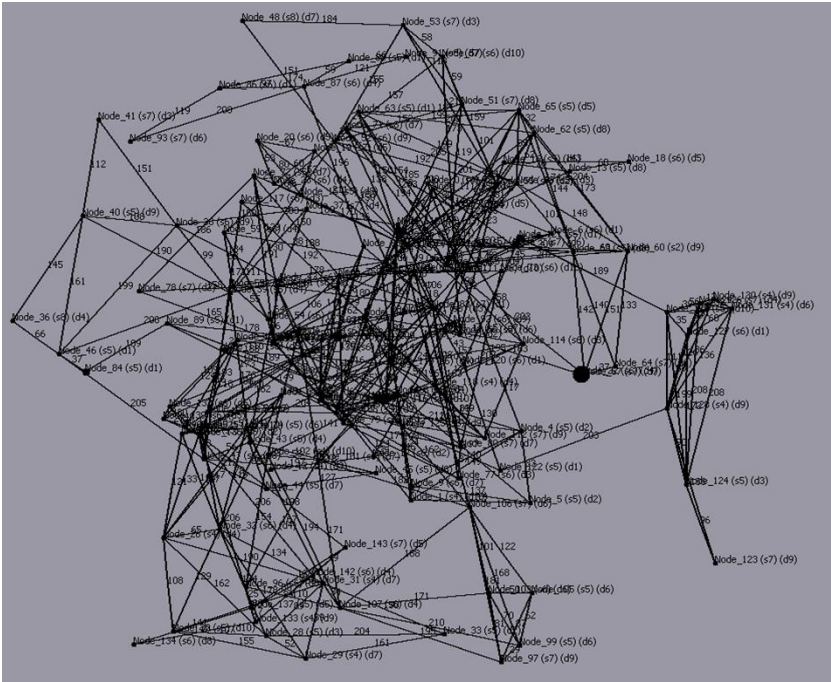


Appendix B Figure 18. Multi-Level SoS O Optimum Candidate

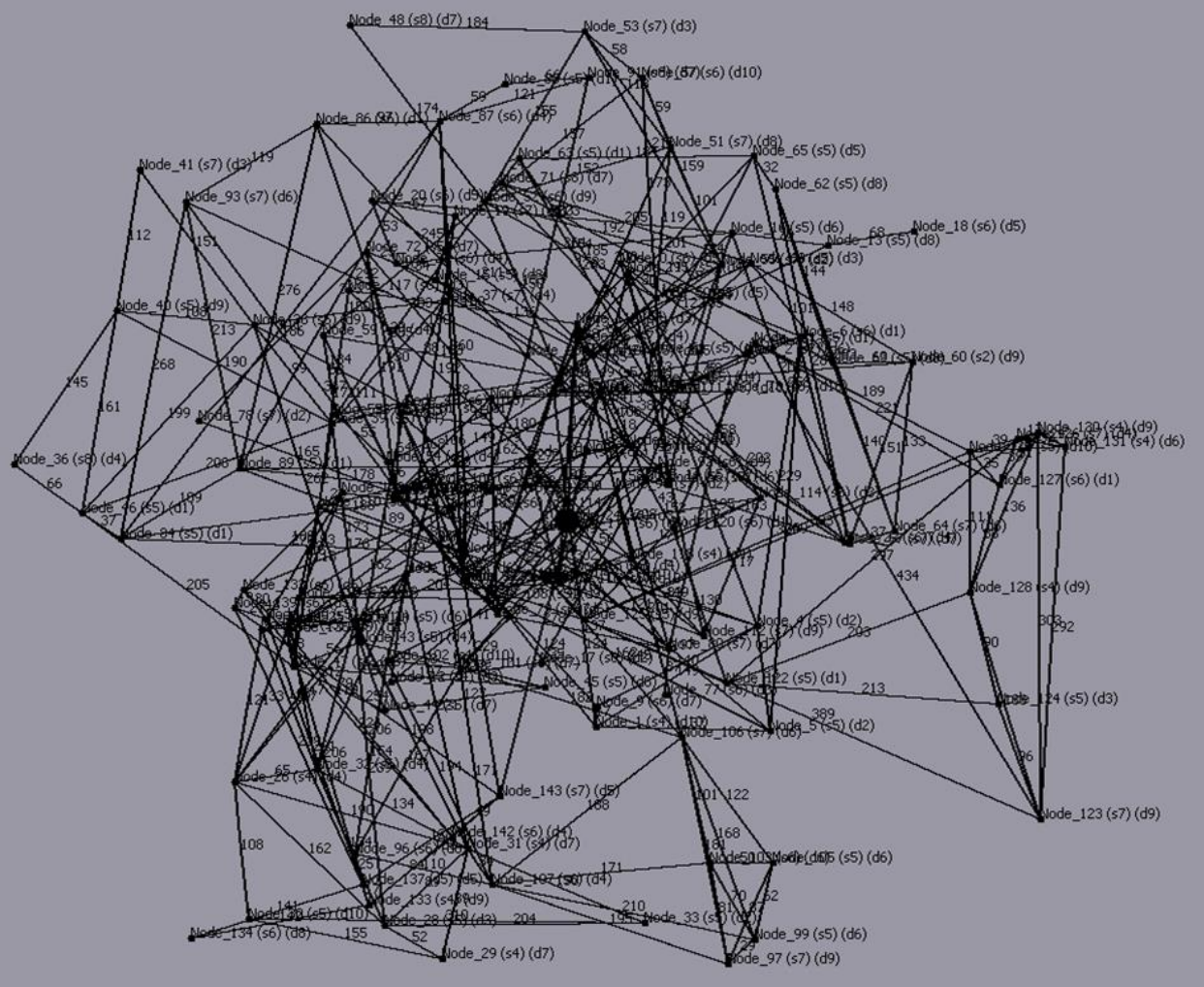
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.19-a, 6.19-b, and 6.19-c.



Appendix B Figure 19. Multi-Level SoS P with Node Status

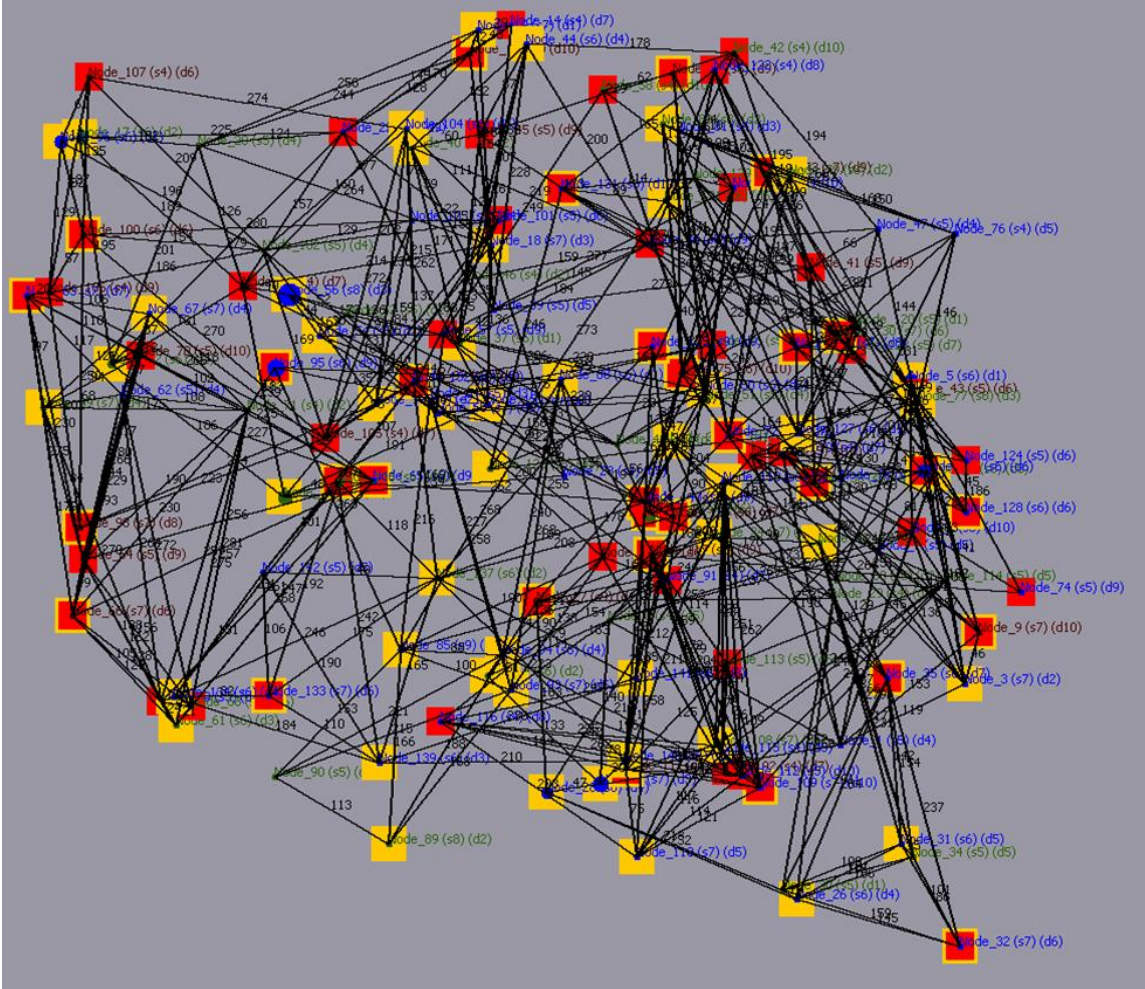


Appendix B Figure 20. Multi-Level SoS P Topology

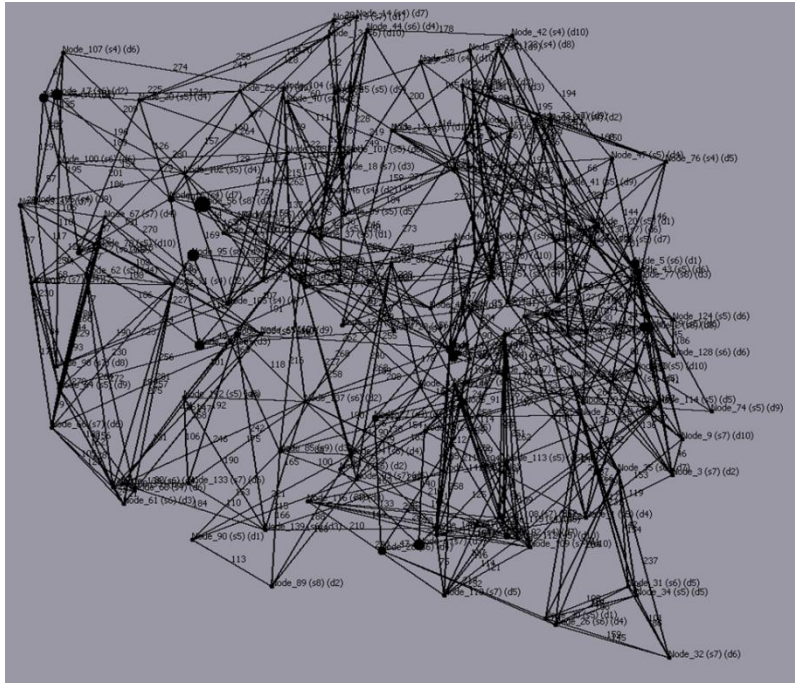


Appendix B Figure 21. Multi-Level SoS P Optimum Candidate

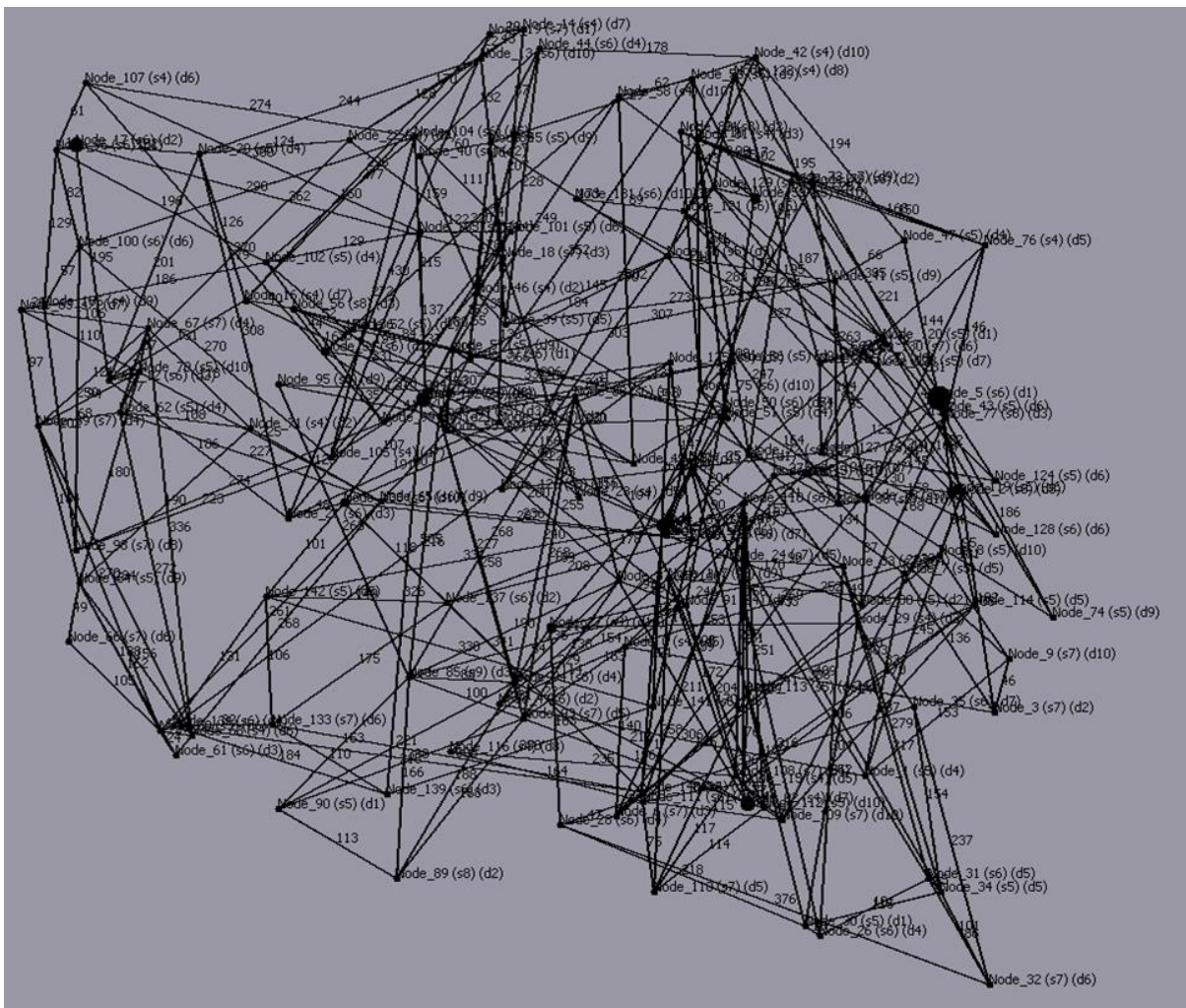
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.19-d, 6.19-e, and 6.19-f.



