Model-based Evaluation: from Dependability Theory to Security

by

Saad Saleh Alaboodi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

© Saad Saleh Alaboodi 2013

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

How to quantify security is a classic question in the security community that until today has had no plausible answer. Unfortunately, current security evaluation models are often either quantitative but too specific (i.e., applicability is limited), or comprehensive (i.e., system-level) but qualitative. The importance of quantifying security cannot be overstated, but doing so is difficult and complex, for many reason: the "physics" of the amount of security is ambiguous; the operational state is defined by two confronting parties; protecting and breaking systems is a cross-disciplinary mechanism; security is achieved by comparable security strength and breakable by the weakest link; and the human factor is unavoidable, among others. Thus, security engineers face great challenges in defending the principles of information security and privacy. This thesis addresses model-based system-level security quantification and argues that properly addressing the quantification problem of security first requires a paradigm shift in security modeling, addressing the problem at the abstraction level of what defines a computing system and failure model, before any system-level analysis can be established. Consequently, we present a candidate computing systems abstraction and failure model, then propose two failure-centric model-based quantification approaches, each including a bounding system model, performance measures, and evaluation techniques. The first approach addresses the problem considering the set of controls. To bound and build the logical network of a security system, we extend our original work on the Information Security Maturity Model (ISMM) with Reliability Block Diagrams (RBDs), state vectors, and structure functions from reliability engineering. We then present two different groups of evaluation methods. The first mainly addresses binary systems, by extending minimal path sets, minimal cut sets, and reliability analysis based on both random events and random variables. The second group addresses multi-state security systems with multiple performance measures, by extending Multi-state Systems (MSSs) representation and the Universal Generating Function (UGF) method. The second approach addresses the quantification problem when the two sets of a computing system, i.e., assets and controls, are considered. We adopt a graph-theoretic approach using Bayesian Networks (BNs) to build an asset-control graph as the candidate bounding system model, then demonstrate its application in a novel risk assessment method with various diagnosis and prediction inferences. This work, however, is multidisciplinary, involving foundations from many fields, including security engineering; maturity models; dependability theory, particularly reliability engineering; graph theory, particularly BNs; and probability and stochastic models.

Acknowledgements

All praise be to Allah, the Creator and Sustainer of this universe.

To begin, I would like to express my sincere thanks to my supervisor, Prof. Gordon Agnew. The freedom he gave me as a graduate student to take ownership of my own research interest along with the valuable guidance, support, and motivation was vital to achieving this degree. It made it a unique and outstanding experience, and I am really appreciative. I would also like to thank my thesis committee: the external examiner, Prof. Dawn Jutla, and internal examiners, Prof. Alfred Menezes, Prof. Sagar Naik, and Prof. Catherine Gebotys, for devoting valuable time to reviewing and commenting on this work.

I would also like to thank all my dear friends and colleagues who have contributed to making my study experience exceptional, beyond expectation, and very enjoyable. A special thank you is inevitably due to my friend Dr. Abdulaziz Alkhoraidly for the insightful discussions and engaging companionship during his time studying here. I am also thankful to Ms. Mary McPherson of Graduate Writing Services for the exceptional applied writing skills.

All of my achievements are owed to my family's sacrifices, and so is my deepest and greatest gratitude. My father, may Allah have mercy on him, has always been in my memory at every step I have taken; my mother, may Allah grant her long and happy life, remains my lifelong pillar and inspiration; I owe my parents the whole world; my wife, my partner in this sojourn, has been an invaluable source of much-needed support and encouragement; my children continue to be the truest joy in my world; and my siblings have consistently stood behind me and boosted my moral throughout the course of this journey. I am sincerely grateful and indebted to each one of you as I could not have gotten where I am today without you all.

Finally, I am also grateful to the Ministry of Higher Education in Saudi Arabia and King Saud University (KSU) for granting me the graduate program scholarship to achieve this work.

Dedication

In honor and memory of my father, Saleh, may Allah have mercy on him;

to my mother, Munirah;

to my wife, Najla;

to my children, Faisal and Shahinaz;

and

to my brothers, Mohammed, Abdulaziz, Abdullah, and Bader, and my sisters, Fawziah, Norah, Shekhah, Ghadah, Bedoor, Arwa, and Sumayah.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Information security is unavoidable wherever a computing system exists. Information can be in one of three states: storage, transmission, or processing. Each state may have different threats, and hence, security requirements. The success of the design, development, and operations of computing systems to perform their primary functions is coupled with the success of their protection. Therefore, although the security function is usually a secondary task to the system [1], it remains a core requirement [2].

The ever-increasing security failures, especially due to cyber-attacks, threats, and associated loss estimates have forced researchers to look into new ways to build secure systems, not only by design but also by responding to operational problems [3]. However, a major question has remained without a reasonable, widely agreed upon answer until today: how to quantitatively evaluate operational security. In our view, addressing this question requires a paradigm shift in security modeling. We need to arrive at a unified abstraction of computing systems and a standardised model of security failures, before any system-level representations, performance measures, and analysis methods can be established.

The work presented herein explores this direction of research, one that falls into the growing field of security engineering. This chapter, first, aims to provide the necessary background of the research problem. The research motivation is briefly presented in Section 1.1. This is followed by a description of the research problem in Section 1.2 and a summary of research contributions in Section 1.3. Section 1.4 depicts the overall structure of the thesis.

1.1 Motivation

A chain is only as strong as its weakest link; this proverb, commonly used in the 18th century [4], has become a prominent phrase in security studies: the weakest link determines the strength of the resultant protection of a system [5]. The sad truth in a nutshell, however, is that while we know (or think we know) how to meet the goals of information security, individually building its links, we have not been so successful in realizing these goals jointly as a chain in the face of failure while in operation, nor have we been able to satisfactorily identify the weakest link in a proactive manner. For instance, we protect confidentiality and integrity by means of cryptography and availability by means of redundancy. Yet, the amount of control we exercise over security systems in general remains disproportionate to the efforts we expend. From the industry point of view, we still lack insight into

the effectiveness of countermeasures and anticipated threats, i.e., what is really going on in computing environments, as concluded by CSI survey [6]. On the research side, existing literature in the security domain has arrived at a consensus that there is not yet a widely-accepted concept of security failure [7], [8], operational measures [7], [8], [9], [10] or metrics [7], [9], [11], [12], system boundaries [9], system-level quantitative methodologies [7], [9], [10], and operational datasets [10], [11], [13]. Some, such as [14], consider the quantitative evaluation of security the necessary step before evolving this field into a "science". Nevertheless, the ever-increasing number of reported security incidents and threats concluded from most security surveys, such as [15], [16], along with the continuous growth of security spending and loss estimates, as reported in [17], represent clear evidence of a setback in security engineering.

An important reality, and a challenge at the same time, is that such protection goals may exist in many forms, with different specifications and configurations implemented across different security controls and boundaries, at various abstraction levels. For instance, they might manifest across different layers at the TCP/IP protocol stack in various applications, platforms, communication devices, and technologies; in addition there is the unavoidable involvement of the human factor in such protection. Alone, neither security in theory nor security at individual controls necessarily leads to the security of overall system while in operation. Information about the strength of controls with respect to both specifications and operated configuration [18], and deployment of comparable security strength across system controls, as stated in NIST SP 800-57 [5], are both especially important to building secure, reliable systems. Without a proper measurement of the chain, these requirements, among others, will remain profound obstacles in the field.

This work is indeed motivated by a practical application of the proposed failure-centric approach and associated evaluation methods to engineer secure systems. These evaluation methods can be employed for various purposes, such as design and configuration of secure systems, certification, and auditing exercises. Ultimately, these methods facilitate control over the intended security element of a system.

1.2 Problem Description

Security and reliability are closely related concepts [7], [9], [19], [20], [21]. Therefore, a secure system is one that can be relied on to meet the essential goals of information security: confidentiality, integrity, and availability. The complexity of the information security problem, however, represents a

key challenging characteristic of today's computing. In our view, this complexity dilemma is centered around four entities: assets, controls, threats, and economics, as depicted in Figure 1-1. Assets such as hardware, software, and data have value and need to be well protected. The protection mechanism, however, requires a set of controls or countermeasures, commonly classified into prevention, determent, deflection, detection, and recovery controls, to remove or reduce vulnerabilities. This protection allows the system to fight against a set of predefined threats, which can be interception, interruption, modification, or fabrication activities. Threats, when realized, lead to breaching[1] the perimeter of security goals, i.e., confidentiality, integrity, and availability, and then can cause security failures. Nevertheless, protection mechanisms cannot be implemented without the investment of some resources, ending up with economics as the deciding factor for affordable amounts of the other three sets. The unavoidable dependency and dynamic interactions among these elements feed into the complex process of information security.

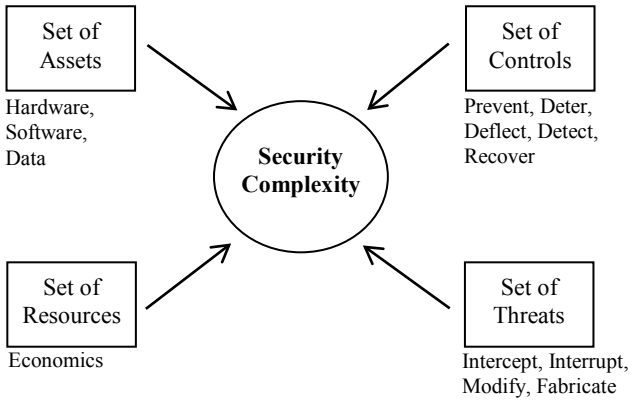

Figure 1-1: Key participants in information security complexity

As such, the security notion has some properties that make it uniquely different from other engineering problems. For instance, security is about a continuous chase between two confronting and competing parties: security defenders on one side and failure sources, whether malicious or nonmalicious, on the other side, in a way analogous to games and battlefields [22], [23]. This view

[1] The phrases "security failure" and "security breach" are used interchangeably.

was considered in modeling the attacker behavior in [24], [25], [26], [27] using game-theoretic approaches.

Additionally, we build security through a chain of comparable strength using a range of means from completely different disciplines, from pure mathematics such as cryptography to applied psychology such as awareness; yet, we break security through any of its links using the same spectrum of techniques, ranging from cryptanalysis to phishing attacks. Thus, cross-disciplinary expertise is required in both securing and breaking systems. In other words, either way, mathematics or psychology alone is not enough anymore, in addition to the fact that it is not a fair game! Operation Aurora in mid December 2009, a sophisticated Cyber-attack originating from China targeting Google and other high-tech US firms, is a clear example. The attack on Google, for instance, was believed to involve combined techniques from reconnaissance missions using social networking sites, such as Facebook and Twitter, through social engineering, such as phishing emails, to high technology, such as exploiting a security flaw in the MS Internet Explorer browser [28], [29].

Furthermore, defining performance measures for operational security is troublesome. To evaluate performance in communication networks, we generally rely on physical quantities that can be captured. For instance, we often use the number of transmitted and dropped bits/packets over a communication channel. Similarly, the performance in circuit theory can be evaluated using the amount of voltage and current flow in a circuit. Unfortunately, the case in security engineering is far more complex as there is no clear, quantifiable "physics" of security by which performance measures of secure systems can be established. Thus, it is hard to quantify the amount of security input fed into a system [19], [30], and the amount of threats a system is facing, i.e., threat input, nor can we eliminate human behavior [22], [23]. However, the evaluation methods and techniques in dependability theory have proved to be useful in evaluating systems' operational capabilities such as reliability and availability, even though such attributes have no direct physical quantities. We build on such theory to propose new methods of quantitative evaluation of security systems.

1.3 Summary of Contributions

The overall contribution of this thesis is in presenting model-based, quantitative methods for evaluating security, as depicted in Figure 1-2. These methods are based on a unified abstraction of computing systems and a consistent failure model, employing failure interdependencies among assets and security controls in these computing systems. To facilitate this study, first, we critically examine

the problem of quantitative security evaluation, considering various aspects important to model development in the security field, such as inductive versus deductive evaluation models, underlying mechanisms versus the impact of failure in security studies, inconsistency in the security modeling paradigm, the issue of dataset unavailability, and then, we describe what went wrong and the suggested measures to overcome current limitations. Following that we present the core research contributions, outlined as follows.

- We introduce the abstraction of a computing system in two sets: assets and controls. We also introduce a failure model based on the impact of failures consequent to malicious and/or nonmalicious causes, along with a discussion of its importance to security studies. Then, we demonstrate the redefined version of the Information Security Maturity Model (ISMM), explaining its architecture, propositions, and how to build the logical network of a security system using Reliability Block Diagrams (RBDs), state vectors, and structure functions. Further, we present common properties of model measures and ways to build the maturity function quantitatively.

- We present evaluation methods extended from reliability theory into security evaluation. Specifically, we present the extended minimal path method, minimal cut method, and reliability analysis based on both random events and random variables. We show how these evaluation methods are built onto the ISMM work as their system model. We also show how to establish the maturity adequacy function and maturity analysis using the reliability measure.

- We introduce the extension of Multi-State Systems (MSS) representation using the Universal Generating Function (UGF) onto the ISMM work. The purpose of this extension is to present a universal approach to evaluating security systems, using multiple performance measures in a multistate setting. We also illustrate how to establish the maturity adequacy function and associated maturity analysis.

- We introduce an asset-control Bayesian network as a candidate theoretical system model to capture interdependencies among the sets of assets and controls. We demonstrate the application of this model by proposing a new risk assessment method. We show the mathematical formulation necessary to carry out risk assessment and demonstrate its use. Furthermore, we provide a mathematical proof on a specific bound of the risk function. We also prove that the method resolves the distinction problem between high-frequency low-

impact events and low-frequency high-impact ones by employing the casual effect property defined by BN topology.

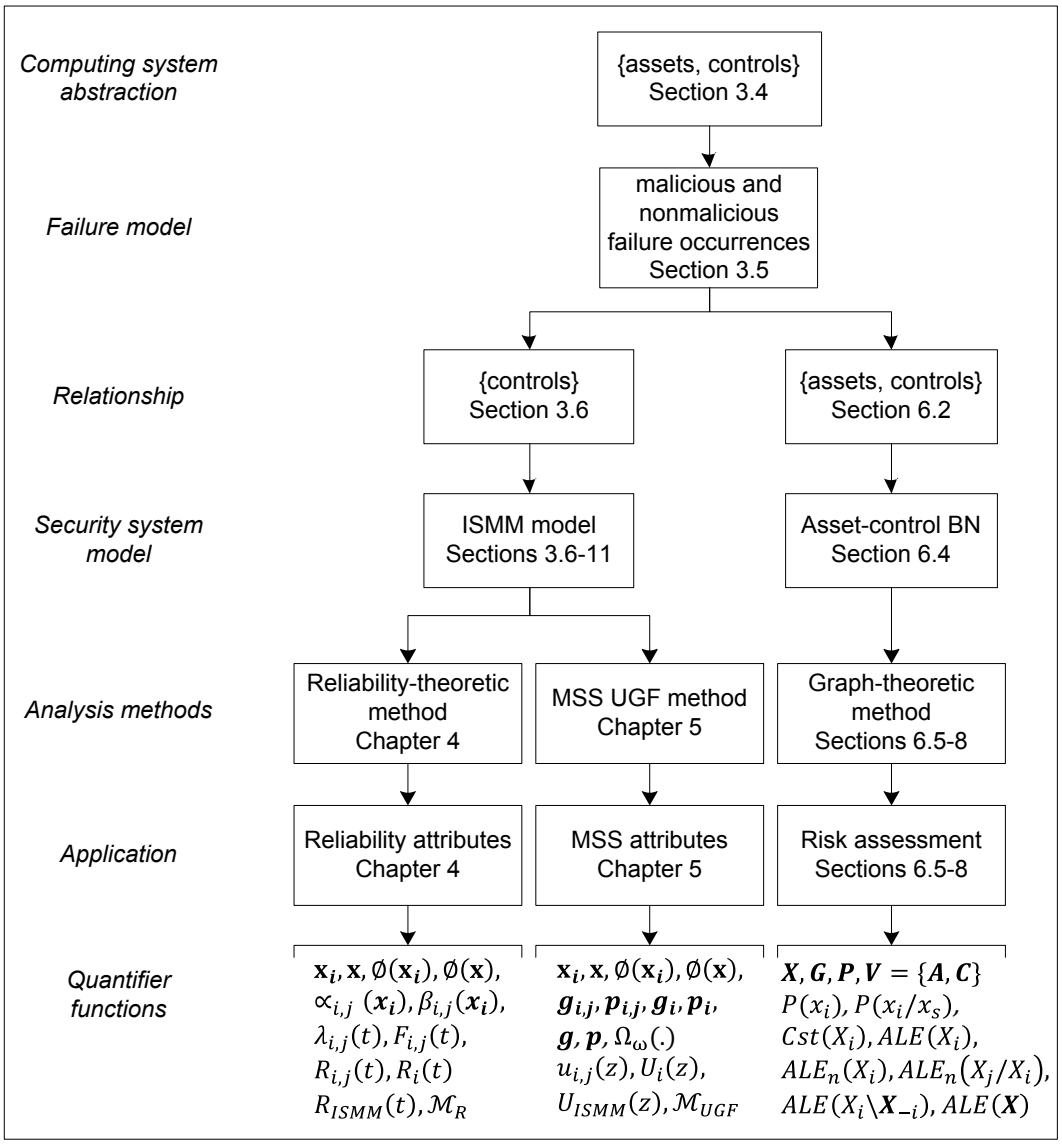

Figure 1-2: Summary of contributions

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 introduces the required background for this research. The first part discusses the problem of quantitative security evaluation from different aspects, such as inductive versus deductive evaluation models, underlying mechanisms versus the impact of failure in security studies and inconsistency therein, the unavailability of failure datasets, and dominant limitations of existing work and how to address them. The remaining parts of this chapter introduce the necessary background for the proposed evaluation methods. In particular, they provide an overview of relevant methods from reliability engineering, then Multi-state System (MSS) representation and the Universal Generating Function (UGF), followed by risk assessment and Bayesian networks (BNs).

Chapter 3 first presents the foundational components of the proposed evaluation methods in this thesis: computing system abstraction and failure model. Then the previous work of the ISMM model is redefined to facilitate the quantitative evaluation methods illustrated later in Chapter 4 and Chapter 5. To facilitate quantitative maturity, ISMM model propositions and architecture are redefined so that the maturity function can be established afterwards. To establish the logical structure of a security system and associated performance measures, relevant tools are extended, mostly from reliability theory, into the ISMM model, such as RBDs, vectors, and structure functions, as well as necessary concepts and definitions. A demonstration on binary systems of these extensions using a simple case study is presented.

Chapter 4 addresses the evaluation of operational security using techniques extended mainly from reliability theory. It presents the extended versions of minimal path set, minimal cut set, and reliability analysis in the cases of random events and random variables. In each method, model formulation and a brief demonstration on binary systems using the earlier case study are presented. The chapter illustrates how the use of these methods can be useful in evaluating operational performance measures and in building upon the quantitative maturity analysis of security systems. This evaluation primarily uses failure statistics and the corresponding logical network of individual controls of the security system.

Chapter 5 generalizes the evaluation approach of security systems to multistate systems with multiple performance measures, and shows how to achieve this objective by extending the Multi-State Systems (MSS) representation and the Universal Generating Function (UGF) method. The

model formulation and a brief illustration using the same case study demonstrated earlier are presented.

In Chapter 6, contrary to preceding evaluation methods, we approach the security evaluation problem by examining the relationship of both sets of a computing system together: assets and controls. To establish the system model, we employ BNs to capture and bound the failure dependency among such entities, establishing the theoretical model of what we call Asset-Control BNs. We then show an application of this model by developing a new risk assessment method, including all necessary formulation and an illustrative case study. Moreover, we present two mathematical proofs: one related to a specific bound on the risk function and the other on the distinction property between high-frequency low-impact and low-frequency high-impact events in the proposed risk assessment.

Chapter 7 provides a conclusion of the preceding chapters and a summary of side notes describing related open problems for each chapter.

Chapter 2
Background

This chapter reviews the background upon which this research is built, beginning with a discussion of related work and its limitations. Section 2.1 critically examines various aspects important to model development endeavours in security studies. It starts with a review of evaluation techniques with focus on model-based quantitative methods. After that, it contrasts inductive models to deductive models, and underlying failure mechanisms to the impact of failure in evaluation models. A brief discussion of inconsistency in security failure models and the issue of failure dataset unavailability is then presented. This section concludes with a summary of overall limitations of existing work and how to resolve them.

The remaining sections of this chapter review relevant preliminaries. In particular, Section 2.2 presents a brief overview of the failure model in reliability engineering, reliability block diagrams (RBDs), state vectors, and structure functions. It then reviews the evaluation methods of interest, mainly minimal path set, minimal cut set, and reliability analysis. Section 2.3 starts with a review of work on Multi-State Systems (MSS) representation, and then describes the Universal Generating Function (UGF) as the analysis method. Section 2.4 presents preliminaries for the proposed asset-control graph. Specifically, it reviews risk assessment in the computing field and Bayesian networks (BNs).

2.1 Various Studies in Security Modelling

2.1.1 Qualitative Versus Quantitative Models

Evaluation techniques: Security evaluation represents one of the essential tasks in security engineering. Security engineering, however, as defined in [31], is a rapidly growing field that requires methods and tools that can be used to build systems that remain dependable in the face of failures, whether malicious or nonmalicious. It includes physical security, information security, and associated economics, and involves multidisciplinary knowledge domains, such as cryptography, secure coding, formal methods, and applied psychology. Evaluation techniques use qualitative and/or quantitative methods in order to provide a systematic and representative view of the phenomena under study, where analysis exercises can be established, and hence can be significant for many security applications. Contrary to measurement-based evaluation, which is more accurate but can be too

expensive, dangerous, or unattainable, model-based evaluation can be less expensive, but less accurate. Moreover, model-based evaluation facilitates a wider range of evaluation activities and analysis tools, such as reasoning, prediction [32], sensitivity analysis, optimization, bottleneck analysis [33], and can be applied flexibly at various stages, including even before the system itself is running [33], [34]. Models are assumed to reflect simplified and sensible pictures of reality, and therefore represent an essential component for evaluation activities of complex systems [32], [33], [35], [36]. Also, the choice of performance metrics is crucial if one is to perform useful analysis, and therefore must be relevant to the context of the application domain and problem of interest [34].