Appendix B Figure 22. Multi-Level SoS Q with Node Status

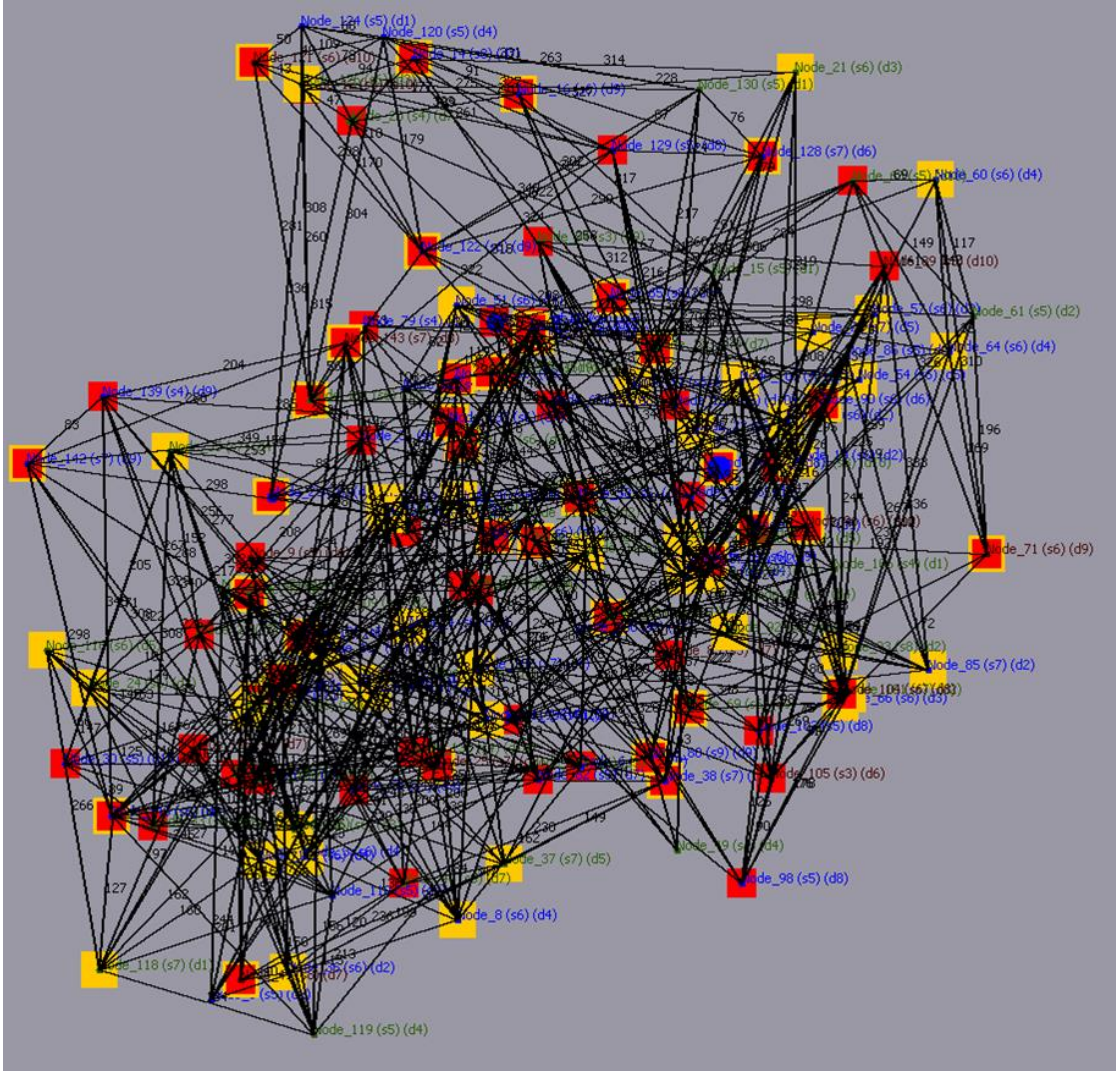


Appendix B Figure 23. Multi-Level SoS Q Topology

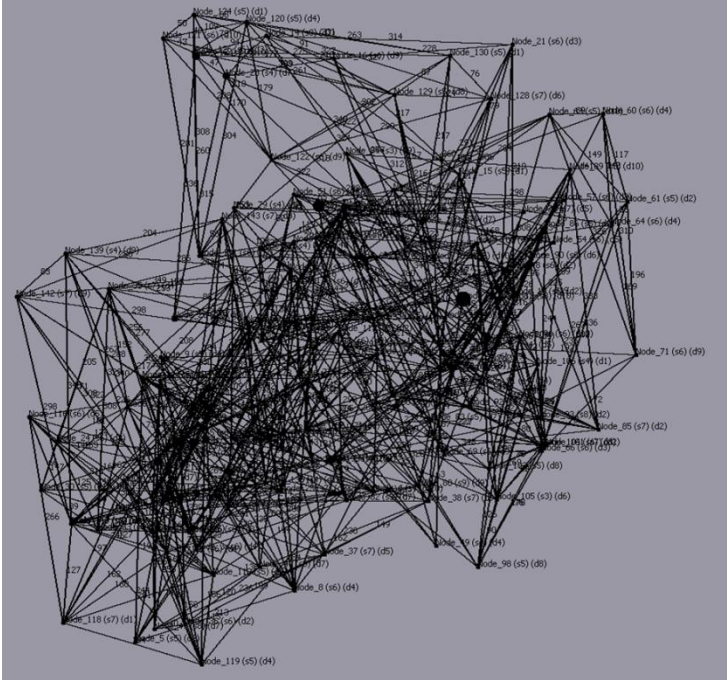


Appendix B Figure 24. Multi-Level SoS Q Optimum Candidate

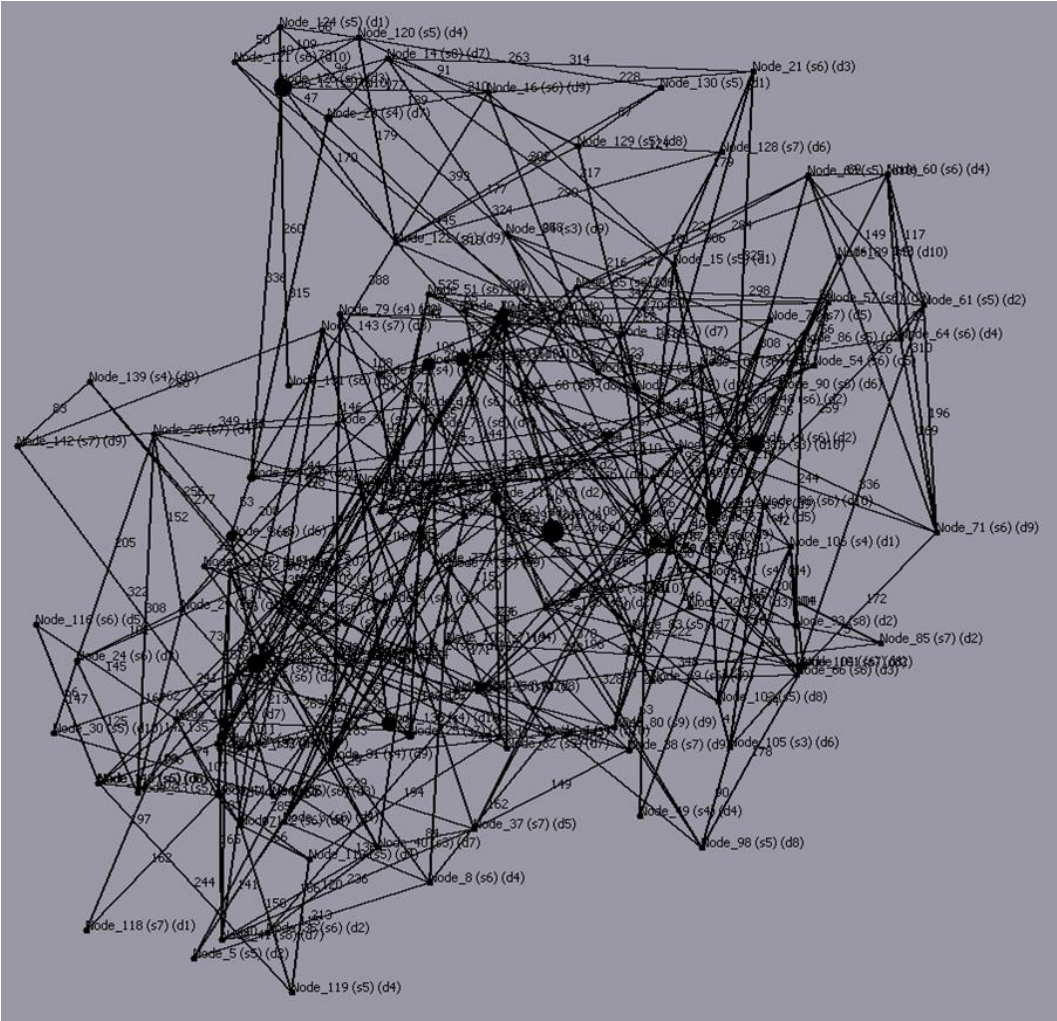
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.19-g, 6.19-h, and 6.19-i.



Appendix B Figure 25. Multi-Level SoS R with Node Status



Appendix B Figure 26. Multi-Level SoS R Topology



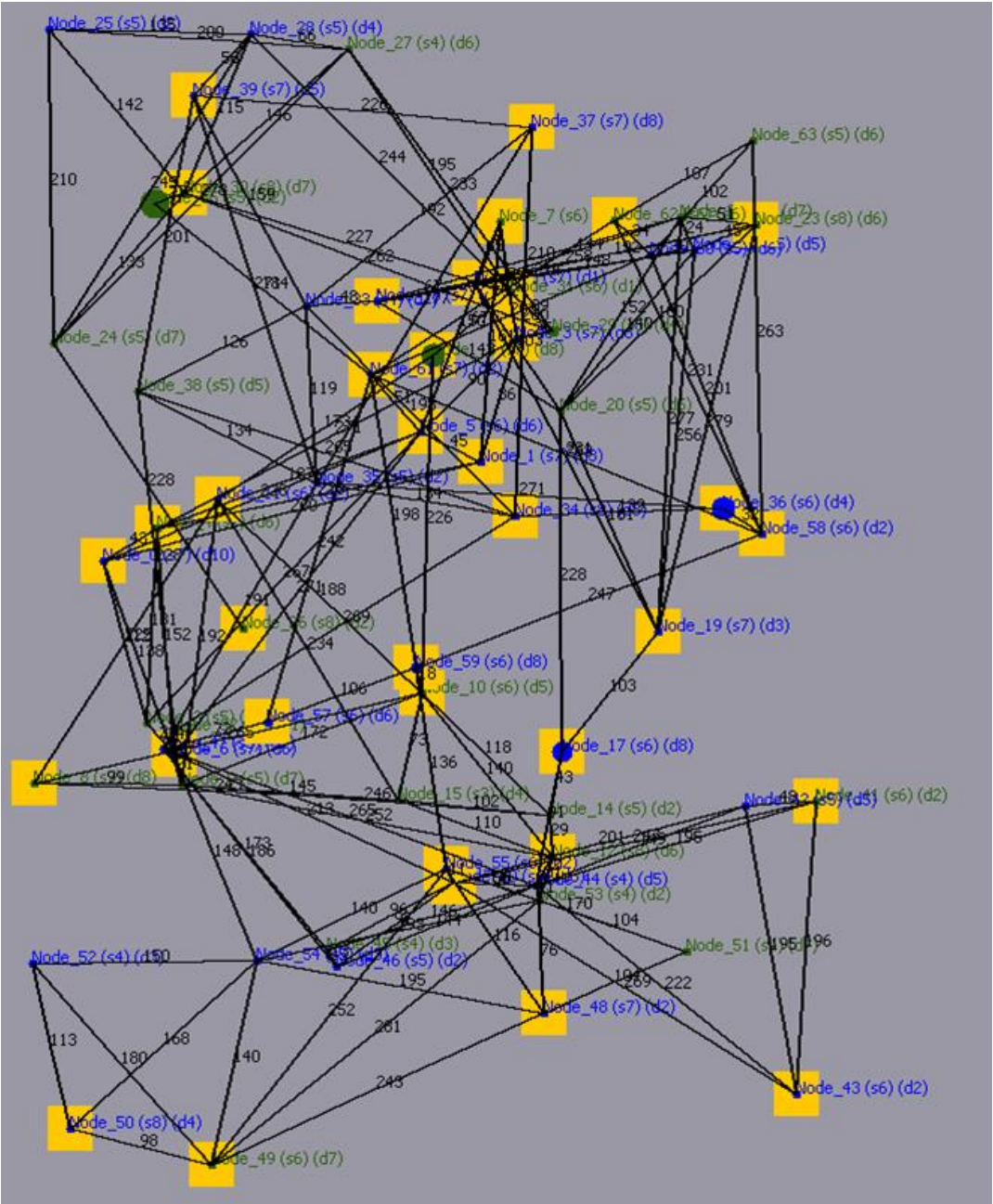
Appendix B Figure 27. Multi-Level SoS R Optimum Candidate

Appendix C

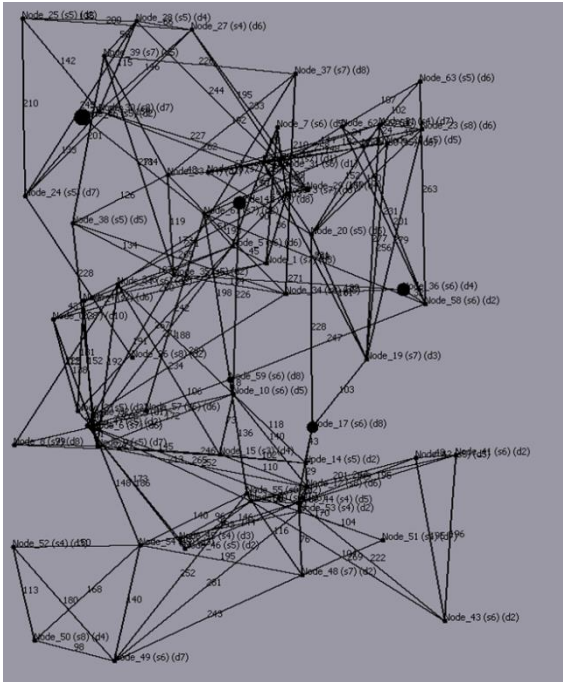
SCRAM Negative Multi-Level SoS

Vulnerability Performance

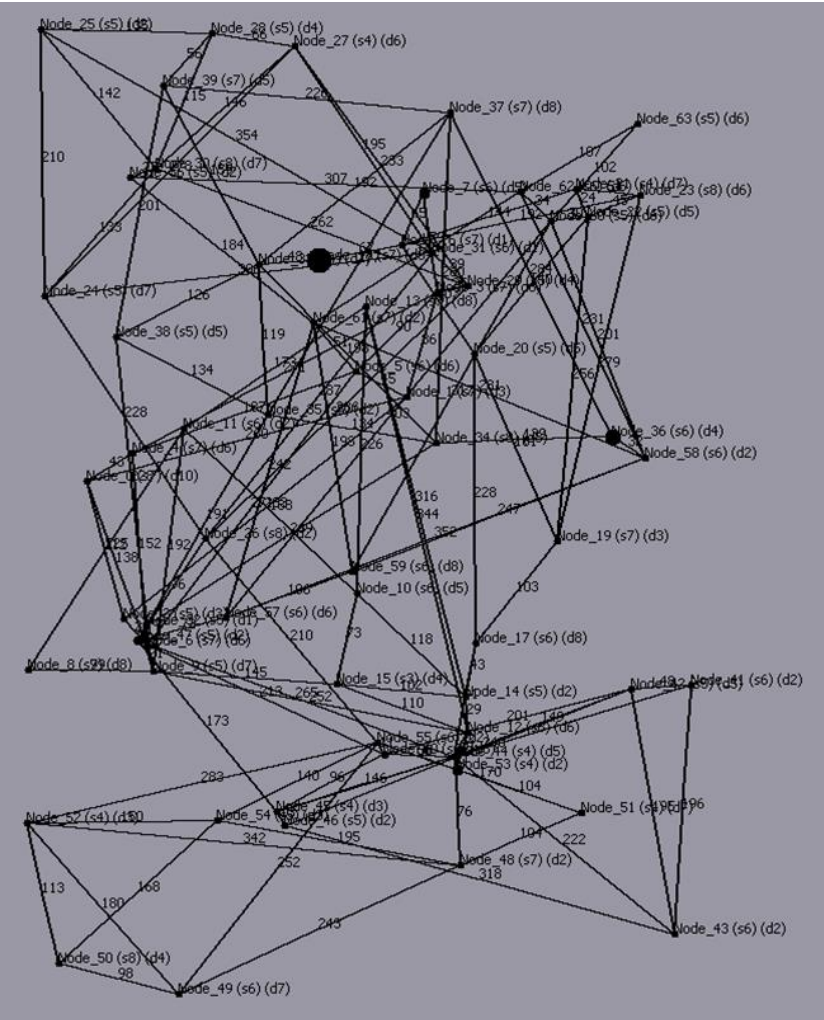
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.23-a, 6.23-b, and 6.23-c.



Appendix C Figure 1. Multi-Level SoS S with Node Status

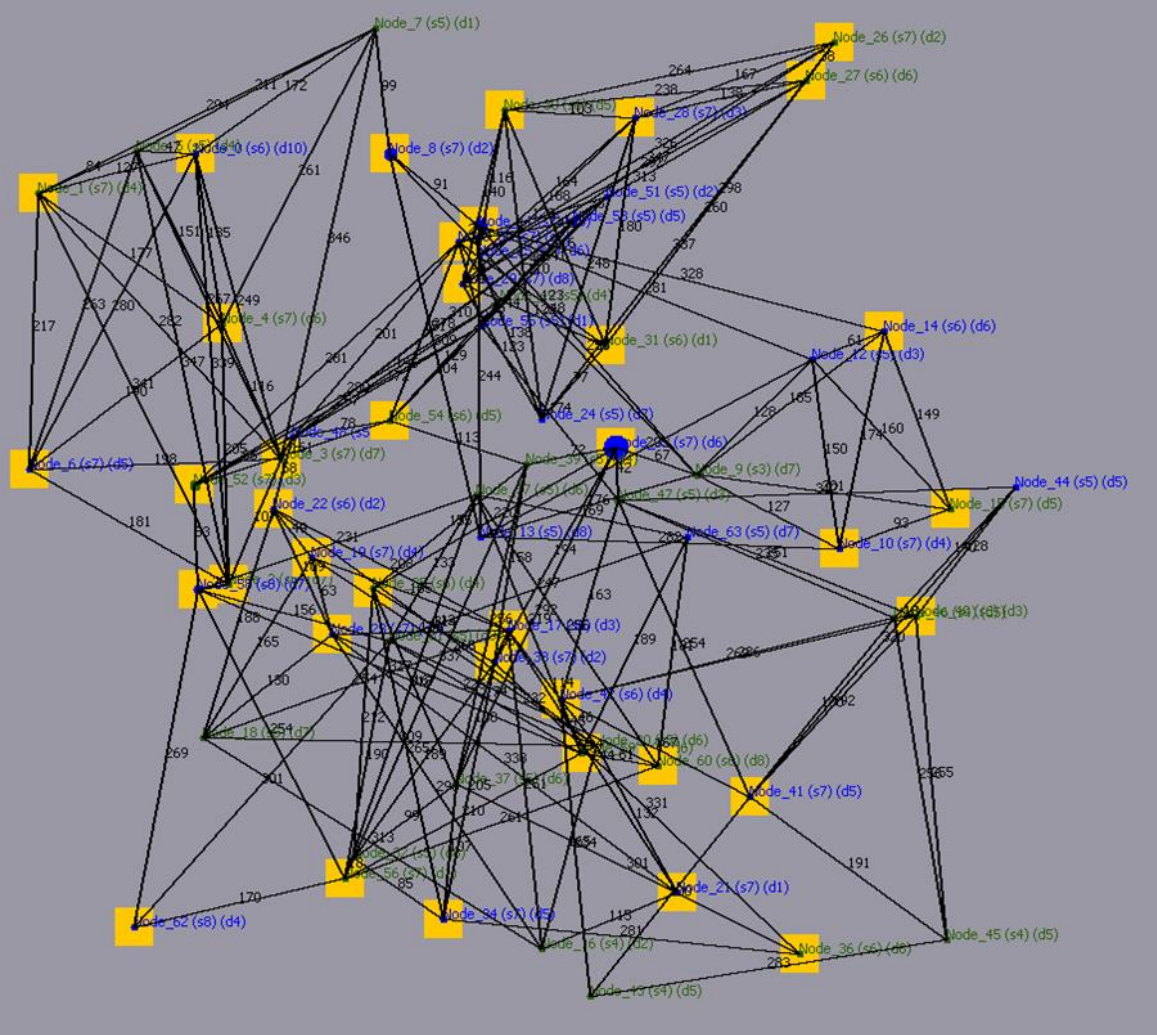


Appendix C Figure 2. Multi-Level SoS S Topology

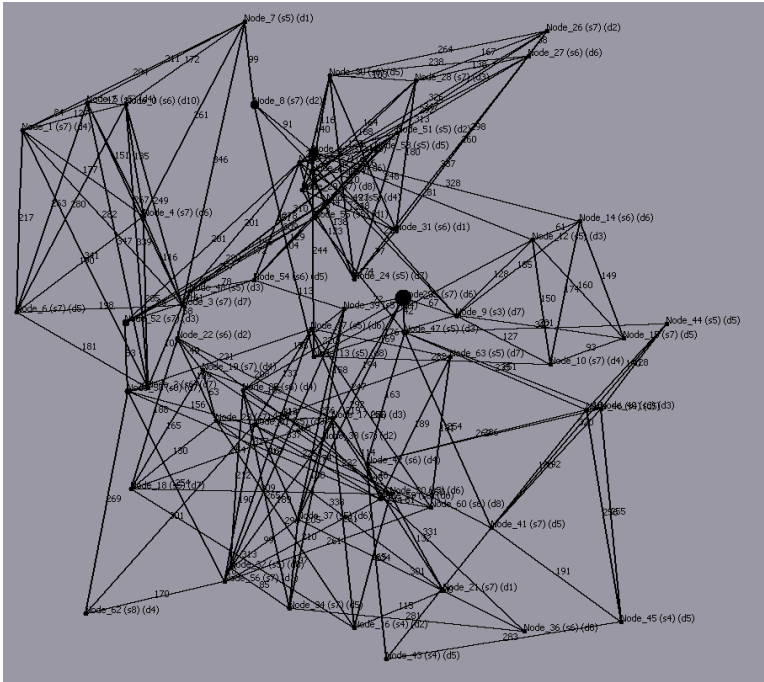


Appendix C Figure 3. Multi-Level SoS S Optimum Candidate

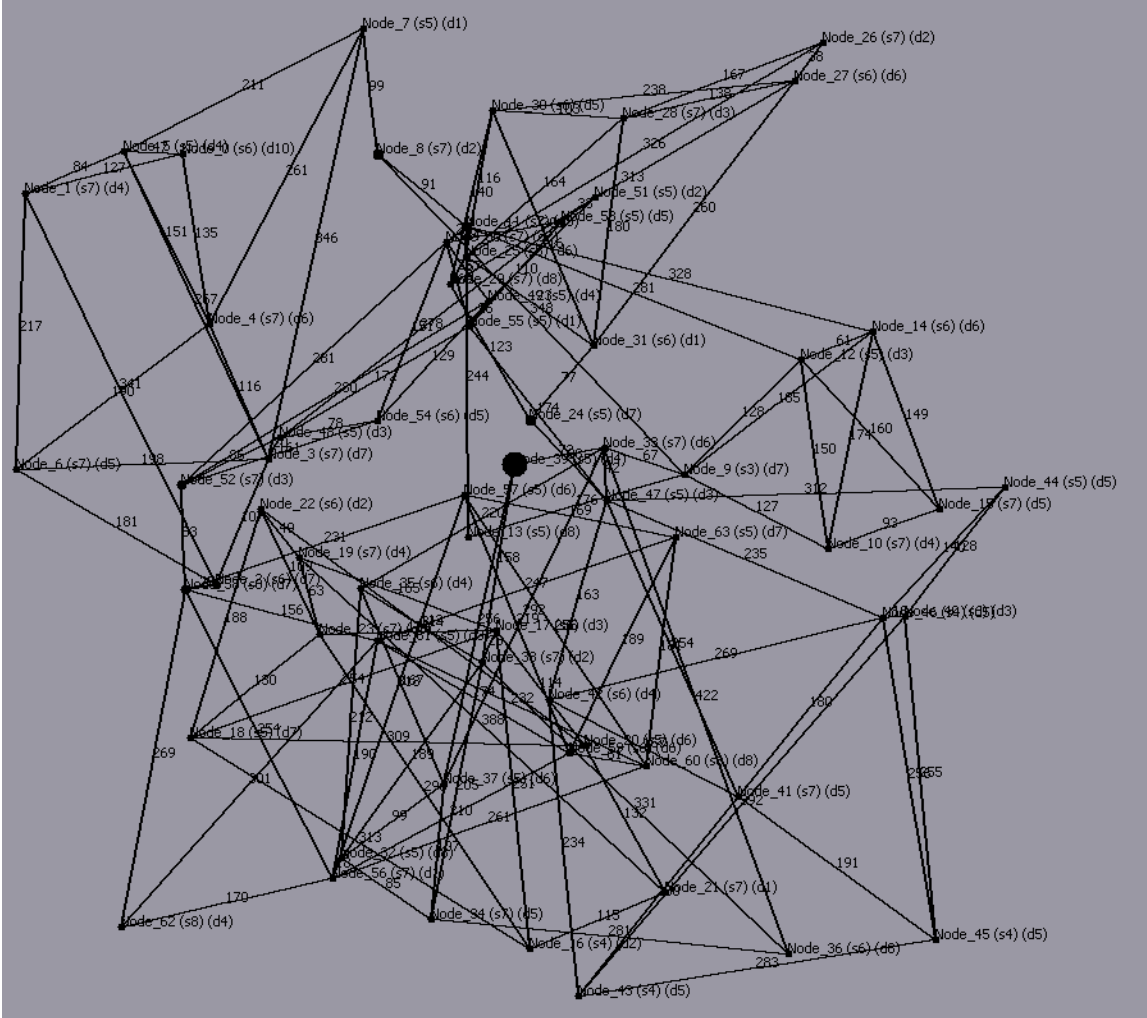
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.23-d, 6.23-e, and 6.23-f.