Security models can be seen as either qualitative or quantitative in nature, although a mix of both is common. Qualitative techniques aim to capture certain qualities of the system and hence are mostly concerned with the process rather than the outcome itself. In contrast, quantitative techniques seek to develop a proper method of measurement. A measure is basically a mathematical abstraction that captures some subset of the characteristics for a given object for the purpose of studying its performance [37].

Security models in a broad sense vary according to their intended scope and objective, and therefore, are designed accordingly to capture certain properties relevant to th

e intended level of abstraction. Security models can be employed for various reasons, generally to test if a particular policy coincides with preset requirements, to develop a certain policy, to assist in planning a particular implementation, or to verify the adherence of an implementation to predefined requirements [19], [38]. Among them are access control models, capability-based models, trust-based models, and auditing and evaluation models. There follows a brief description of some of the common qualitative and quantitative security models.

Qualitative models: Common Criteria, initially called The Orange Book, is an international security certification standard (ISO/IEC 15408). It originated as a result of three other standards: i) the European standard, known as the Information Technology Security Evaluation Criteria (ITSEC), ii) the Canadian standard, known as the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), and iii) The US standard, known as the Trusted Computer System Evaluation Criteria (TCSEC). CC is widely accepted and used for evaluating general security features of computer systems. CC contains a set of security requirements and attributes along with a certification process that defines various classes of trust for a security system. If these measures are correctly implemented, they are likely to increase the security of a system, but still cannot be used to

continuously evaluate or guarantee operational security [9]. Measures in the Orange Book, however, are defined in terms of classes or ranking, representing the qualitative nature of security postures.

As an example, on models considering the integrated view of dependability and security, the work of [12], [21] proposed a high-level conceptual system model that represents a security system as a black-box, showing interactions with its environment as system input and output. This approach requires system bounds to be clearly defined, and these can be set at various levels, from a single platform to a whole organization. A system's input is mainly regarded as fault introduction, which could lead to subsequent errors and failures. A system's output is considered the system behaviour: the service delivery or denial to users. This model also reflects the traditional attributes of security: confidentially, integrity, and availability, and maps them into new attributes of security and dependability combined: integrity, correctness, and trustability. The measures of the system are divided into behavioural and preventive ones. Remedies to failures are classified into three methods: threat reduction, boundary protection, and recovery. The model is proposed for reasoning about security with regard to the delivery and deniability of service, i.e., unwanted service disruptions, and seeking its use as a baseline for assigning security metrics to the suggested attributes. However, the quantifiability portions, particularly security metrics or measures, in this model are not defined.

Another form of model is attack trees, a graphical analysis method developed by Bruce Schneier [39], [40]. Although high-level quantitative analysis is possible, attack trees are used primarily to provide qualitative failure documentation, particularly attack data, and their countermeasures in structured and reusable trees. In a tree structure similar to that of fault trees, attack trees use logic gates to encode relationships among tree nodes. Once completed, different values can be assigned to each node, such as possible, impossible; expensive, inexpensive; probability of attack success; cost; etc. An extended demonstration of attack trees is presented in [13] to refine attack data into reusable attack patterns. This work also demonstrated the reusable structure property and referential transparency property of attack trees. The reusability of attack trees increases their application. The referential transparency property, as described in [41], makes the represented abstraction in attack trees scalable, that is, the higher-level description of an entity contains the information of lower-level entities.

With regard to risk assessment, several qualitative methods are described in [30]. Among them is the Facilitated Risk Analysis and Assessment Process (FRAAP) and its variations. FRAAP is a disciplined process intended to document security-related risks with business operations, conducted

for each system, platform, or application at a time. Another technique is the risk assessment matrix, which is a simple matrix that combines the three security objectives, i.e., confidentiality, integrity, and availability, with the two security classes of risk, i.e., accidental acts and deliberate acts. System experts can use this matrix as part of a risk assessment process to qualitatively analyse and identify threats and controls.

Other common qualitative methods are described in earlier work in [23], including the Systems Security Engineering Capability Maturity Model CMM (SSE-CMM) [42], which has been accepted by the International Organization for Standardization (ISO) as ISO/IEC 21827 standard; the COBIT maturity model (short for Control OBjectives for Information and related Technology) [43]; the NIST maturity model (developed by the Computer Security Resource Center (CSRC) under the National Institute of Standards and Technology (NIST)) [44]; the OCTAVE risk model (short for Operationally Critical Threat, Asset, and Vulnerability Evaluation, developed at Carnegie Mellon University) [45]; and the CRAMM risk model (developed by the Central Computer and Telecommunications Agency (CCTA), UK, and named CCTA Risk Analysis and Management Method) [46]. Unlike with quantitative models, mapping between various qualitative models is common to further augment their analysis processes. Examples for such mapping include mapping ISO17799 and HIPAA; ISO 17799 and Sarbanes–Oxley; ISO 17799 and CobiT [30].

Regardless of which qualitative method is used, expert judgements are heavily involved in using qualitative models and the evaluation results. In addition, such methods evaluate the delivered security process, not the security element itself [7], [9], [23].

Quantitative models: Quantitative models represent the focus of this work. The quantitative modelling process in general should be kept simple enough to avoid complicating the calculations of intended system measures, but at the same time, should be descriptive enough to capture the intended behaviour of the real-world phenomenon under study [47], [48]. Over-parameterised models have always proved to have poor predictive capacity [49]. From this point onward, all discussion is assumed to be about quantitative models.

For decades, statistics and related modeling applications have been used extensively in theory and practice to evaluate the dependability and risk metrics of complex systems. In recent times, statistics have been widely used in specific security studies such as complexity theory (e.g., algorithms and cryptographic complexity measurements), intrusion detection systems, and threat modelling. More recently, probabilistic techniques are being increasingly applied in various security-evaluation tasks

and in analyzing underlying attack mechanisms in particular. A wide set of examples can be found in [3], [7], [9], [50], [51], [52], [53], [54], [55], [56], [57], [58]. We briefly demonstrate some of these examples, noting the overlapping between representation techniques (e.g., graphs, random processes, Markov models) and analysis goals (e.g., risk assessment, intrusion detection, trust, attack behaviour).

The work of [55], for instance, proposed a rule-based intrusion detection system that monitors the standard operations on a target system, establishing normal operational patterns. It then uses a statistical method to detect whether a new observation is abnormal compared to previous observations. A lower level of modeling abstraction can be found in [52], [59], which use state transition models to stochastically capture the attacker behaviour in intrusion-tolerant system models.

In network routing, the work of [60] modeled probabilistically node faults, malicious or nonmalicious, and developed accordingly a probabilistic-based route detection and fault-avoidance algorithm for dynamic byzantine adversarial environments. A loss in this model can be caused by packet drop irrespective of the underlying details of how it was dropped. In [56], a trust model is presented defining the levels of security in terms of the probability distribution of the level-of-threat required to achieve a penetration involving information of a particular classification.

Graph-based models in particular are increasingly used in various types of security modeling. For instance, the work of [61] presented a formal model to study quantitatively enterprise-level network risks using a graph-based approach. Graph arcs represent attacks or stages of attacks. Graph edges encode analysis metrics such as probabilities of success, average time to succeed, or costs, representing an attacker's level-of-effort. Probabilistic-based attack analysis can then be performed using various graph algorithms such as shortest-path algorithm. This work, [61], computes near-optimal shortest paths and considers this approach a good measure of security since it models time dependencies, multiple attack attempts, and multi-prong attacks. The work of [11] also introduced a related model for the probabilistic risk measurement of an enterprise network. To assign cumulative probabilities of successful attacks, this model uses attack graphs, described in [61], for model representation, and uses Common Vulnerability Scoring Systems (CVSS) for individual component metrics, described in [62], [63]. Further, the work of [10] adopted influence diagrams as a tool for establishing a decision-driven risk model. The paper uses statistics of the annual frequency of bad events and consequences of bad events along with a predefined set of safeguards in the proposed model. The modelling approach aims to quantify computer security risk and associated annual losses, and derives cumulative distributions of benefits of safeguards and forecast attack rates.

The work of [3], [64] addressed the quantification of system-level security by proposing an assessment model that uses a privilege graph to exhibit operational security vulnerabilities. Attacks in this work are based on both effort and time estimation. The graph is transformed into a Markov chain that represents all possible successful attack scenarios. The Mean Time To Failure (MTTF) is then used as the quantification measure, representing the mean time for a potential intruder to reach the target. Experimental results using this model are presented in [54]. Moreover, a special class of graphs, the Bipartite graph, is used in [65] to probabilistically model the dependency of failures among end-to-end and host-to-host services in a communication network.

Other work focussed on the attack behaviour itself. For instance, in a study presented in [66] about error and attack tolerance of the World-Wide Web and the Internet, attack behaviour is defined by an agent targeting nodes in a preference manner, selecting the most connected ones. Thus, the removal of the most connected nodes in decreasing order of their connectivity is used to simulate attacks. Conversely, random removal of nodes is used to simulate random failures. The work of [67] presented a quantitative model of social behavior of an attack (e.g., skill level, tenacity, financial ability) over resources of a network as an accumulation of a sequence of steps represented by an attack graph. Possible attack paths are modeled as a sequence of probabilistic steps using Bayesian estimates that build the corresponding attack graph. Bayesian networks-based analysis is used to perform vulnerability assessment of the targeted network. This method uses a predefined threshold value for each resource to define an attack-prone point. Other detailed failure aspects such as intentional attacks upon a system containing accidental and nonmalicious intentional vulnerabilities with respect to the effort taken by the attacker are modeled in [9], where a breach process is realized as a stochastic point process in elapsed effort as opposed to a time variable. However, in modelling attacks in particular, nonmalicious faults are more widely accepted to be modelled compared to malicious faults.

Contrary to qualitative models, many software packages for solving various modelling methods exist today, such as Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) [33], [68]; the Hybrid Automated Reliability Predictor (HARP) [69]; and Stochastic Petri Net Package (SPNP) [70].

Plausibility of statistics in modeling attacks: The debate among members of the security research community about the plausibility using statistical techniques to model certain classes of security breaches is still not over [7], [8], [9], [10], [21], [71], especially debate on those failures resulting

from intentional malicious faults or vulnerabilities such as Trojan Horses. For example, with respect to security vulnerabilities, while the work of [71] argues that past data is not representative of valid samples of vulnerabilities and is thus irrelevant to future trends, the work of [10] counters that similar arguments could have been made against gathering mortality statistics in the early insurance industry. Also, [10] continues to stress that past data is relevant to the future but naturally provides partial answers to predicting trends, which can be employed constructively by statistical models. This view is also supported by research conducted at the Computer Emergency Response Team Coordination Center (CERT/CC) at Carnegie Mellon University [72], which hosts one of the largest publicly available security incidents statistics. This research, [72], concluded that new breaches continue to use some of the old information about vulnerabilities and associated attack mechanisms. Today's information systems suffer from the same or similar vulnerabilities that existed years ago [13]. Nevertheless, recent literature addresses the class of intentional malicious faults in the taxonomy of faults [73]. Security modeling approaches also considers both malicious and nonmalicious faults, although the attack process itself remains of a non-stochastic nature [21].

However, it should be noted that this debate is not central to this work. Suffice it to say that the failure definition in the context presented reflects the manifested failure and its impact, as opposed to the specifics of the underlying failure mechanisms, i.e., underlying faults or vulnerabilities, which are mostly the center of the aforementioned debate.