Appendix C Figure 4. Multi-Level SoS T with Node Status

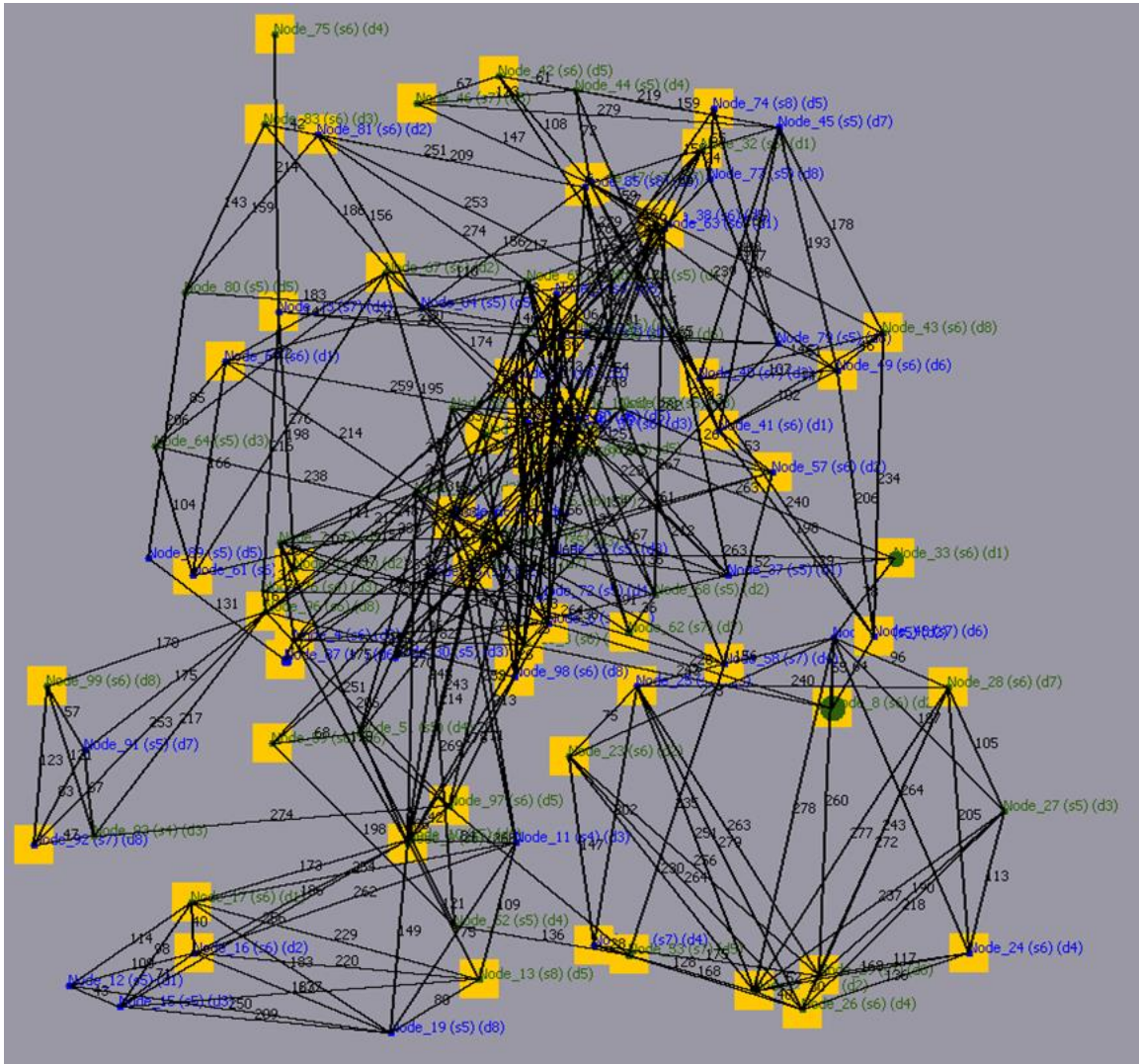


Appendix C Figure 5. Multi-Level SoS T Topology

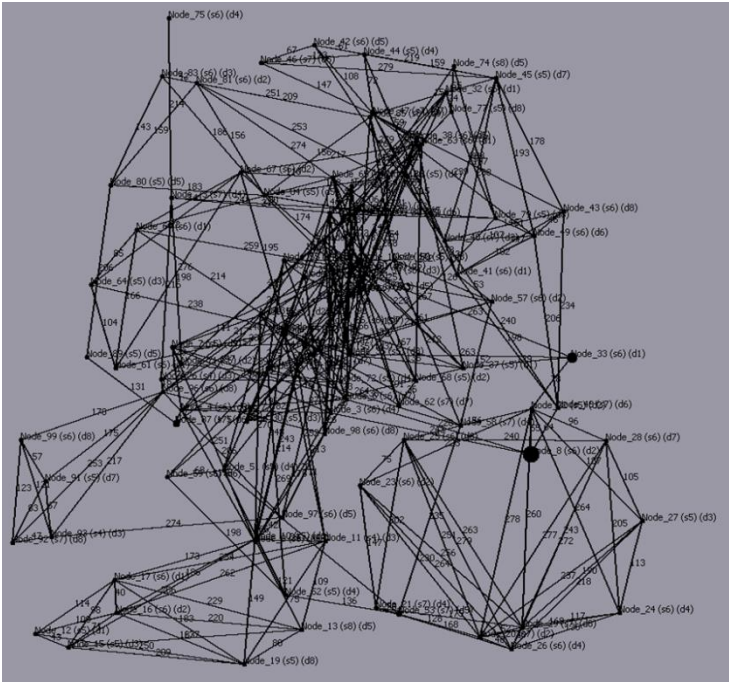


Appendix C Figure 6. Multi-Level SoS T Optimum Candidate

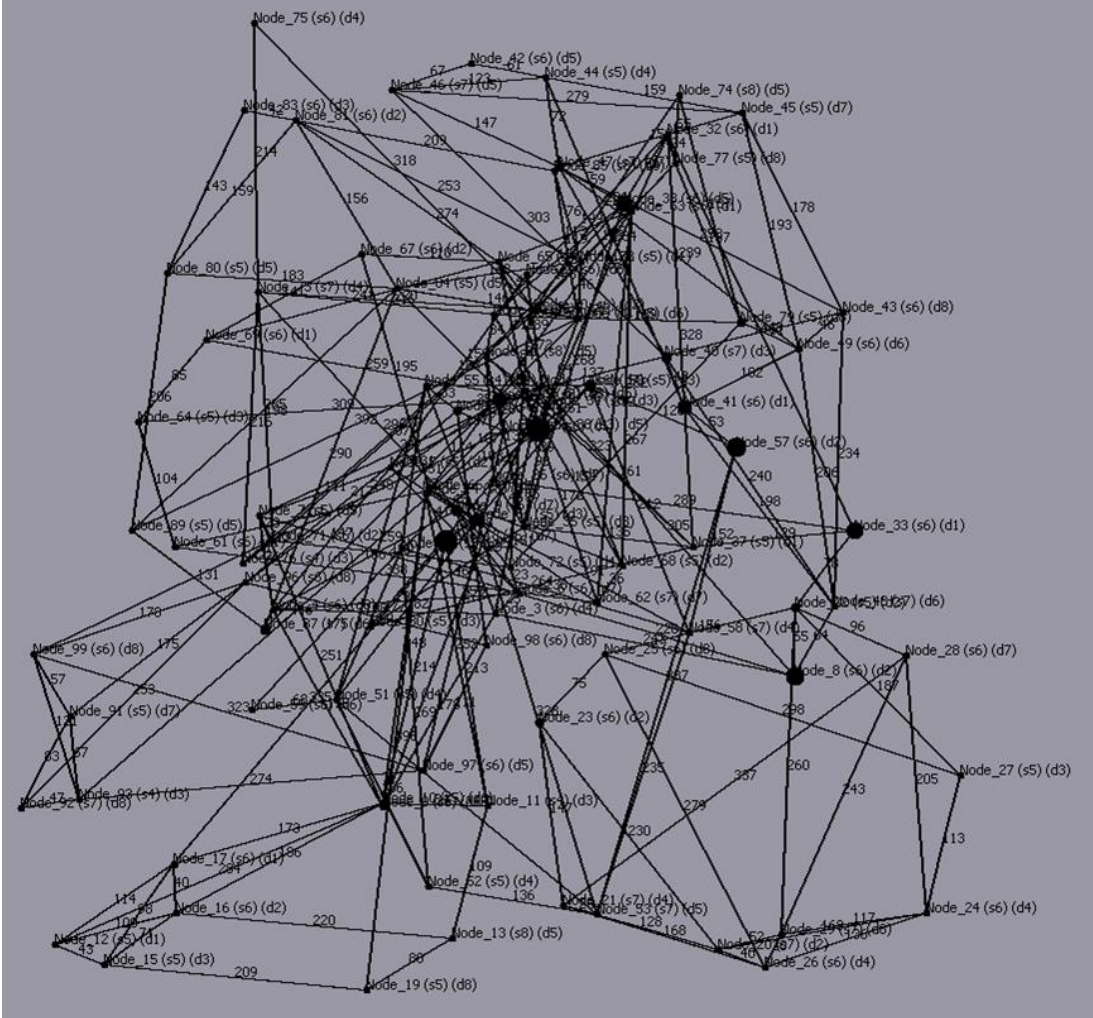
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.24-a, 6.24-b, and 6.24-c.



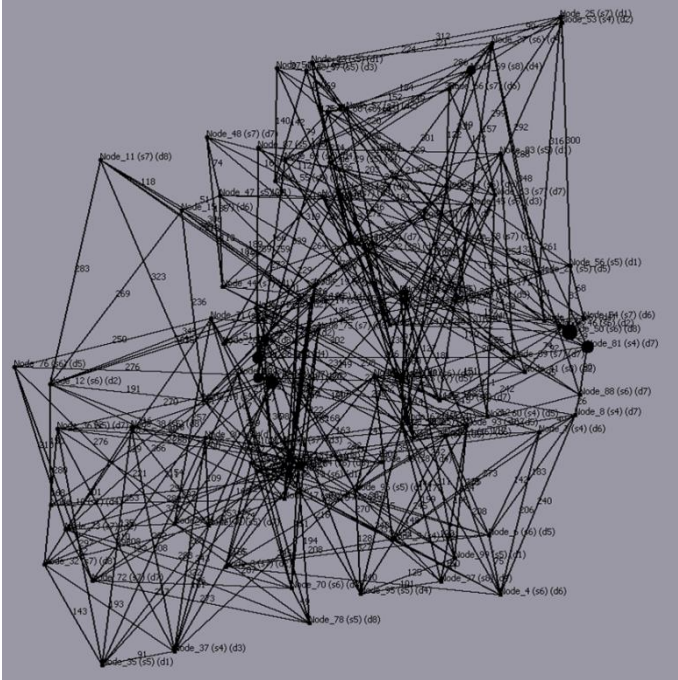
Appendix C Figure 7. Multi-Level SoS U with Node Status



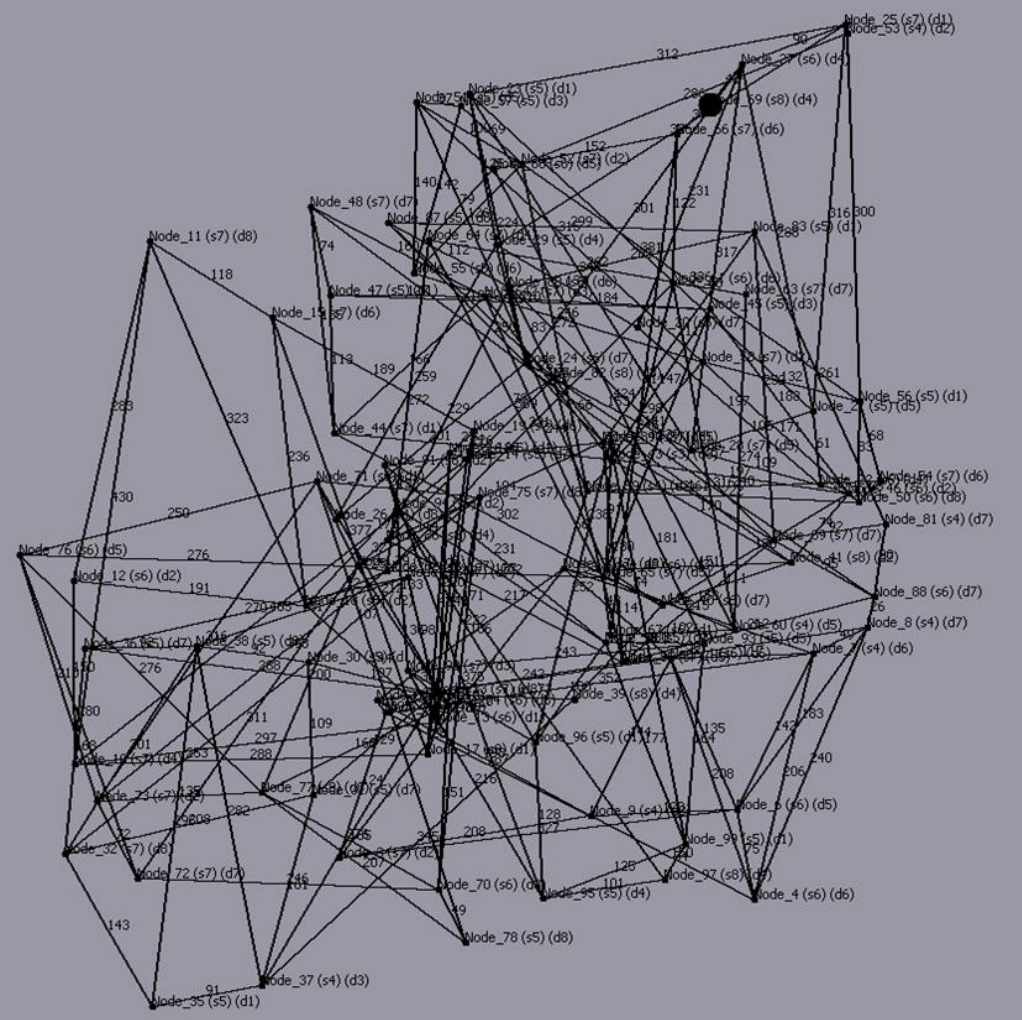
Appendix C Figure 8. Multi-Level SoS U Topology



Appendix C Figure 9. Multi-Level SoS U Optimum Candidate

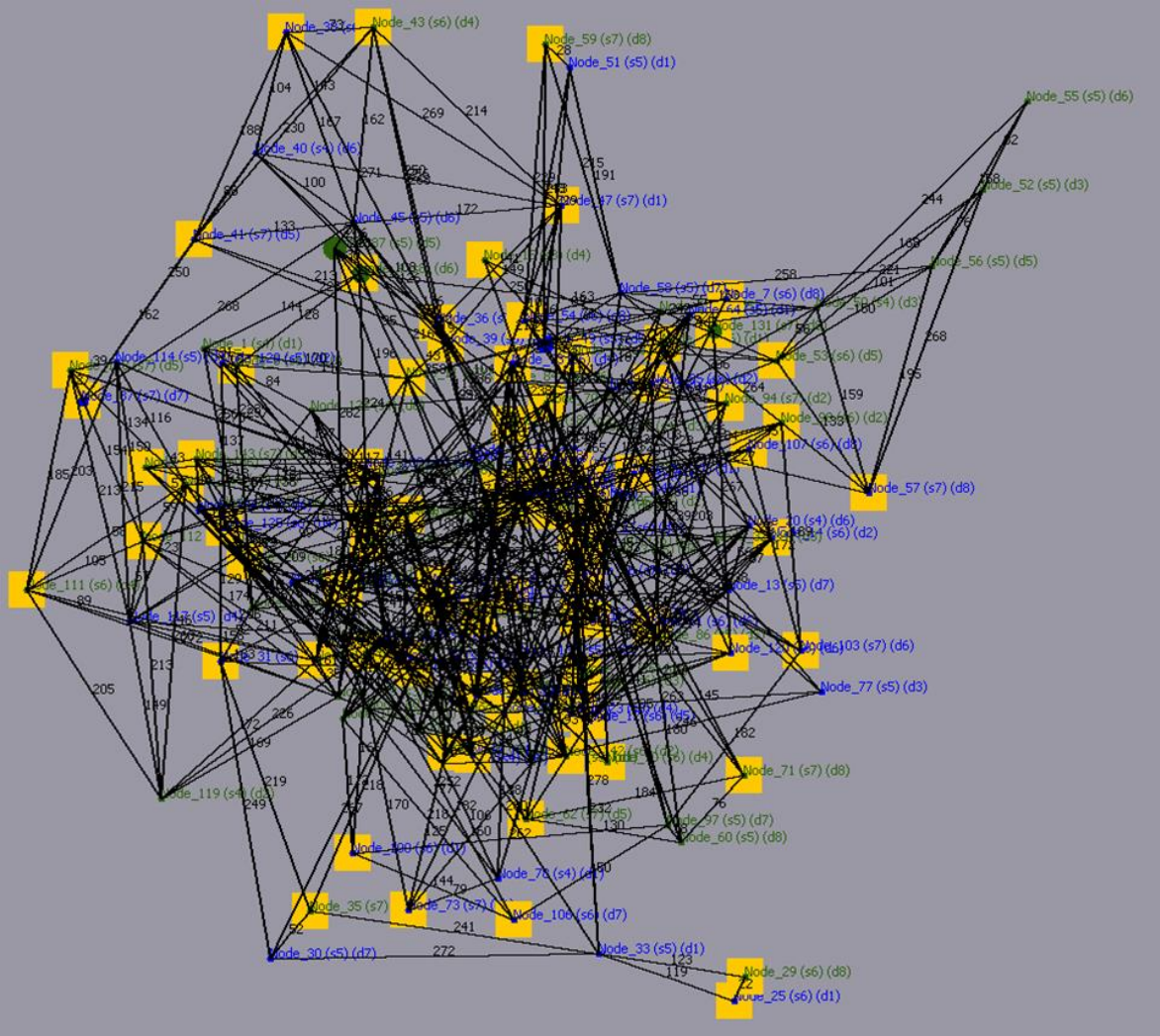


Appendix C Figure 11. Multi-Level SoS V Topology

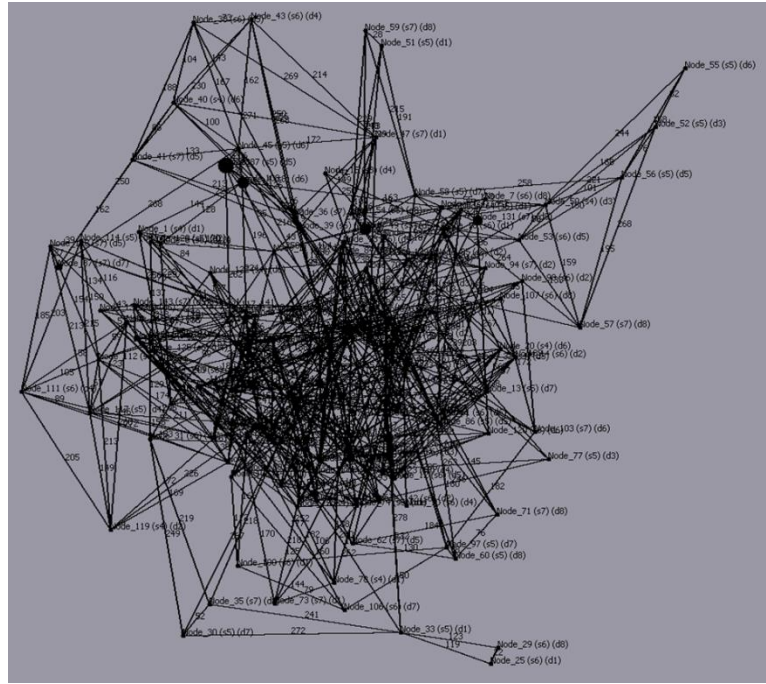


Appendix C Figure 12. Multi-Level SoS V Optimum Candidate

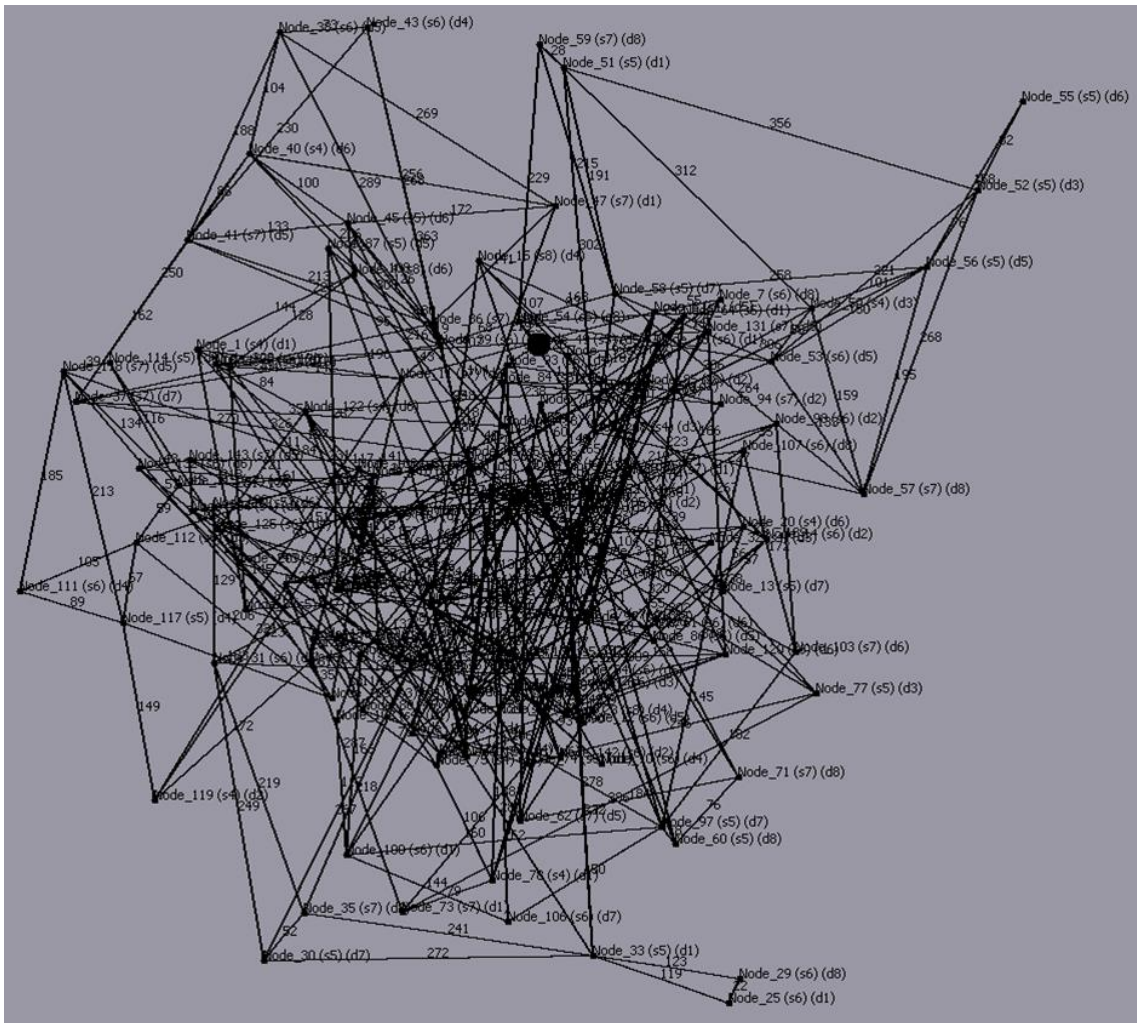
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.25-a, 6.25-b, and 6.25-c.



Appendix C Figure 13. Multi-Level SoS W with Node Status



Appendix C Figure 14. Multi-Level SoS W Topology



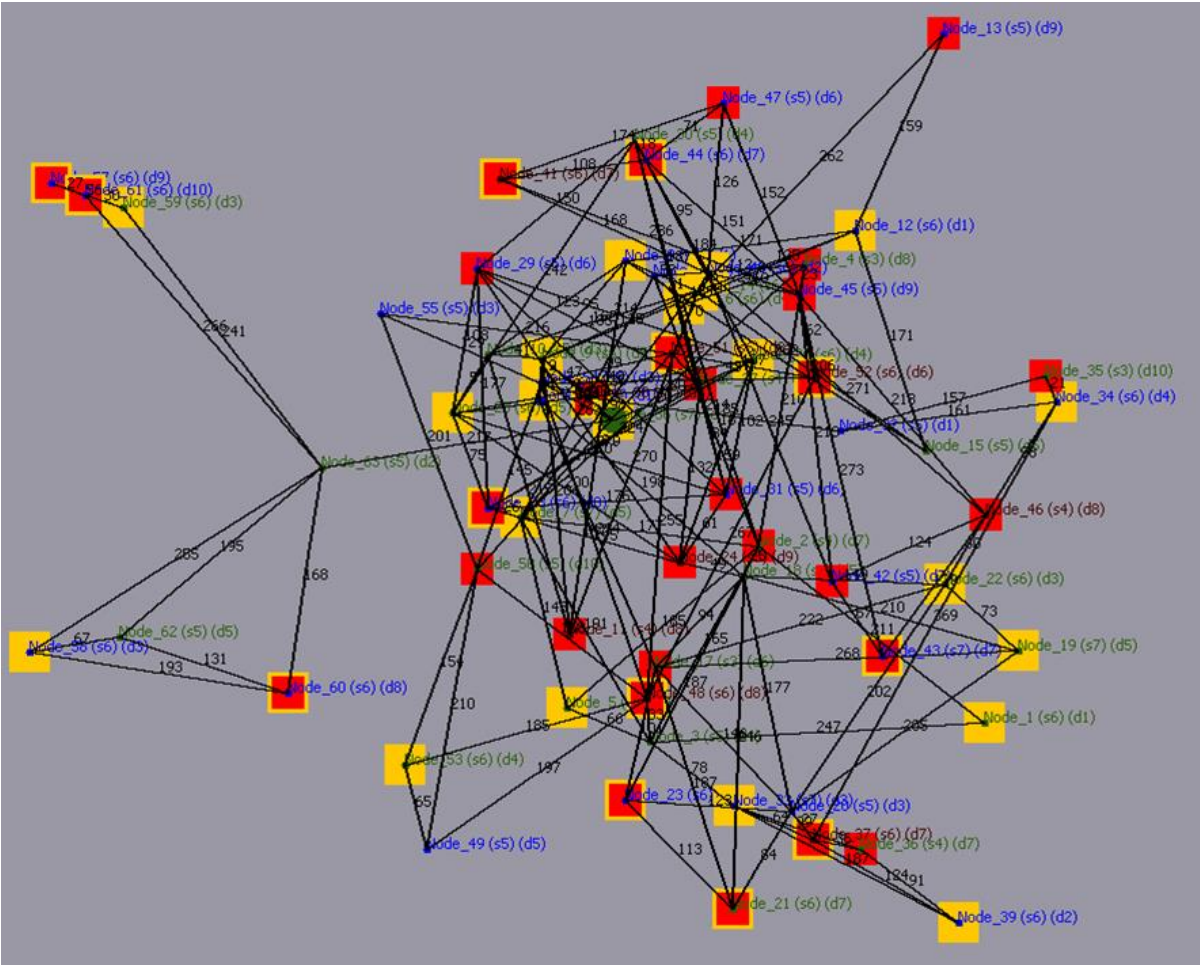
Appendix C Figure 15. Multi-Level SoS W Optimum Candidate