2.1.2 Inductive versus Deductive Analysis

The analysis of systems with respect to failure (or success) can be classified into two common approaches: inductive methods and deductive methods. The direction of the analysis represents the main difference between these approaches [74]. In the induction approach, reasoning is built from specific cases to reach a general result. The analysis starts from an initiating event, and then moves forward to find out its effect on the entire system failure (or success). These methods are also termed bottom-up approaches, answering questions like "What happens if…? What fails?" Examples of inductive analysis methods include Failure Mode and Effect Analysis (FMEA), Reliability Block Diagrams (RBD) [74], Event Tree Analysis (ETA) [74], [75], privilege graph-based system security assessment [3], and consequence-oriented risk assessment [10], [11], [30], [76], [77].

In the deductive approach, reasoning starts with the general result, and then it moves backward to find out the states that contribute to system failure (or success). These methods are also called top-down approaches, answering questions in the form "How does a given state occur? How does it fail?"

Examples of deductive methods include Fault Tree Analysis (FTA) [74], [75], Success Tree Analysis (STA) [74], attack trees [13], [40], and cause-oriented risk assessment methods [67], [78].

Some methods report the ability to perform both types— inductive and deductive analysis. Among those is the work of [61], which uses attack graphs with cycles to quantitatively model enterprise-level network vulnerabilities. This method is presented to analyze how an attacker might be able to compromise a specific asset, or analyze the universe of possible consequences following a successful attack. Also, the proposed conceptual system, the qualitative model in [21], addresses failure as unwanted system behaviour with respect to the underlying chain of fault (attack)-error (intrusion)-failure events, and failure consequences as delivery-of-service and denial-of-service to users and non-users, respectively. In a nutshell, the inductive methods are employed to answer questions on *what* system states are possible; the deductive methods are employed to answer questions on *how* a system state can occur.

Other classifications, although not that common, include morphological models, meaning analysis is mainly based on the structure of the system, and non-system oriented models, such as appeared in [79].

Some literature stresses the importance of studying complex systems from the perspective of both success space and failure space [76]. However, the majority of research indicates that studying systems from the failure space perspective is usually more attainable and practical than the success space perspective [74], [80].

2.1.3 Underlying Mechanisms versus Impact of Failure

Some research focuses on underlying mechanisms of the incident of interest (e.g., fault, failure, attack). For instance, in [13], [40] attack trees use attack patterns, as a generic representation of malicious attacks that occur in common scenarios. These patterns are used to characterise attack goals, preconditions, attack steps, and consequences if successful. In [67], underlying mechanisms of an attack are used to estimate the risk level of resources that might be compromised based on attacker behavior in a given network. The attacker behavior in this work is characterized by a predefined set of social attributes such as skill level and tenacity, which are used to create the network topology of an attack graph of the network resources. These steps represent a course of action that might be taken by an attacker to compromise a particular network resource. In the work of [61], however, both outside and inside attacks are used to build attack graphs. Attack graphs represent the cumulative attack steps

that enable an attacker to gain privileges in the network. Attack templates, configuration files, and attacker profiles are used as an input for each attack scenario among network nodes.

Alternatively, other researchers focus on the effects of the incident of interest. For example, the risk measurement model presented in [11] aggregates CVSS metrics to encode the cumulative effect of vulnerabilities in an enterprise network onto attack graphs. This approach allows cycles in the attack graphs. It also considers both specific features of vulnerabilities, such as the skill necessary to exploit the weakness, and more generic features, such as exposure of weaknesses. The privilege graph in [3] is used to model the effect of an attacker's privileges, encoded by the attacker's node, onto a targeted node, representing some privileges in the system. The work of [10] in computer security risk modeling identified security breaches as bad events with a specific frequency of occurrence and consequences. These statistics are utilized in the derivation of various probabilistic analyses such as annual loss expectancy, net benefits of safeguards, and forecasted attack rates. Further, in the example demonstrated in [78] to model operational risk in financial institutions using Bayesian networks, the frequency of failures is used to model various network risk factors, making up the corresponding BN random variables. This frequency analysis includes failures from malicious sources, such as hacker and virus attack nodes, and nonmalicious sources, such as network failure and power surge nodes. In [30] the qualitative security risk assessment method uses the loss of security objectives, i.e., confidentiality, integrity, and availability, as the bases for defining threat incidents and subsequent risk assessment processes. The impact of security breaches, however, is further extended [81] to financial terms in an empirical study of economic cost that involves publicly traded US corporations. The study took the approach of the impact of security breaches on an overall organization by using the stock market return as the indicator of caused economic impact, realising that consequences vary according to assets affected by the breach.

Other models, though not very common, try to capture both underlying mechanisms and impacts of failure. For instance, the work of [21], which is conceptual and qualitative, addresses underlying failure mechanisms through the chain of fault (attack)-error (intrusion)-failure events and failure consequences through user-level delivery.

2.1.4 Inconsistency in Security Modeling

The inconsistency in security studies with respect to the quantification problem can be easily seen by analogy with other well-founded, related disciplines, such as electrical circuits, communication networks, and dependability theory. These fields share common properties with respect to their

quantitative evaluation methods, which merit further examination. In electrical circuits, a system is abstracted by a set of electrical elements such as capacitors, inductors, and resistors that make up a circuit. Each element takes certain value(s) capturing some physical attribute(s). To study circuits, the system model is first represented by a logical structure defining the electrical relationship among these elements, using a set of predefined arrangements such as series, parallel, and delta. It can also be depicted based on the physical structure of its elements by defining the actual design characteristics. Various analysis techniques and quantifiable performance measures are then established, such as the use of Ohm's law for current, voltage, and resistance calculations. Strong assumptions are usually made so that the consequent mathematical model can be solvable, yet provide informed analysis, such as the linearity assumption to ensure charges and voltages obey the superposition rule. As a result of this modeling foundation, various software packages have been developed for circuit evaluation, such as the SPICE and CircuitLogix tools.

In communication networks, a system is abstracted by a set of networking nodes, links, and terminals, whether it is a wireless network, wired, or a mix of both. The communication capabilities of these components are characterised by well-defined networking attributes such as channel capacity and transmission rate. Network topology is used to depict the arrangement of various components with respect to the physical arrangement for actual design characteristics or the logical arrangement considering the flow of data. Likewise, networks are also evaluated using various widely-accepted performance measures such as the bit error rate, network utilization, delay, and throughput. These measures are calculated using various techniques, including heuristics. As such, this abstraction paradigm of communication networks has facilitated the development of various simulation and modeling packages, such as ns2/ns3 and OPNET.

Similarly, in reliability theory, a system is traditionally abstracted by a set of items that together make up the function of the whole system. The reliability of the system is a relative measure, defined by the reliability of the logical structure of its constituent items with respect to failure, and bounded probabilistically between 0 and 1. Various quantifiable performance measures have been established such as MTTF and MTBF. Accordingly, evaluation techniques are built on such a unified abstraction, such as Fault Tree Analysis (FTA), Reliability Block Diagrams (RBDs), and Event Tree Analysis (ETA). Also, various software packages have been developed for evaluation purposes, such as SHARPE, HARP, and SPNP.

Obviously, certain aspects of the evaluation problem remain common among all the above different systems: the existence of an abstraction paradigm of constituent components of the system; the existence of a bounding system model that defines the logical system structure, i.e., the logical arrangement of system components with respect to particular operational attributes; the existence of quantifiable performance measures; and the existence of evaluation techniques. Being well-founded, not only are these evaluation methods scalable, but they can plausibly be connected with the economics of the corresponding systems. For instance, the evaluation of reliability attribute is linked with its economics through various repair and maintenance analysis methods to the point where we often see metrics such as MTTF and MTBF printed on many electronic components at manufacturing stages.

Unfortunately, this foundational modeling paradigm is not available in security studies. As such, while circuit, network, and reliability engineers, for instance, can sensibly quantify the contribution of a redundant component in their systems operationally and economically, the security engineer cannot perform this kind of task on a redundant firewall.

In what follows, we briefly demonstrate a few examples of such discrepancies from the work reviewed earlier. The work addressing attack modeling in general has neither a unified system model nor a consistent notion or modes of attacks. The work presented in [61] defines attacks based on three inputs: a set of predefined common attacks, a system's configuration, and an attacker profile. This definition allows cycles in the addressed attack scenarios and excludes human behaviour involved in attacks. In [11], the cumulative success probability of an attack in an enterprise network is assigned using an attack graph structure and the individual component metrics known as CVSS. A full attack graph is then defined by three types of nodes: attack-step nodes, representing individual attack steps; privilege nodes, representing a single network privilege; and configuration nodes, representing network configuration facts. In [67], attack behaviors, including social attributes (e.g., skill, tenacity, and financial ability) are used in building attack graphs, whereby an attack is defined by a sequence of actions taken by an attacker against a targeted network. In [3], experts with deep knowledge of the system analyze attacks to construct a database of attack breaking rules; and then use these rules to derive the respective estimation of time and effort values to build a privilege graph that encodes all successful attacks in the system. Other higher-level models tend to use more-generalized definitions of failure. In [13], [40], a more generic representation of underlying attack mechanisms is presented using attack patterns, consisting of attack goals, success conditions, steps, and consequences. In [21],

the proposed system-level conceptual model of security and dependability defines failure as unwanted system behaviour as a result of a previous chain of fault (attack)-error (intrusion)-failure events.

In risk models, the problem extends to defining failure and its consequence as opposed to defining failure alone. For instance, the work of [10] used the notion of a bad event index to encode these general security breaches: information theft, information modification, information destruction, system outage, employee theft, and system degradation. In the example demonstrated in [78] to model operational risk in financial institutions, the frequency of both malicious and nonmalicious failures was used beside other qualitative measures (such as network availability level and server hardware quality) to model various network risk factors that make up the model. The work of [30] sets the notion of failure as events that lead to the loss of confidentiality, integrity, or availability, including both accidental and deliberate acts, which in turn could have severe or catastrophic adverse effects on organizational operations, assets, or individuals.

To summarize this part, the current notion of security failure is diffuse, as it has always been a nonstandardized, inconsistent term [10]. While security attributes, i.e., confidentiality, integrity, and availability, have remained consistent in their definitions in research and industry, their use cannot extend to consistent, universal quantitative measures to system-level evaluation. Therefore, the current metrics/measures are imprecise [12], and satisfactory quantitative security models are not available [7], [9], [10], [11]. Table 2-1 provides a generic comparison between dependability engineering and security engineering in the light of the above discussion.