Appendix D

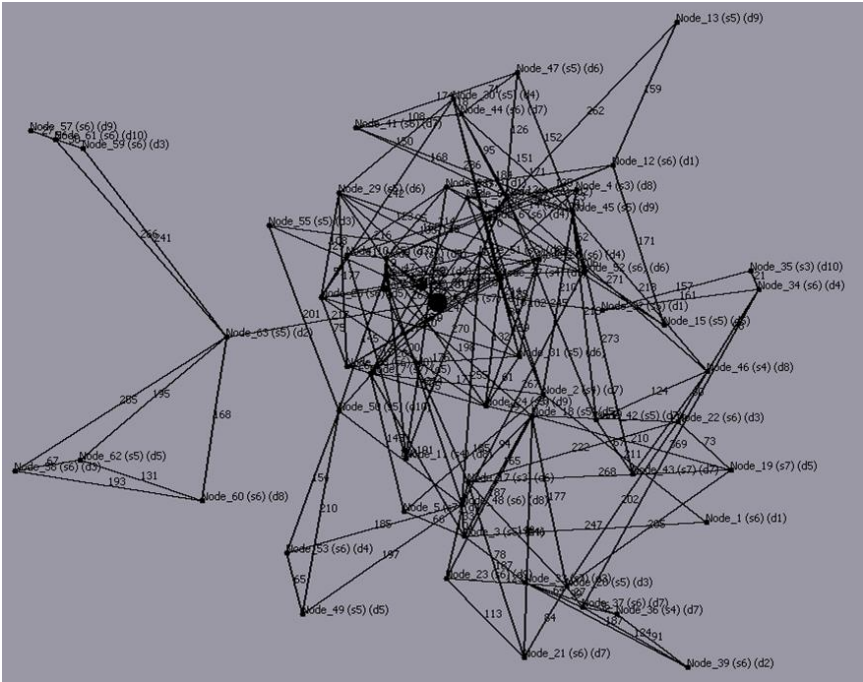
SCRAM Negative Multi-Level SoS

Vulnerability Performance

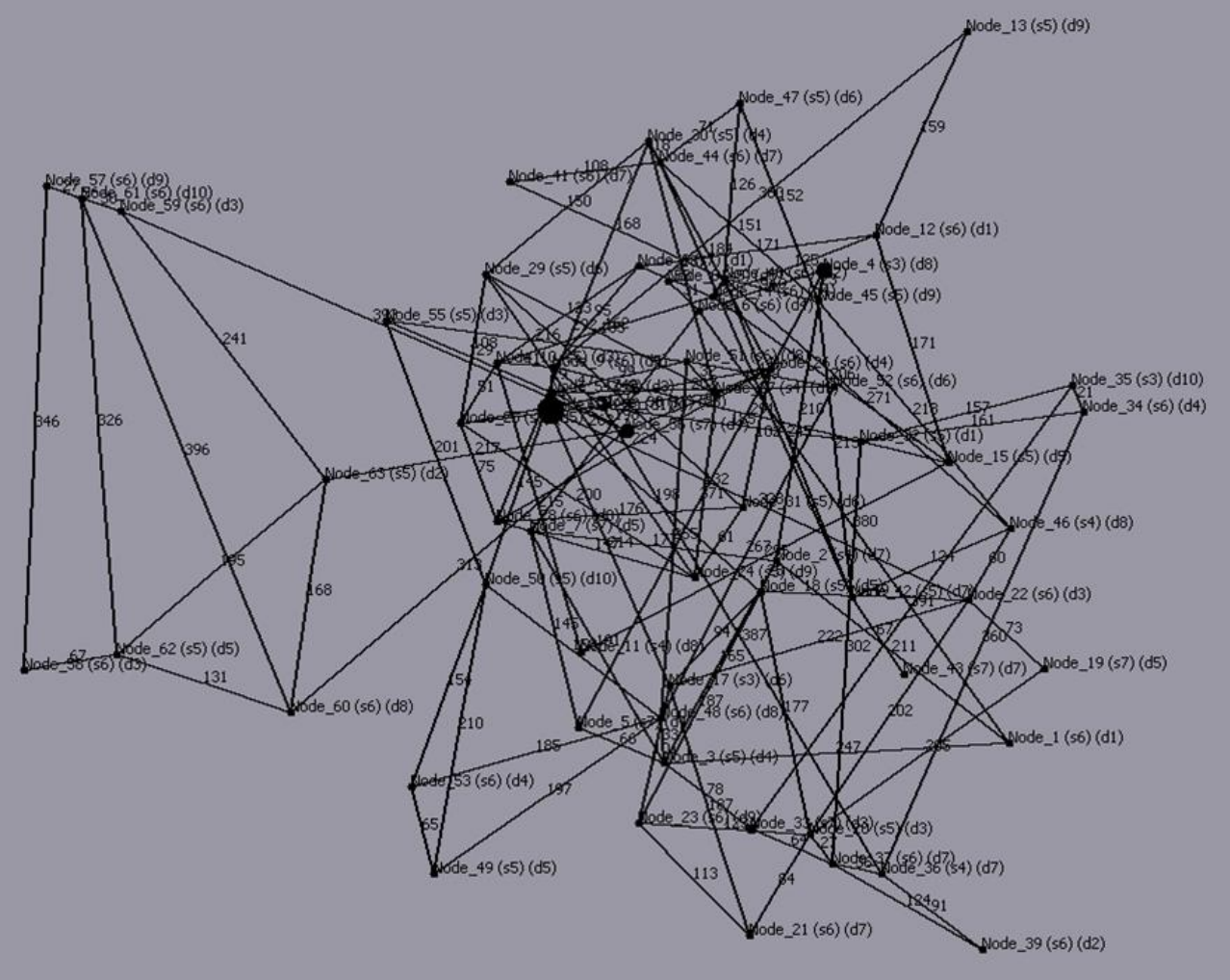
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.28-a, 6.28-b, and 6.28-c.



Appendix D Figure 1. Multi-Level SoS Y with Node Status

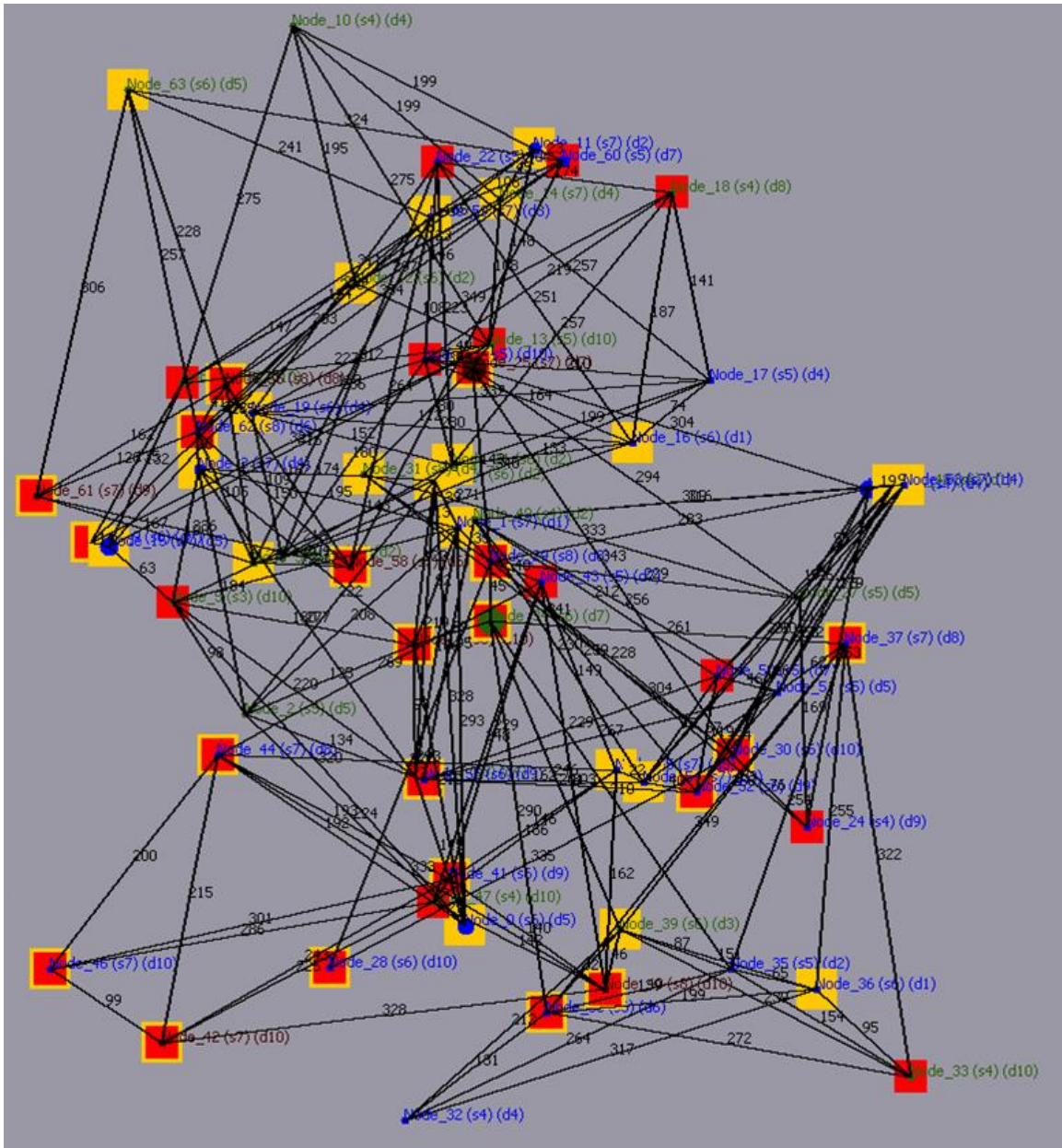


Appendix D Figure 2. Multi-Level SoS Y Topology

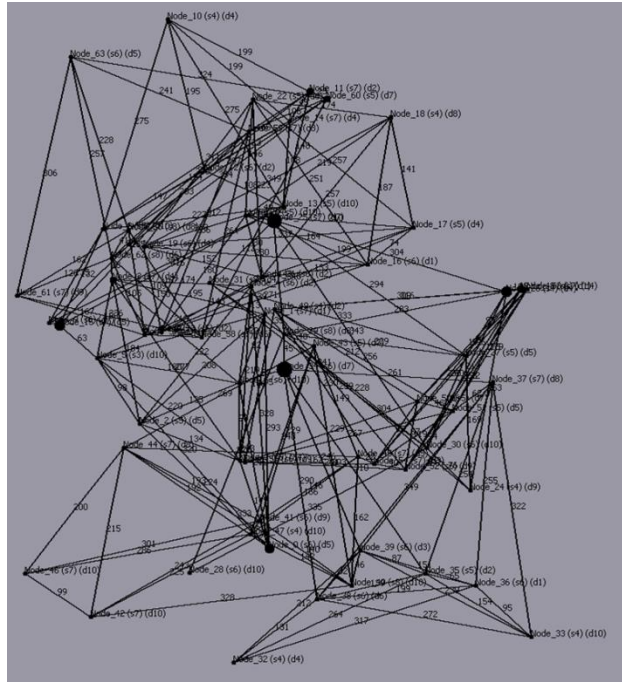


Appendix D Figure 3. Multi-Level SoS Y Optimum Candidate

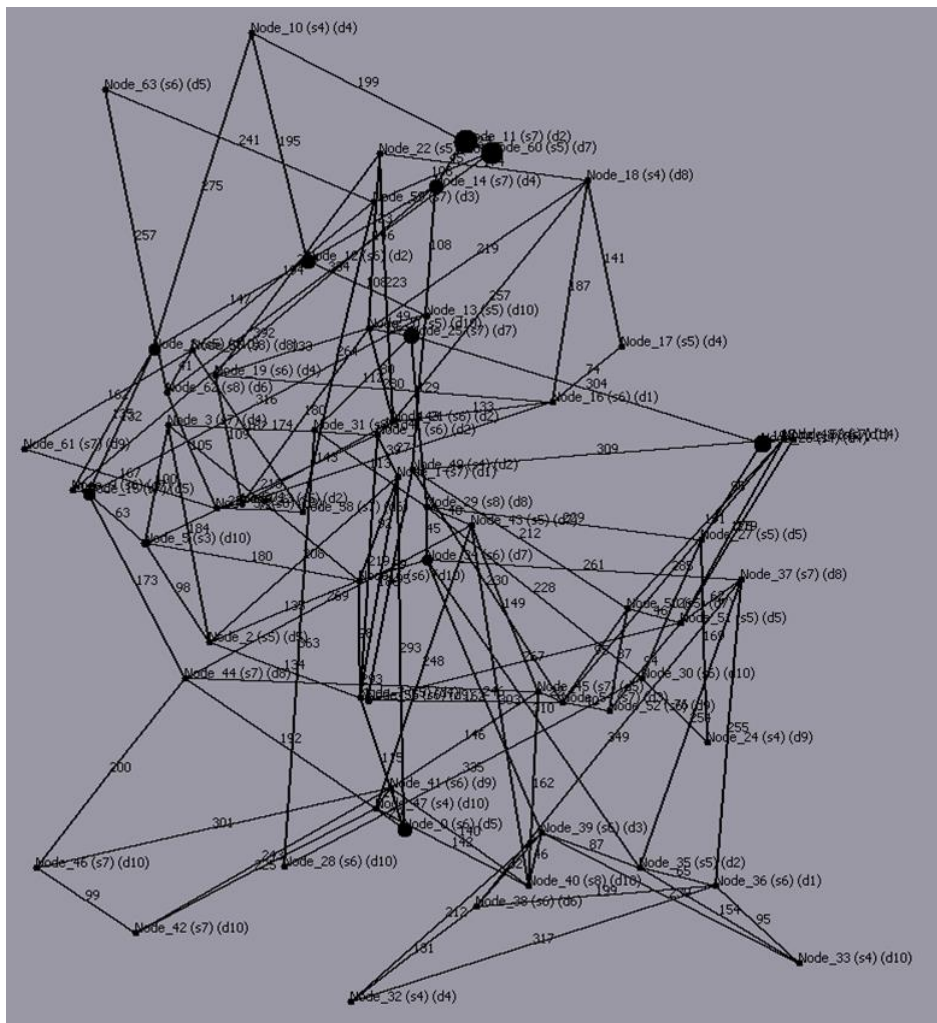
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.28-d, 6.28-e, and 6.28-f



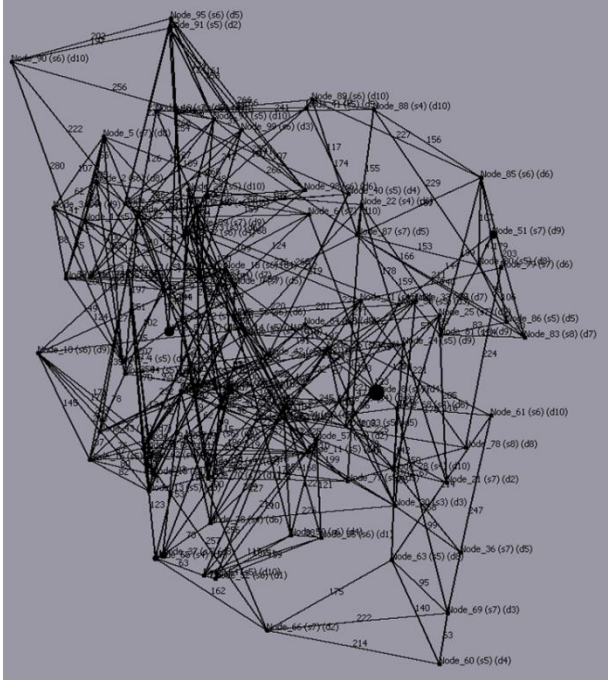
Appendix D Figure 4. Multi-Level SoS Z with Node Status



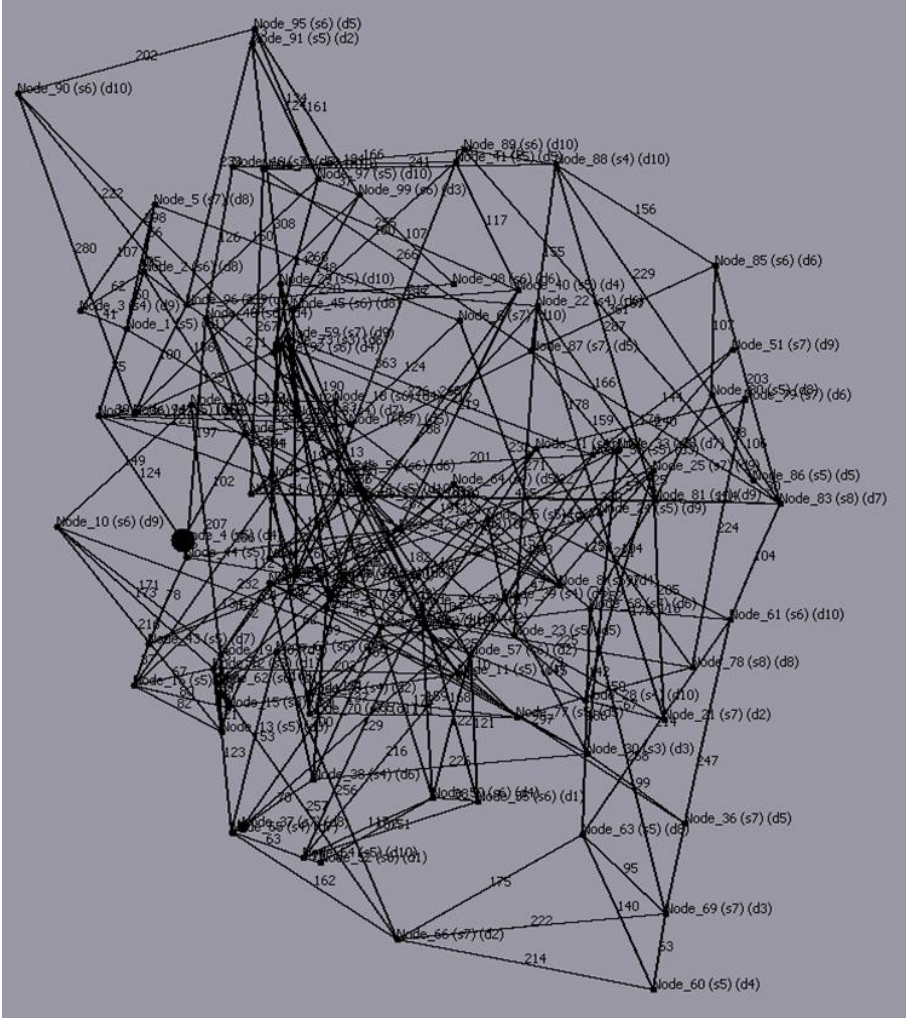
Appendix D Figure 5. Multi-Level SoS Z Topology