Table 2-1: Gap analysis: dependability engineering versus security engineering

Criteria	Dependability engineering	Security engineering
definition/concept: success, failure, or both?	Success and failure are well defined	Success (non-failure) is well defined, but failure has no widely accepted definition
	Success: dependability goals	Success: security goals
	Failure: deviation from correct service	Failure: case-specific definitions
Goal:	Dependability of systems	Protection of systems
Attributes:	Reliability, availability, maintainability, etc.	Confidentiality, integrity, availability
qualitative, quantitative, or both?	Both qualitative and quantitative use	Mostly qualitative use
Quantitative metrics/measures:	Reliability, availability, MTTF, MTBF, etc.	No widely accepted quantitative measures of CIA attributes
Implementation tools:	FTA, ETA, RBD, FMEA, etc.	For access controls: RBAC, MAC, DAC, etc. For individual protocols or technologies: confidentiality by cryptography, integrity by hash functions; and availability by redundancy, etc.
System-level or case-specific use?	System-level use	Case-specific use
Datasets: operational data with physical and logical diagrams of system configuration	Available	Not available
Evaluation models:	FTA, ETA, RBD, FMEA, etc. Using common system abstraction and failure model	System-level but qualitative (COBIT, OCTAVE, etc.) or quantitative but specific (attack graphs, privilege graphs, attack tress, game-theoretic models, risk assessment models, trust models, etc.) Using different system abstraction and failure models; thus, different measures and functions
System-level and quantitative?	Yes	No

2.1.5 Dataset Unavailability

Whether the evaluation purpose is for a software component, hardware component, behavioral attribute, or any combination of these, the unavailability of appropriate datasets represents one of the

main challenges to validating the accuracy of models. Unfortunately, the security domain as a study field lacks representative statistical failure datasets [10], [11], [13], metrics [7], [9], [11], [12] whose attributes can be studied and whose quantitative measures can be established and validated, and concrete benchmarks to which model insights can be compared [11]. Sharing accurate, complete failure datasets is very limited due to various concerns, among them, legal liability, competitive advantages or reputation [10]; fear of attackers using such data, or loss of public confidence [13]; or privacy issues [82]. As a result, various assumptions such as the use of hypothetical data and scenarios or statistical distributions are commonly used to build and validate such evaluation models. In a nutshell, validating a security quantification method is an especially difficult issue [7].

In what follows, we briefly demonstrate the common practices used in work addressing modeling for security evaluation. To demonstrate the risk measurement model presented in [11], hypothetical attack graphs and success probabilities are assumed. In [67], Bayesian probability estimates are used to demonstrate a proposed vulnerability analysis of network resources based on attacker behavior. Threshold risk values are also estimated for network resources to define the points where resources are attack prone. The work presented in [61] to perform network vulnerability analysis used simulation techniques to model scenario-specific attack templates, system configuration, and attacker profiles. As characteristics of intruder's profiles are not available to the work in [3], scenario assumptions are used in the assessment model to exhibit system-level vulnerabilities on a privilege graph, bounding all successful attack scenarios. Also, the work of [13] used a hypothetical enterprise environment and structure to demonstrate the application of attack trees, showing how attacks can be represented in a structured and reusable form. In the security risk methodology proposed in [10], a hypothetical example with data estimates that represent a scenario common in many organizations is used. In the example demonstrated in [78] to model operational risk in financial institutions using Bayesian networks, a hypothetical scenario, including network nodes and their relationships and associated datasets, is assumed.

Some research addresses the dataset generation itself. For instance, the work of [82] presented an approach to dynamically create network intrusion datasets as opposed to one-time-use data. The approach is based on a set of predefined detailed description of intrusion profiles and guidelines that define acceptable datasets. An experimental network setup is used to capture and establish network traffic and intrusion behavior in a testbed environment. Regardless of the quality of such data, the generated traffic and associated anomaly patterns remain restricted to the scale of the experiment.

However, the work of [10] argues that the challenges facing the development of appropriate risk assessment for computing systems are not unique to the IT industry; financial markets and the insurance industry have dealt with risk quantification, irrespective of the uncertainty involved, the unavailability of appropriate statistics, and the technical challenges. However, some, such as [13], argue that attack datasets, although not yet at the preferred level, are becoming more available than before as a result of increased public interest in and media coverage of Internet security. The work of [10] also identified three forces for pushing towards a new security quantitative framework: security insurance needs, avoidance of liability, and market competition. Once insurance claims and compensations start rolling, statistics will proportionally develop, including metrics such as frequency of incidents, losses, and so forth. Furthermore, we argue that with the recent advances in computing paradigms, these forces of change will accelerate significantly. In the cloud paradigm, for instance, the change will initiate especially from the Cloud user's side with respect to the insurance needs and exposure liability, pushing the establishment for a sensible quantification ground before risk is transferred to the cloud provider.

Today, there are only a few credible statistical reports and surveys about failures, including cyber-attacks, such as those compiled by CERT SEI [83]. In our work, however, failure datasets alone cannot be meaningful for the proposed evaluation methods if not combined with the corresponding system configuration and logical topology with respect to failure. Therefore, to evaluate and validate the proposed work, we have extended the literature review, trying to locate a suitable failure field dataset with system logical diagrams. We have searched available BSTJ Journal, Bell Labs, and NASA historical data; and we have also contacted Google and Cisco with requests for a suitable dataset, but we achieved no success in these endeavors. Moreover, we realize that setting up a representative experimental environment would be very expensive; and would need to run for quite some time in order to establish quality statistics [84], which was not possible either. Nevertheless, major parts of the proposed models follow analytically from reliance on well-founded theories and modelling techniques, such as reliability theory, Multi-State Systems (MSS) and Universal Generating Function (UGF), and Bayesian networks and their inference algorithms.

If more resources and time were available, to establish parameter values of a failure model, one could uses: 1) security auditing tools, as suggested in [7]: vulnerability scanners such as Nessus; network scanners such as nmap; security scanners such as Tiger; and a host scanner such as COPS; and 2) event logging tools: such as Tripwire and InTrust.

2.1.6 What Went Wrong

Defining a plausible, quantitative evaluation of the actual behaviour of security systems that one expects remains an unavoidable challenge within the security community [7], [9], [10], especially in today's computing where cloud and ubiquitous computing, mobility, and constrained resources are becoming normal features for computing devices and platforms. Most current evaluation methods suffer from one major limitation: the inability to capture the actual strength of a security system. To do so, we need to arrive at a clear abstraction of a computing system and definition of a failure model, before any system-level representations and analysis methods can be established. Then, we need to develop models that can sensibly capture and bound the effect of security controls and impact of associated failures using a common model foundation, facilitating dual, confronting views of security systems. Doing so will allow consistent, structural semantics to be defined, so quantifiable performance measures of security controls, at any stage, from goal setting through design to operational lifetime, can be established.

We argue that two flaws have led to the above limitation, which is the inability to capture the security element itself. First, the inheritance of some irrelevant notions from other modeling approaches into security models renders the latter unable to capture confronting parties (i.e., a game-theoretic paradigm), dynamic progression, and the human factor. Second, the center of focus of model development in security studies is extremely misaligned. That is, it is either too specific or too general.

Inheritance issues: In our view, the aspect of inherited issues has appeared as a result of extending some modeling semantics from other engineering fields that do not necessary lead to a successful application when adopted in security modelling [22], [23]. We present three features that differentiate security models from other models. First, capturing two competing parties, i.e., game-theoretic paradigm, onto the same model foundation is intrinsic to security modelling to reflect the adversarial chase between both the protection system (i.e., security system) from one side and the failure system (i.e., failure sources, whether malicious such as attackers or nonmalicious such as accidents) on the other side, somehow as a battlefield. In contrast, capturing this paradigm is not common in other modelling endeavors. That is, it is either not applicable or simplified and modeled in isolation when applicable.

Second, the progression in a security system is dynamic and multi-directionally evolving in nature once the system is operational. This setting makes security start first with continuity and resilience in

mind against failures. To deliver security, comparable security strength must be implemented across system controls and security boundaries, and security principles must be jointly realised in the face of failure as a chain while in operation. Thus, a security system might, for instance, require the redesign of certain controls due to a recent technology, auditing, or enhancement, and a recovery of others due to a recent attack, all in the same time and in a mixed order, dictated by the current state of the security system. In contrast, the progression in other systems usually proceeds in steps or increments with a clear start and an end. This setting is much less volatile and more structured than in security.

Third, the human factor in the security context is the most important element that affects security behaviour, representing an intrinsic element for both protecting and breaking security. Thus, its inclusion when building models becomes a necessary condition. In contrast, other contexts do not normally reflect the human factor as an identifiable element among model components.

Several examples exist that support these observations. For instance, the noise and interference in communication networks, representing the confronting party per se, are commonly modeled in isolation by known distributions. That is, the behaviour of the noise and interference does not intentionally change to break the flow of packets in an adversarial manner. The delivery of packets is also governed by a sequenced layered architecture, such as the TCP/IP protocol stack, without actively involving human behavior. Likewise, the semantics used in the software engineering capability maturity model, such as defined in [85], particularly maturity sequencing qualitatively, not directly capturing confronting parties, and not directly capturing human involvement, are not too successful when extended to security maturity models.

Coverage issues: The second aspect of the limitation above centers around models being too extreme in their application scope. To a large extent, security evaluation models we have today are often either too specific, especially for quantitative models, or too general, especially for qualitative ones.

Being too specific is basically due to the limited coverage in the case of the quantitative models, as they cover partial areas of the system, and therefore, cannot capture the behaviour of the overall system [7]. This limitation applies to many examples, such as the approaches presented in [3], [9], [54], [86] to evaluate security controls and model some attack behaviours, and approaches in [24], [25], [26], [27] to address the security dilemma from a game-theoretic perspective. Hence, the usefulness is limited to the specific cases where assumptions made may hold.

On the other side, being too general is basically due to the qualitative nature in the case of the comprehensive models, such as [12], [39], [42], [43], [44], [45], [46]. These models can only reflect the existence and adherence to a predefined set of controls. Therefore, they reflect a high-level view of the system, measuring how well a security process is implemented [7], [9], [23], addressing organizational acceptance issues [10].

So, the limitation exists mostly with respect to the breadth level for the quantitative models and to the depth level for the qualitative models. Consequently, either way, minimal inference can be drawn about system-level situational security states.

2.2 Reliability Engineering

Reliability at its broadest level is considered a performance measure. It is an attribute of dependability analysis and is defined as the ability of an entity to perform a required function under given conditions for a given time interval. So, it is basically the continuity of correct service [87]. Essentially, the reliability term has a wide spectrum of application. It can be applied to evaluate all various types of human activities (a reliable person is dependable and trustworthy) and physical systems (a reliable system or functional object is dependable and trustworthy), and it is associated with the behaviour of operations [47].

Reliability studies cover two main elements: quantitative and qualitative. Historically, it started as a qualitative measure when first introduced in the aircraft industry in 1930s after World War I (for example, the judgement of two engineers is more reliable than one), and then slowly became quantifiable through the 1940s using the theory of probability and statistics (for examples, the average number of failures or incidents in a particular time frame of operation under certain conditions, or one incident per 1,000,000 hours of flying time for specifying aircraft requirements). Later, reliability concepts became well developed, were quantified by the 1950s, and were applied to various applications, such as space mission programs, electric power systems, and telephone and communication systems [47].

In the late 1960s, models for quantitative software reliability started to appear in the literature [88], [89], [90], [91], [92], representing a major contribution in the field. The earliest published work applied Markov processes to software development [90]. Later, research addressing modeling and analysis of systems featuring both hardware and software errors and faults commenced [93].

Later advancements in this field took various directions. For data estimation as an example, the works of [94], [95] presented methods for extracting lifetime data of various products from product warranty claims. Other work addressed the modeling technique itself, such as the work [96], [97], which extended the use of Bayesian Networks to estimate reliability.

Furthermore, the International Federation for Information Processing Working Group 10.4 (IFIP WG10.4, established in 1980, revised in 1988) has proposed the concept of dependability as an umbrella term for the attributes availability, reliability, safety, confidentiality, integrity, and maintainability [33], when researchers started to dig deeper into the correlation of these attributes [7], [9]. This move has led to the realisation of their common aspects and the development of new integrated understanding and paradigms, such as the combined security and dependability tree of attributes, threats, and means to attain the combination [8], [20].

Evidently, classical dependability offers a wide range of concepts and analysis methods that can be useful in security studies [7], [8], [9], [20], [20], [21], [27], [73], [86], [98], [99], [100]. In this section, we review the relevant subjects from reliability engineering upon which some of the proposed evaluation methods are built, particularly in Chapter 3 and Chapter 4. We first review the failure model. Next we review how structure functions and reliability block diagrams are built as a representation technique defining logical connectivity between system components. This is followed by an introduction about minimal path and cut sets and how reliability analysis is established. These tools help to establish the study of components and their dependency effect on overall system functionality.

2.2.1 Failure Model

The identification and evaluation of failure is central to risk and dependability studies. Therefore, defining what failure means is an intrinsic component in studying various operational attributes of systems such as reliability, availability, maintainability, and risk [74], [75], [89], [90], [101], [102], [103], [104], [105].

The relationship among faults, errors, and failures is usually represented by a chain, as depicted in Figure 2-1. A *fault* in the system is the assumed cause of an *error* and is considered active when it causes an error. Error is a subsequent state to fault(s) representing the part of the system state that could lead to subsequent failures. *Failure* is defined as the event that causes the delivered service to deviate from correct service [8], [33], [87], [100].

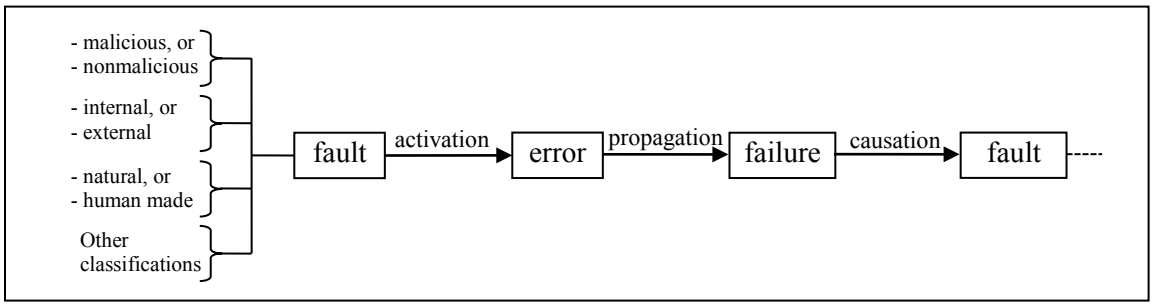

Figure 2-1: The fundamental chain of dependability and security threats, modified from [8]

The complete failure of an entity or a system is not necessary for it not to perform its intended function successfully, as partial failure might still allow operation continuity [47]. The time interval does not have to be always in hours or days, as it depends on the system under study, and thus it could be, as examples, clock time, operation time, or number of cycles. Operating conditions should include information about load and environmental setting. Furthermore, while a failure definition should constantly reflect the deviation from correct service [87], failure modes and its specifics are not necessary identical among the different components [88]. As such, in traditional reliability, failure modes among different hardware components are not necessarily identical; and when reliability study was extended to software engineering, software failure modes were defined differently, reflecting the failure attributes of software components. For instance, failure of an electronic component might be defined as the inability of a thyristor to withstand the *n*th voltage spike; for a mechanical component it might be the event that the strengths of the component are smaller than its stresses [47]; and for software components it might be defined as an unacceptable departure of program operation from requirements, modeled by the mean time to failure (MTTF) based on execution time, not calendar time, such as in the Musa model [89], [106], or it might be modeled based on Bayesian interpretation of probability, such as in the Littlewood model [107]. What matters is the establishment of the qualitative aspects of failure, particularly the full range of component failure modes and the error processes that lead to failure [88].

The combined consideration and utilization of dependability and security attributes has facilitated a better understanding of the fault-error-failure chain model and analysis of various threats that might affect a system [8]. Thus, the traditional definition of failure is increasingly applied to security breaches [9], [98], [108] but in an inclusive analogy of both security breaches and traditional system

failures. The subjectivity issue of security in general and the requirement to fulfill both dependability and security attributes has led to considering their attributes from a probabilistic perspective, as faults and subsequent failures can never be totally eliminated from any system [8]. The reliability characteristics of security systems can consequently be defined and analyzed using the reliability characteristics of its components, i.e., security controls, with the presence of the appropriate system abstraction and failure logic.

However, to analyse failure data, two approaches are generally considered: parametric and nonparametric [103], [109]. Parametric analysis involves the choice of probability distribution first and then the evaluation of its parameters to fit the data available. The choice of a particular distribution is usually made based on similar previous tests or the phenomena basis itself. Probability plotting is used to estimate distribution parameters and represent them graphically. On the other hand, nonparametric analysis involves various techniques such as constructing histograms and calculating sample statistics (e.g., sample mean and variance). In this case, no particular assumptions are made about the underlying distribution, although these analysis tools often provide enough information to allow selection of a suitable distribution afterwards.

Reliability can be increased by decreasing the hazard rate, which represents *the proportion of components in service that fail per unit interval*. If the hazard rate increases with time, the cumulative distribution of the time to failure is defined as Increasing Failure Rate (IFR) distribution. On the other hand, if the hazard rate decreases with time, the cumulative distribution of the time to failure is defined as Decreasing Failure Rate (DFR) distribution. Moreover, if the hazard rate function is constant of time, then it is called Constant Failure Rate (CFR) distribution, which leads to studying the exponential distribution (also known as the negative exponential distribution) [49].

Many researches consider Exponential distribution to be the most commonly used distribution in reliability theory. The main underlying assumption is that the failure rate at which the system, or component, fails is independent of time or use. So, it is suitable for systems that operate, or at least intended to operate, continuously with no significant wearout mechanisms and early defect failures. Although this analysis is not realistic for all time, the approximation of the constant failure rate is sufficiently accepted even though a system, or a component, may experience some early failures or aging effects, as long as it provides a good approximation during the useful time. The effect of early failures is usually treated by quality control measures, while the magnitude of aging effect is usually treated by continuous preventive maintenance and timely replacement policies. For explicit cases

where a system's failure rate varies over time, a constant failure rate that encompasses the whole failure curve might be used to ensure it contains the whole failure variation. So, the use of the constant failure rate can actually be extended to apply to many cases where it would not be the correct theoretical model [110]. Exponential distribution, however, can be applied on a wide range of systems such as aircraft and spacecraft electronics, satellite stations, communications equipment, and computer networks [111].

Nevertheless, the mathematical properties of the exponential model are unique to its definition and important to its wide applications. The failure distribution is completely defined by the knowledge of only one parameter, which is the Mean Time To Failure or $MTTF$, often denoted by θ. $MTTF$ sufficiently defines the only distribution parameter, that is, *failure rate*, and is often denoted by λ. These two variables define each other and are used interchangeably as $\lambda = 1/\theta$. Another useful property of the exponential model is the simple mathematical manipulation of reliability functions, as many calculations involve the integration and multiplication of exponential functions that are easy operations [104], [111], [112]. This model is demonstrated graphically later in Section 2.2.4 in Figure 2-7 and Figure 2-8.

However, time-dependant failure rate models facilitate the study of failure nature across time, whether they are infant mortality failures, random failures, or aging effect failures. These models are suitable for situations where different failure stages need to be treated and analysed explicitly. In contrast to the exponential model used for random failures, such models need at least two parameters to reflect failure behaviour. Normal and lognormal distributions are frequently used in some of these situations, but Weibull distribution is considered the most universal and widely accepted one [109], [113].

Regardless of the failure model in use, a parametric or nonparametric, measured or estimated, appropriate definition of failure (and hence, success) represents a building block before any operational analysis methods can be made good use of.

2.2.2 Reliability Block Diagrams (RBDs) and Structure Functions

In the mathematical sense, reliability is measured by the probability that a system or a component will perform its intended function for a specified period of time under stated conditions [48], [49], [114]. Reliability Block Diagrams (RBDs), as a graphical representation, and structure functions, as a mathematical representation, are commonly used in building and analyzing reliability models.

The Reliability Block Diagram (RBD) is an inductive, event-based method used to graphically depict the logical arrangement of components with respect to failure (or success) of the system under consideration. RBDs allow the identification and enumeration of failure (or success) pathways. RBDs can also be developed in multiple levels, allowing top-level RBDs to be decomposed into smaller ones repeatedly, until reaching the level of abstraction of interest [74], [115].

We now review the so-called indicator, or Boolean, function x_i for unit or component i

$$x_i = \begin{cases} 1, & if\ the\ ith\ component\ is\ functioning \\ 0, & otherwise \end{cases} \tag{2-1}$$

The vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is called the *state vector* that shows the status of each component of being either working or failed. So, for a system of n components, there are 2^n different states determined by the states of individual components. The *structure function* $\emptyset(\mathbf{x})$ of the vector $\mathbf{x}$ for the system as a whole is defined as follows.

$$\emptyset(\mathbf{x}) = \begin{cases} 1, & if\ the\ system\ is\ functioning \\ 0, & otherwise \end{cases} \tag{2-2}$$

The property of systems *coherency* has facilitated the study of system performance by analyzing its components using the structure functions, which dictates that the system is coherent if and only if structure function $\emptyset(\mathbf{x})$ is nondecreasing in each argument x_i for $i = 1, \dots, n$ and every component is relevant [114]. In this sense, $\emptyset(\mathbf{x})$ is a Boolean function of x_i. $\emptyset(\mathbf{x})$ is also a monotonic increasing function of $\mathbf{x}$, which means if $x_i \leq y_i, i = 1, \dots, n, then\ \emptyset(\mathbf{x}) \leq \emptyset(\mathbf{y})$.

A *series* system will function if and only if all of its components are functioning, whereas a *parallel* system needs only one of its components to function. So, for a series system, we write the structure function as

$$\emptyset(\mathbf{x}) = min(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} x_i \tag{2-3}$$

And, the pure series structure can be represented using the reliability block diagram as shown in Figure 2-2.

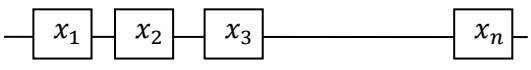

Figure 2-2: System representation of series structure

For a parallel system, the structure function is written as

$$\emptyset(\mathbf{x}) = max(x_1, x_2, \dots, x_n) = 1 - \prod_{i=1}^{n}(1 - x_i) \tag{2-4}$$

And the pure parallel structure can be represented using the reliability block diagram as shown in Figure 2-3.

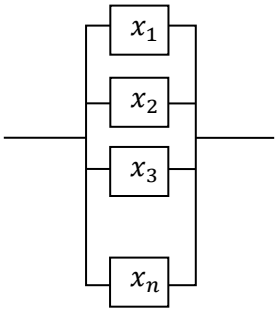

Figure 2-3: System representation of parallel structure

Note that both parallel and series structures are special cases of *k*-out-of-*n* structure, sometimes referred to as a voting system. This structure functions if and only if at least k components are functioning. The structure function of *k*-out-of-*n* structure is written by

$$\emptyset(\mathbf{x}) = \begin{cases} 1, & \sum_{i=1}^{n} x_i \geq k \\ 0, & otherwise \end{cases} \tag{2-5}$$

where $\sum_{i=1}^{n} x_i$ represents the number of working components.

Using this representation, a pure series structure is *n*-out-of-*n* system, whereas a pure parallel is *1*-out-of-*n* system. However, it is unlikely for systems in practice to consist only of either pure series or pure parallel components [48], [113]. Mixed structures are commonly used to represent such systems of mixed components.