Appendix D Figure 6. Multi-Level SoS Z Optimum Candidate

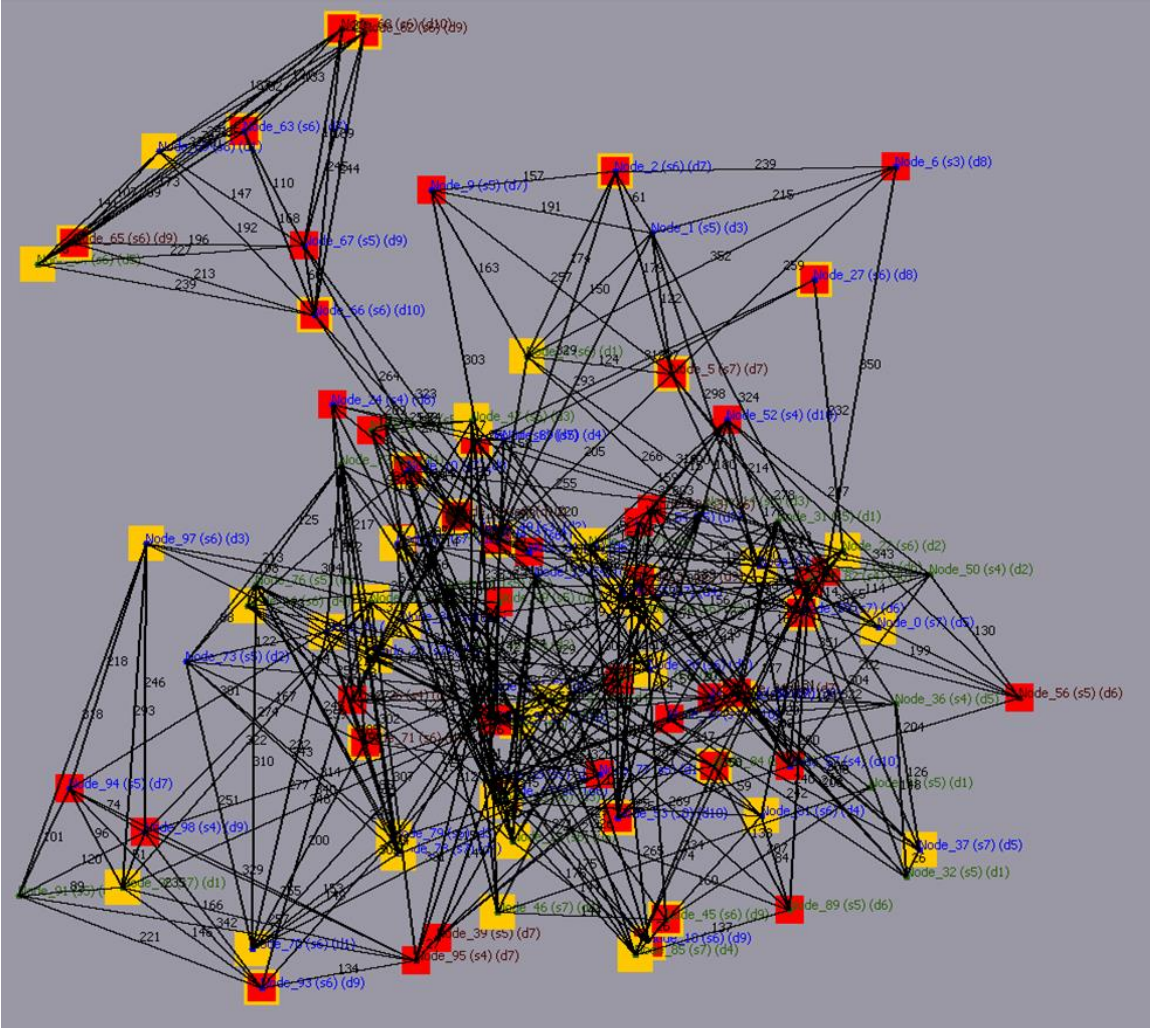


Appendix D Figure 8. Multi-Level SoS AA Topology

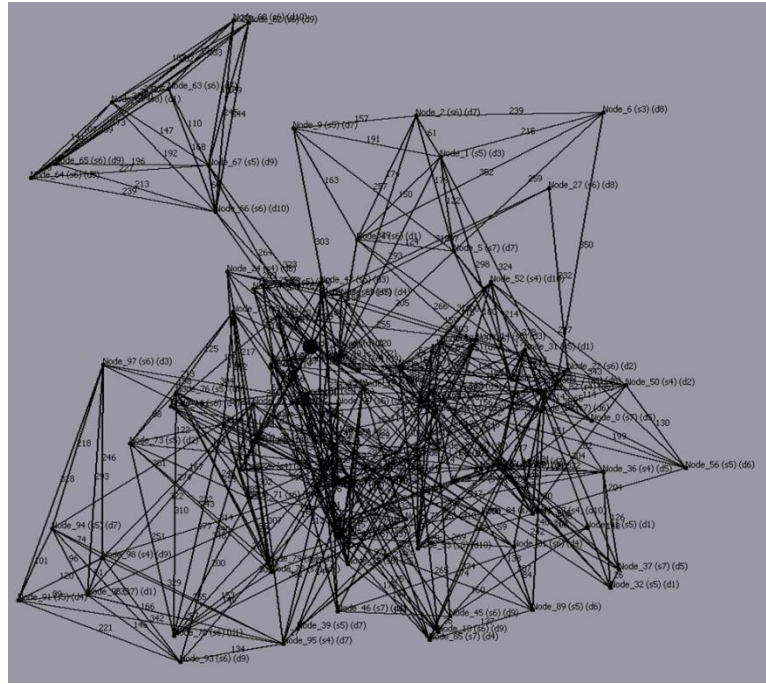


Appendix D Figure 9. Multi-Level SoS AA Optimum Candidate

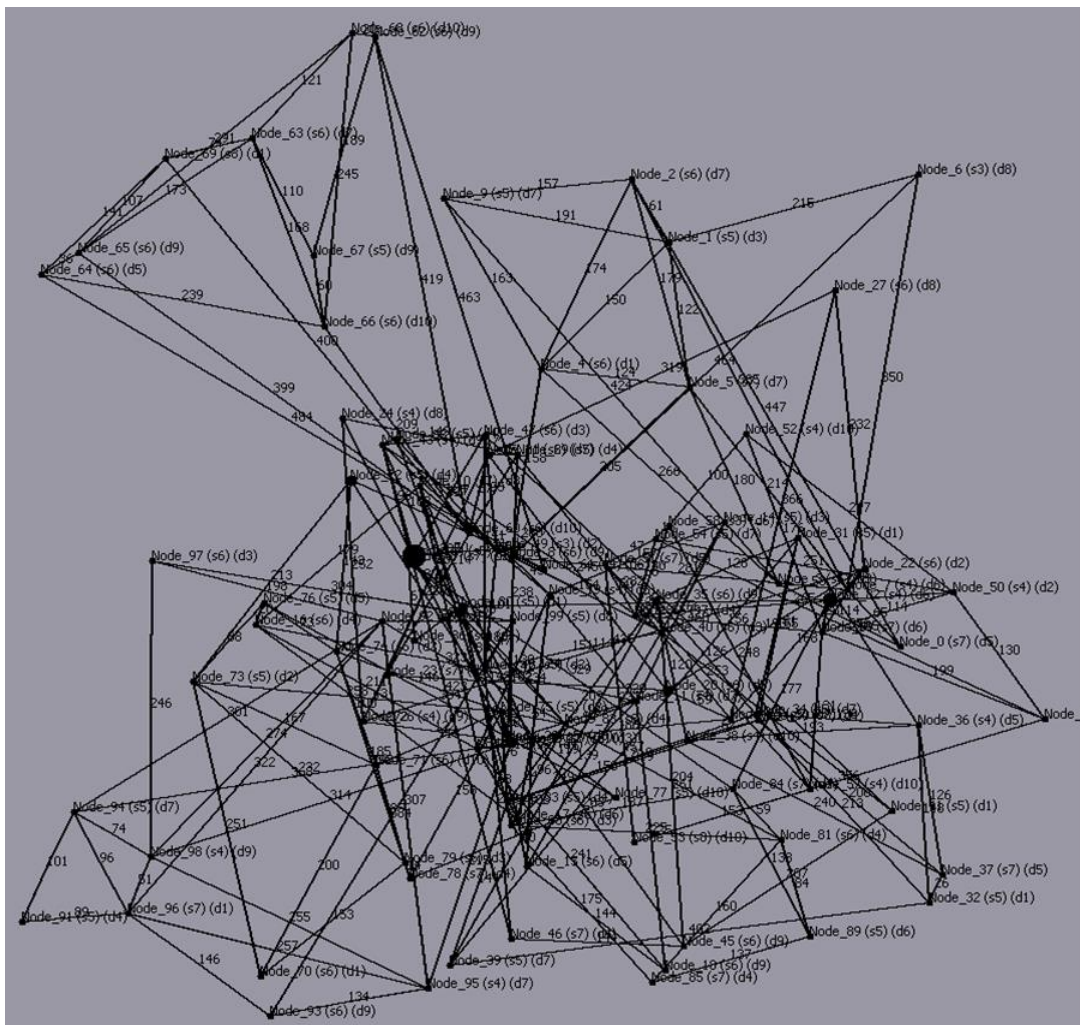
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.29-d, 6.29-e, and 6.29-f.



Appendix D Figure 10. Multi-Level SoS BB with Node Status

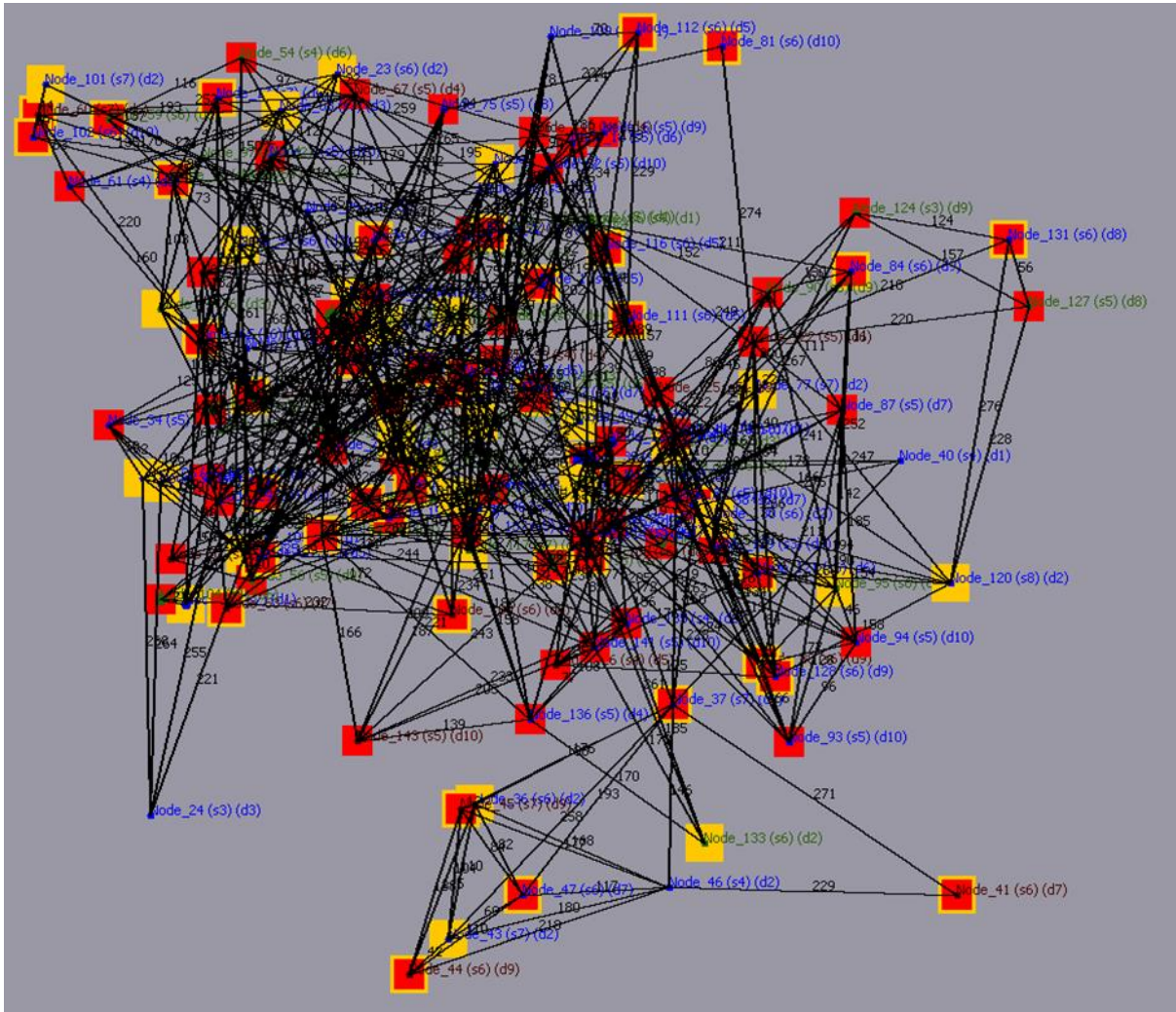


Appendix D Figure 11. Multi-Level SoS BB Topology

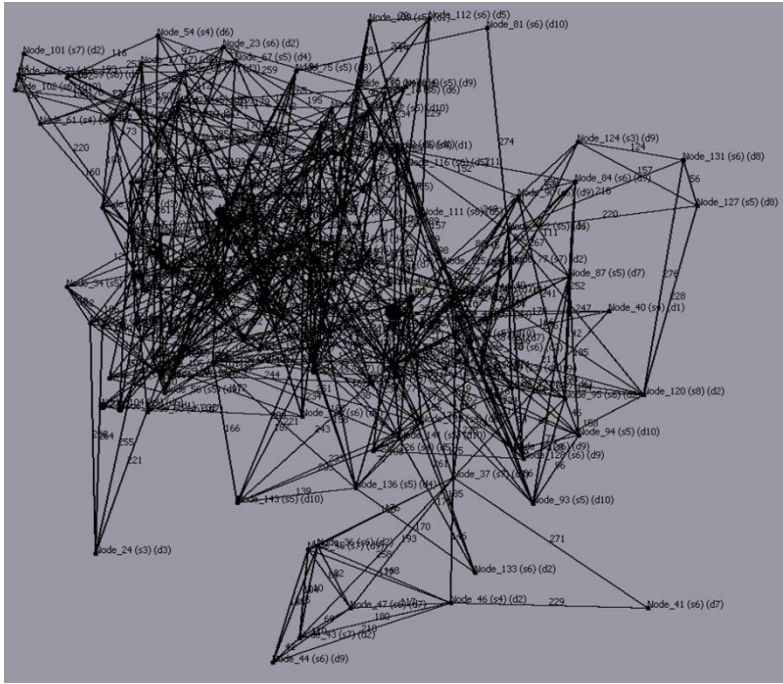


Appendix D Figure 12. Multi-Level SoS BB Optimum Candidate

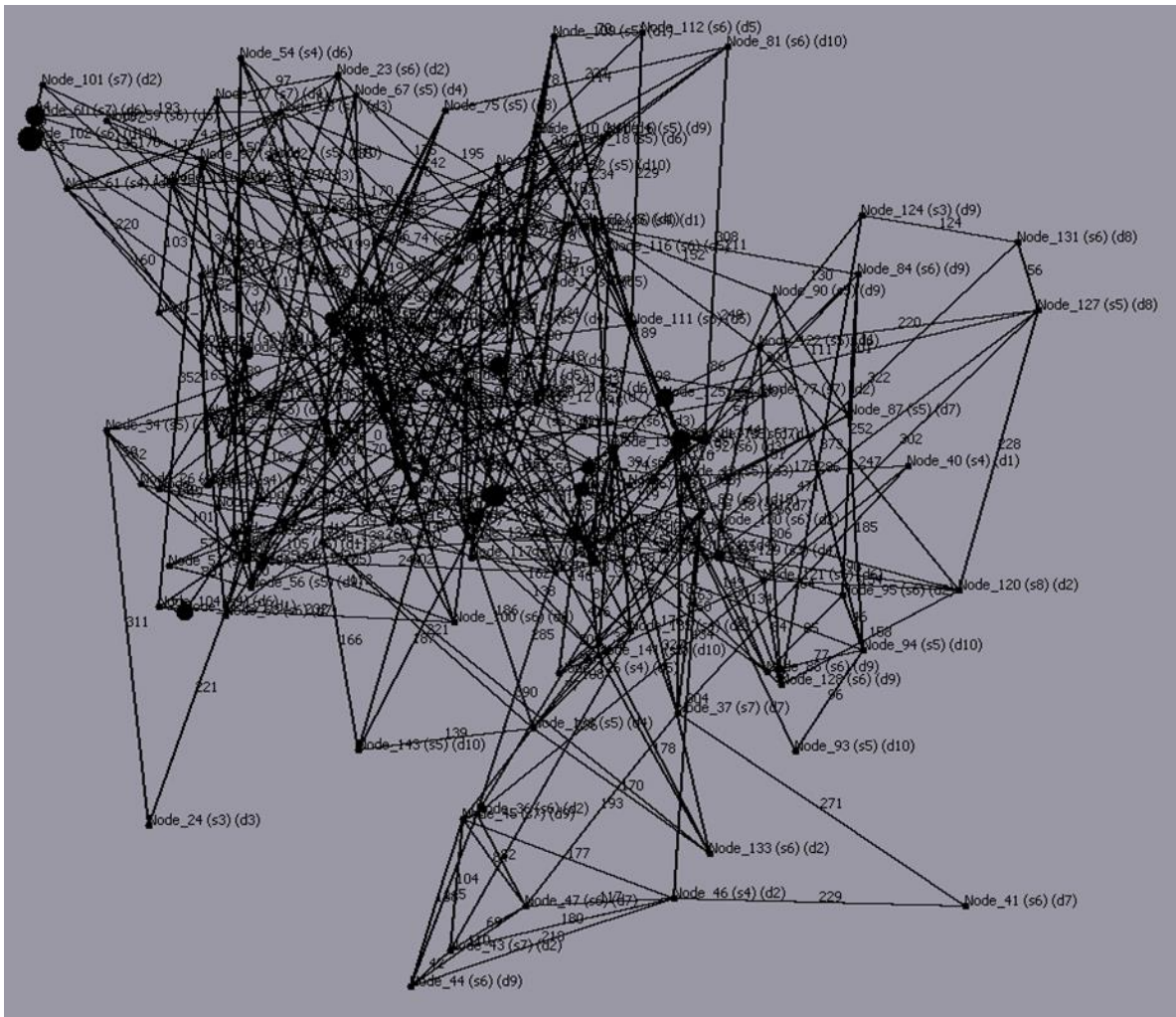
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.30-a, 6.30-b, and 6.30-c.



Appendix D Figure 13. Multi-Level SoS CC with Node Status

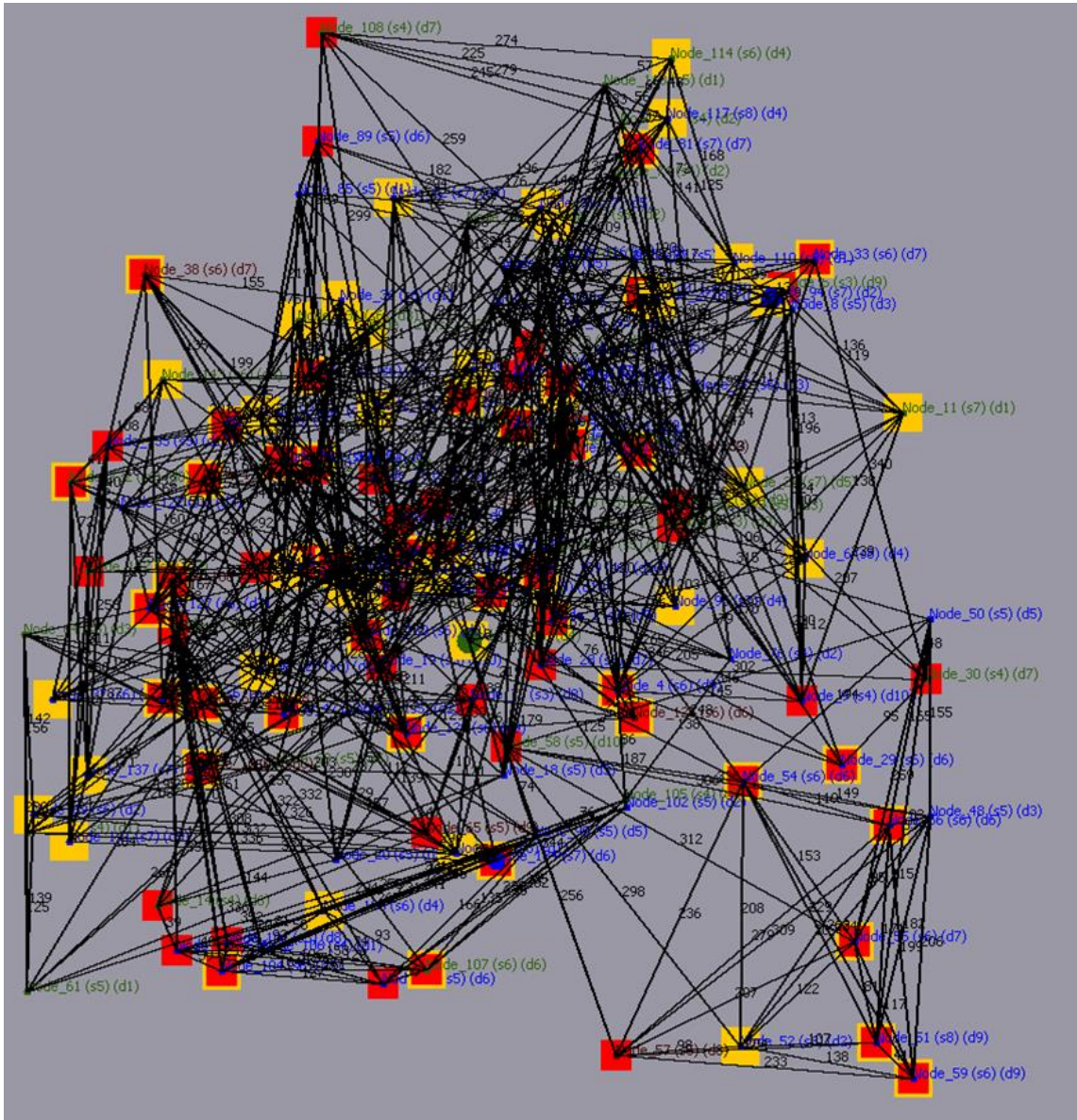


Appendix D Figure 14. Multi-Level SoS CC Topology

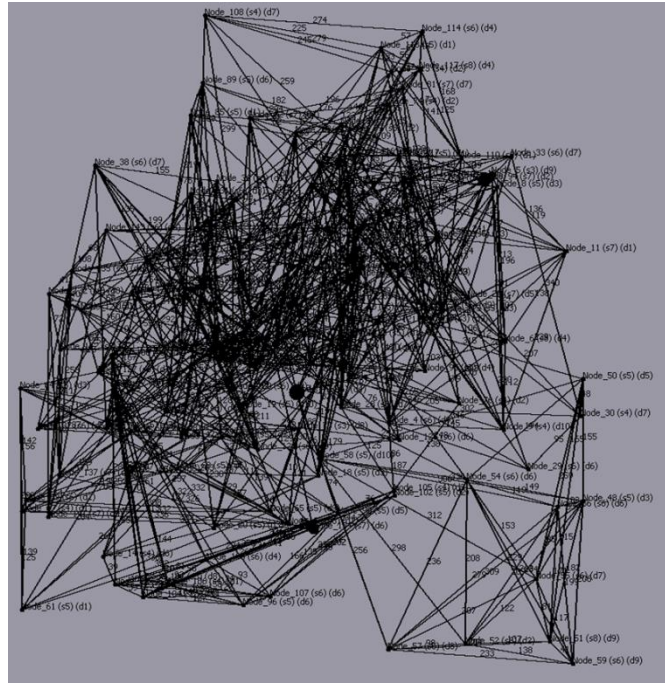


Appendix D Figure 15. Multi-Level SoS CC Optimum Candidate

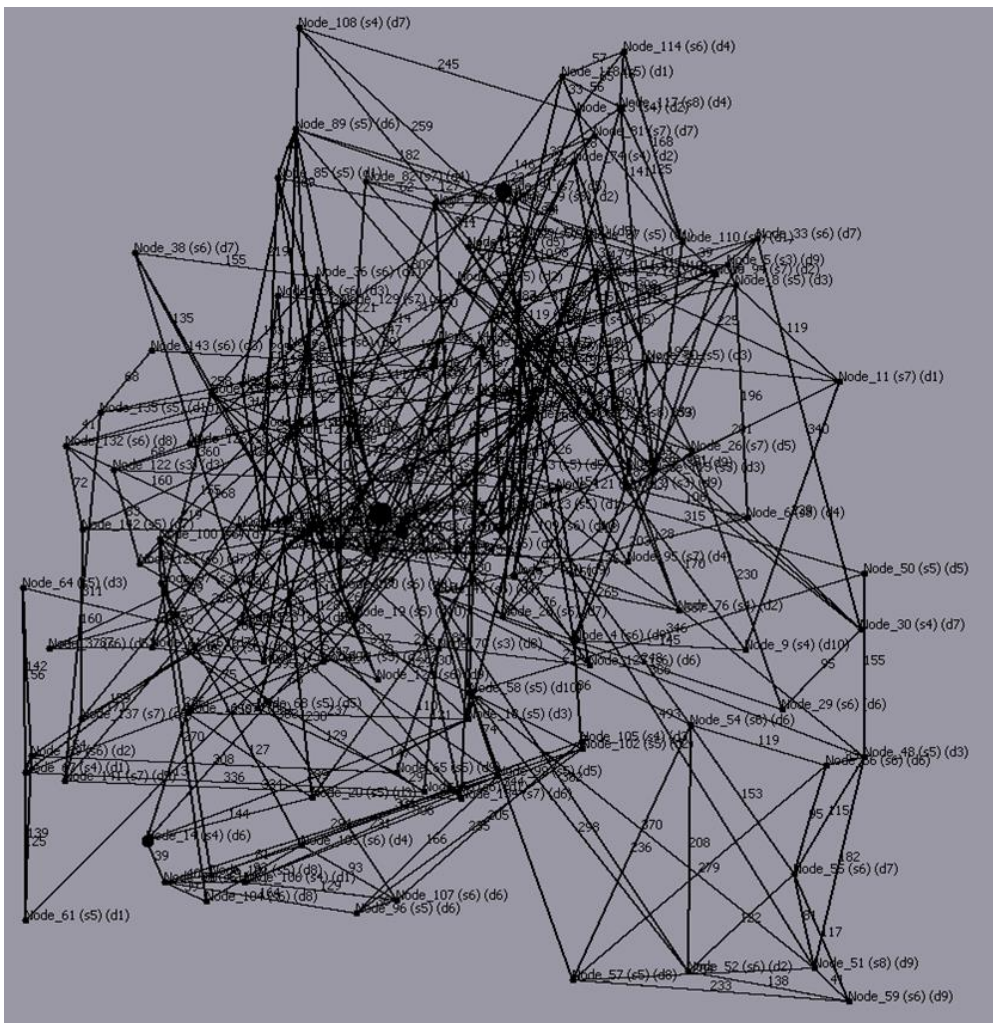
The following figures provide detailed visualisation of the multi-level SoS experiment presented in Figures 6.30-d, 6.30-e, and 6.30-f.



Appendix D Figure 16. Multi-Level SoS DD with Node Status



Appendix D Figure 17. Multi-Level SoS DD Topology



Appendix D Figure 18. Multi-Level SoS DD Optimum Candidate