Figure 2-4 shows the reliability block diagram of a parallel-series mixed structure in a detailed form in (a) and a more aggregated form in (b). The same equations mentioned earlier can be used to represent these structures as follows.

For all n components in each serial line we first find

$$O_m = min(x_{m1}, x_{m2}, \dots, x_{mn}) = \prod_{i=1}^{n} x_{mi} \tag{2-6}$$

Then we calculate system, $\boldsymbol{S}$, structure function as

$$\emptyset(\mathbf{S}) = max(O_1, O_2, \dots, O_m) = 1 - \prod_{i=1}^{m} (1 - O_i) \tag{2-7}$$

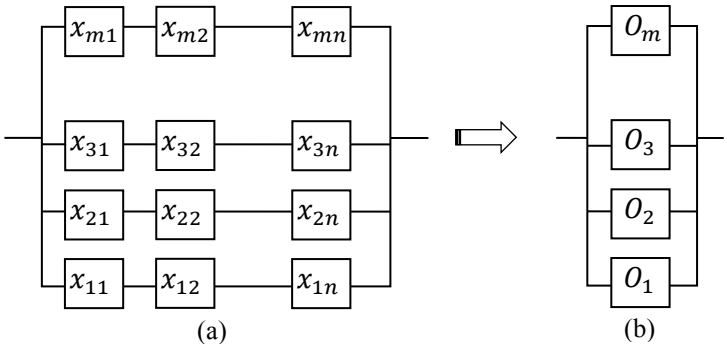

Figure 2-4: System representation of parallel-series mixed structure

Another form of mixed structures is series-parallel. Figure 2-5 (a) shows its reliability block diagram in a detailed form and (b) represents the aggregated form. Again, we first find the structure function of a parallel subsystem as

$$O_n = max(x_{1n}, x_{2n}, \dots, x_{mn}) = 1 - \prod_{i=1}^{m} (1 - x_{in}) \tag{2-8}$$

Then, we calculate the system $\boldsymbol{S}$ structure function as

$$\emptyset(\mathbf{S}) = min(O_1, O_2, \dots, O_n) = \prod_{i=1}^{n} O_i \tag{2-9}$$

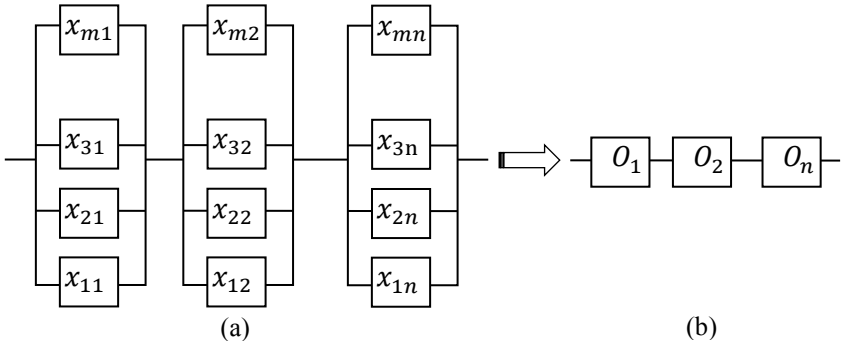

Figure 2-5: System representation of series-parallel mixed structure

One must note that these structures are *reducible structures* into smaller units that simplify the mathematical calculations of their reliabilities. Also, n and m may not be equal in each row and column, respectively, in both mixed structures. Complex structures, however, such as bridge and delta structures are addressed using other simplification techniques, which can be found in [47], [115], [116], and fall outside the contribution of this thesis.

Structure functions, however, have been used for various performance measures, other than the states of structures or systems, such as processing speed and capacity [117], [118], [119]. Nevertheless, it is important to differentiate between the physical and logical arrangements of the system when establishing its structure function and reliability block diagram. The physical arrangement reflects the actual layout of items, which is important in studying physical attributes such as dimension, size, and space. Alternatively, the logical arrangement is drawn with respect to the failure process, and hence plays a major role to studying dependability attributes of the system. A physical network is translated into a logical network based on a solid understanding of the system's physical arrangement, its functional requirements, and operational behaviour. Thus, the logical arrangement does not necessarily match the physical arrangement. For example, the physical arrangement could be represented in parallel, while its reliability block diagram, and hence structure function, are defined by a series structure, and vice versa [104].

2.2.3 Minimal Path and Minimal Cut Sets

The study of structure functions leads to introducing another combinatorial method, that is, minimal sets, which are useful in analysing system functionality and failure states with respect to minimum

components. A *minimal path set* is the minimum set of components that are required to ensure a system is functioning. Further, a *minimal cut set* is the minimum set of components whose failure leads directly to the failure of the whole system. So, minimal path sets and minimal cut sets allow us to represent any system as a parallel-series structure and a series-parallel structure, respectively [47], [48], [105], [114], [120].

For a state vector $\mathbf{x}$ to be called a *path vector*, it is required to have $\emptyset(\mathbf{x}) = 1$. However, it is said to be a *minimal path vector* if $\emptyset(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} < \boldsymbol{x}$ ($y_i \leq x_i, i = 1, \dots n$ and $y_i < x_i \; for \; some \; i$); also, the set $A = \{i: x_i = 1\}$ is called a *minimal path set*. Now, for a given system let $A_1, A_2, \dots, A_s$ denote the minimal path sets. We define a new Boolean function $\propto_j (\mathbf{x})$ of the *j*th minimal path set as follows.

$$\begin{aligned} \propto_j (\mathbf{x}) &= \begin{cases} 1, & if \; all \; A_j components \; are \; functioning \\ 0, & otherwise \end{cases} \\ &= \prod_{i \in A_j} x_i \end{aligned} \tag{2-10}$$

To ensure the system is functioning we need to ensure that there exists at least one of its minimal path sets where all the components are functioning. This leads to the following relationship.

$$\begin{aligned} \emptyset(\mathbf{x}) &= \begin{cases} 1, & if \;\; \propto_j (\mathbf{x}) = 1 \text{ for some } j \\ 0, & if \;\; \propto_j (\mathbf{x}) = 0 \text{ for all } j \end{cases} \\ &= max_j \propto_j (\mathbf{x}) \\ &= max_j \prod_{i \in A_j} x_i \end{aligned} \tag{2-11}$$

Additionally, a state vector $\mathbf{x}$ is called a *cut vector* if $\emptyset(\mathbf{x}) = 0$ and a *minimal cut vector* if $\emptyset(\boldsymbol{y}) = 1$ for all $\boldsymbol{y} > \boldsymbol{x}$. The set $C = \{i: x_i = 0\}$ is then called a *minimal cut set*. Now, for a given system let $C_1, C_2, \dots, C_q$ denote the minimal cut sets. We introduce a new Boolean function $\beta_j(\mathbf{x})$ of the *j*th minimal cut set as follows.

$$\begin{aligned} \beta_j(\mathbf{x}) &= \begin{cases} 1, & if \; at \; least \; one \; component \; of \; the \; C_j \; set \; is \; functioning \\ 0, & otherwise \end{cases} \\ &= max_{i \in C_j} x_i \end{aligned} \tag{2-12}$$

To ensure the system is functioning we need to ensure that there exists at least one component from each of its minimal cut sets that is functioning. In other words, the system is not functioning if and only if all the components of at least one minimal cut set are not functioning. This leads to concluding the following relationship.

$$\begin{aligned} \emptyset(\mathbf{x}) &= \prod_{j=1}^{q} \beta_j(\mathbf{x}) \\ &= \prod_{j=1}^{q} max_{i \in C_j} x_i \end{aligned} \tag{2-13}$$

These sets provide significant use in reliability studies as they define qualitatively lower bounds of the number of components required to ensure system functionality in the case of minimal path sets and system failure in the case of minimal cut sets. The use of this method allows sensitivity analysis of existing system configuration and possible alternatives to be carried out in an efficient manner. We believe the adaptation of these tools in studying security systems can be useful, as critical and weak points in security systems can be identified and addressed proactively.

2.2.4 Reliability Model

As stated before, reliability is measured mathematically by the probability of a system functioning without encountering a failure for a specified period of time. Hence, the reliability function $R(t)$ is defined as the probability that the system has operated over the time *interval* 0 *to* t. In contrast, the availability is the fraction of time the system is up, and hence, the availability function $A(t)$ is defined to be the probability that the system is operating at time t. However, $A(t)$ has no information on how many failure-repair cycles in the interval 0 *to* t. In the case of irreparable systems, $A(t) = R(t)$, and with repairable systems, $R(t) \leq A(t)$ [48], [49], [80], [113], [114], [120], [121]. Both reliability and availability represent two different but related operational measures of system performance. Availability is a combined measure of maintainability and reliability, and it has been widely used as a measure of system effectiveness [122]. Yet, availability calculations require more information about maintenance and logistical support of the system, and so, it is generally more difficult than reliability to attain. For critical applications, however, a reliability measure is the most important and stringent measure [115]. Since we are dealing with security, the reliability measure is the measure of choice in the proposed extensions from dependability theory.

There are two general approaches to studying reliability: i) reliability based on random events and ii) reliability based on random variables. For completeness, we show both approaches.

Reliability modeling based on random events: To demonstrate reliability based on random events, let S denote the event that a system of n components will be functioning, and the event that component i is working is denoted by $x_i, i \in \{1, \dots, n\}$; thus $\overline{x}_i$ represents the failure of component i (the event when component i is *not* working).

Now, if the system components are logically arranged in series, the system will be working if and only if all n components are working. Hence, event S will be the intersection of all $x_i's, i = 1,2, \dots, n,$ events as follows.

$$S = x_1 \cap x_2 \cap \dots \cap x_n \tag{2-14}$$

And the probability of the system working, i.e., reliability of the system, is given by

$$R_s = P(S) = P(x_1 \cap x_2 \cap \dots \cap x_n) = P(x_1 x_2 \dots x_n)$$

If failure events of the system are not independent (all the random events $x_i's, i = 1, \dots, n,$ are dependent), we write

$$R_s = P(S) = P(x_1)\, P(x_2/x_1)\, P(x_3/x_1 x_2) \dots P(x_n/x_1 x_2 \dots x_{n-1}) \tag{2-15}$$

And when failure events are independent (all the random events $x_i's, i = 1, \dots, n,$ are independent), we write

$$R_s = P(S) = P(x_1)\, P(x_2)\, P(x_3) \dots P(x_n) \tag{2-16}$$

So,

$$R_s = \prod_{i=1}^{n} P(x_i) = \prod_{i=1}^{n} R_i\,, \qquad where\ R_i = P(x_i) \tag{2-17}$$

On the other side, if system components are logically arranged in parallel, the system will be working if at least one of the n components is working. This arrangement is sometimes called a redundant configuration. Consequently, event S will be the union of all $x_i's, i = 1,2, \dots, n,$ events,

$$S = x_1 \cup x_2 \cup \dots \cup x_n \tag{2-18}$$

And the probability of the system working, i.e., the reliability of the system, is given by

$$R_p = P(S) = P(x_1 \cup x_2 \cup \ldots \cup x_n) = P(x_1 + x_2 + \cdots + x_n)$$

If failure events of the system are not independent (all the random events $x_i's, i = 1, \ldots, n,$ are dependent), then we write

$$\begin{aligned} R_p = P(S) = [P(x_1) + P(x_2) + P(x_3) + \cdots + P(x_n)] \\ - \left[P(x_1x_2) + P(x_1x_3) + \cdots + P\left(x_ix_j\right)_{i \neq j}\right] \\ + \cdots + (-1)^{n-1}P(x_1x_2 \ldots x_n) \end{aligned} \tag{2-19}$$

The formula above can be written in a simpler way if one deals with the probability of system failure instead. Parallel system failure occurs if all components fail, yielding the intersection operation of all the events of components failures

$$R_p = 1 - P(\overline{x}_1\overline{x}_2 \ldots \overline{x}_n) \tag{2-20}$$

And when failure events are independent (all the random events $x_i's, i = 1, \ldots, n,$ are independent), we write

$$R_p = P(S) = 1 - P(\overline{x}_1)\, P(\overline{x}_2)\, P(\overline{x}_3) \ldots P(\overline{x}_n) \tag{2-21}$$

So,

$$R_p = 1 - \prod_{i=1}^{n} P(\overline{x}_i) = 1 - \prod_{i=1}^{n} \overline{R}_i\,, \qquad where\ \overline{R}_i = P(\overline{x}_i) \tag{2-22}$$

Reliability modeling based on random variables: To demonstrate reliability based on random variables, we suppose that X_i, the state of the *i*th component, is a random variable. So,

$$R_i = P\{\mathrm{X}_i = 1\} = 1 - P\{\mathrm{X}_i = 0\} = p_i \tag{2-23}$$

and is called the reliability of the *i*th component. Also,

$$\begin{aligned} R = P\{\emptyset(\mathbf{x}) = 1\} = 1 - P\{\emptyset(\mathbf{x}) = 0\}, \\ where\ \mathbf{x} = (X_1, X_2, \ldots, X_n) \end{aligned} \tag{2-24}$$

and is called the reliability of the system. Since $\emptyset(\mathbf{x})$ is a Bernoulli random variable, the reliability of the system R can be computed by taking the expectation, that is,

$$R = P\{\emptyset(\mathbf{x}) = 1\} = E[\emptyset(\mathbf{x})] \tag{2-25}$$

For systems connected in series, when all the random variables $X_i's, i = 1, \dots, n$, are independent, we can write

$$\begin{aligned} R_s &= P\{\mathrm{X}_1 = 1\}P\{\mathrm{X}_2 = 1\} \dots P\{\mathrm{X}_n = 1\} \\ &= \prod_{i=1}^{n} p_i \end{aligned} \tag{2-26}$$

And when all the random variables $X_i's, i = 1, \dots, n$, are dependent we write

$$R_s = P\{X_1 = 1\}P\{X_2 = 1/X_1 = 1\} \dots P\{X_n = 1/X_1 = 1, X_2 = 1, \dots, X_{n-1} = 1\} \tag{2-27}$$

Also, for systems connected in parallel, when all the random variables $X_i's, i = 1, \dots, n$, are independent, we can write

$$\begin{aligned} R_p &= 1 - P\{X_1 = 0\}P\{X_2 = 0\} \dots P\{X_n = 0\} \\ &= 1 - \prod_{i=1}^{n} (1 - p_i) \end{aligned} \tag{2-28}$$

And when all the random variables $X_i's, i = 1, \dots, n$, are dependent we write

$$R_p = 1 - P\{X_1 = 0\}P\{X_2 = 0/X_1 = 0\} \dots P\{X_n = 0/X_1 = 0, X_2 = 0, \dots, X_{n-1} = 0\}$$

And the same logic applies to mixed structure systems, i.e., systems that contain parallel and series components together.

Above we have considered the reliability of the system in terms of static probabilities, i.e., probabilities are considered as constants. Systems, however, operate and hence fail over time. As such, it is commonly accepted to model the reliability of systems as a function of time. To show that, let T be a random variable that represents the *time to failure* of the system. The reliability can then be defined in terms of probability of failure as a function of time, as illustrated in Figure 2-6, and written as

$$R(t) = P(T > t) = 1 - F(t) \tag{2-29}$$

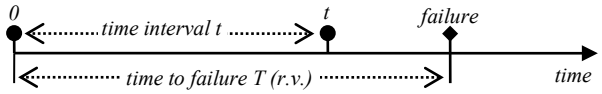

Figure 2-6: The failure process in reliability context

This function means the system will function for time t or greater if and only if it is still functioning at time t. $R(t)$ is a monotonic non-increasing function of t with unity at the start of life: $R(0) = 1$ and $R(\infty) = 0$ [49]. $F(t)$, failure distribution, however represents the probability that the system will fail before time t, and equals

$$F(t) = P(T \leq t) = 1 - R(t)$$

Nevertheless, time-dependent reliability is the technique of decomposing the system into a set of subsystems, or components, where their reliabilities are known or to be computed with respect to time, so

$$R(t) = R(P_1(t), P_2(t), \dots, P_n(t)) \tag{2-30}$$

where,

$$P_i(t) = P\{lifetime\ of\ component\ i > t\} = 1 - F_i(t)$$

As in [123], for a series system with n independent components, we can write

$$\begin{aligned} R(t) &= R_1(t) \times R_2(t) \times \dots \times R_n(t) \\ &= \prod_{i=1}^{n} R_i(t) \end{aligned} \tag{2-31}$$

And, for a parallel system with n independent components, we can write

$$\begin{aligned} R(t) &= 1 - [(1 - R_1(t)) \times (1 - R_2(t)) \times \dots \times (1 - R_n(t))] \\ &= 1 - \prod_{i=1}^{n} (1 - R_i(\mathrm{t})) \end{aligned} \tag{2-32}$$

Yet, one must note that R can be expressed as a function of components' reliabilities, $R = R(\mathbf{p})$, that is, a monotonic increasing function of $\mathbf{p}$, where $\mathbf{p} = (p_1, p_2, \dots, p_n)$ [48].

Regardless of the modelling approaches above, a common structure in reliability evaluation is a *k*-out-of-*n* system with identical and independent components [47], [48], [104], [113], [115]. Considering this system as an application of the binomial distribution, its reliability function is given by

$$R(\mathbf{p}) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{2-33}$$

where $p = p_1 = p_2 = \cdots = p_n$

Note that calculating the reliability of a *k*-out-of-*n* system when its components are not identical can be very complicated procedure, as the state enumeration approach is used to sum up all the probabilities of possible system realisations with the number of the working components is not less than k.

In light of the above, two important conclusions stated in [49], [113], [120] must be taken into account for series and parallel arrangements. Firstly, for a series arrangement system, the larger (lower) the number of components connected in series, the lower (larger) is the reliability of the system. Therefore, the reliability of the $(n+1)$ components series system is upper-bounded by the reliability of the same system having (n) components. So, adding an extra component $(n+1)$ for a series arrangement of n components, say $R_{n+1}(t), where\ R_{n+1}(t) < 1$, we write

$$\begin{aligned} R(t) = R_1(t) \times R_2(t) \times \ldots \times R_n(t) &= \prod_{i=1}^{n} R_i(t) \\ &> R_1(t) \times R_2(t) \times \ldots \times R_n(t) \times R_{n+1}(t) \\ &= \prod_{i=1}^{n+1} R_i(t) \end{aligned} \tag{2-34}$$

Also, the reliability of the system is less than the reliability of its least reliable component; and the system's reliability decreases (increases) if any component's reliability decreases (increases). Therefore, the reliability of the series system is upper-bounded by the reliability of its least reliable component. This feature is analogous to the weakest link of a security system, which is considered a measure of the security strength as discussed before. Thus, considering a series arrangement of n components with $R_s(t)$, where $s \in \{1, \ldots, n\}$, denoting the least reliable component, we write

$$R(t) = R_1(t) \times R_2(t) \times \ldots \times R_n(t) = \prod_{i=1}^{n} R_i(t) < R_s(t) \tag{2-35}$$

Secondly, for a parallel arrangement system, the larger the number of components in parallel, the larger is the reliability of the system. Therefore, the reliability of the $(n+1)$ components parallel system is lower-bounded by the reliability of the same system having (n) components. So, adding an extra component $(n+1)$ for a parallel arrangement of n components, say $R_{n+1}(t), where\ R_{n+1}(t) < 1$, we write

$$\begin{aligned} R(t) &= 1 - [(1 - R_1(t)) \times (1 - R_2(t)) \times \ldots \times (1 - R_n(\mathrm{t}))] \\ &= 1 - \prod_{i=1}^{n} \big(1 - R_i(\mathrm{t})\big) \\ &< 1 - [(1 - R_1(t)) \times (1 - R_2(t)) \times \ldots \times (1 - R_n(\mathrm{t})) \times (1 - R_{n+1}(\mathrm{t}))] \\ &= 1 - \prod_{i=1}^{n+1} (1 - R_i(\mathrm{t})) \end{aligned} \tag{2-36}$$

Furthermore, the reliability of the system is larger than the reliability of its most reliable component. Therefore, the reliability of the parallel system is lower-bounded by the reliability of its most reliable component. Considering a parallel arrangement of n components with $R_l(t), where\ l \in \{1, \ldots, n\}$, as the most reliable component, we write

$$\begin{aligned} R(t) &= 1 - [(1 - R_1(t)) \times (1 - R_2(t)) \times \ldots \times (1 - R_n(\mathrm{t}))] \\ &= 1 - \prod_{i=1}^{n} (1 - R_i(\mathrm{t})) > R_l(t) \end{aligned} \tag{2-37}$$

We conclude this section by reviewing four important functions related to reliability evaluation: (i) the failure density function, which describes how the failure probability is spread over time and denoted by $f(t)$. $f(t)$ is always non-negative and the total area beneath it is always equal to one as it is basically a probability distribution function, so

$$\int_0^{\infty} f(t)dt = 1 \tag{2-38}$$

(ii) The failure distribution function $F(t)$, which is the cumulative distribution function of the failure density function $f(t)$. As mentioned earlier, $F(t)$ represents the probability that the system will fail before time t and gives the area beneath the failure density function until time t, and equals

$$F(t) = P(T \leq t) = \int_0^t f(v)dv \,, where\ v\ is\ a\ dummy\ variable \tag{2-39}$$

$F(t)$ and $f(t)$ satisfy the following relation:

$$f(t) = \frac{d}{dt}F(t) = -\frac{d}{dt}R(t) \tag{2-40}$$

(iii) The reliability function $R(t)$, which gives the area beneath the failure density function after time t and equals,

$$R(t) = P(T > t) = 1 - F(t) = 1 - \int_0^t f(v)dv = exp\left[-\int_0^t \lambda(v)dv\right] \tag{2-41}$$

(iv) The hazard function $\lambda(t)$, sometimes called instantaneous failure rate, which is defined as the limit of the failure rate as the interval length approaches zero. It then equals

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1 - F(t)} = -\frac{d}{dt}[\ln R(t)] \tag{2-42}$$

An important parameter to these equations is the *Mean Time To Failure* ($MTTF$). In this context, it is the expected value of the continuous random variable T and gives the area beneath the reliability function. $MTTF$ is given by

$$MTTF = \int_0^\infty tf(t)dt = \int_0^\infty R(t)dt \tag{2-43}$$

Considering the exponential distribution model as an example, the above functions are demonstrated in Figure 2-7 and Figure 2-8. Given the constant failure rate property, the hazard rate function yields,

$$\lambda(t) = \lambda = \frac{1}{\theta} = \frac{1}{MTTF}, \qquad \text{a constant}$$

The failure density is given by

$$f(t) = \lambda e^{-\lambda t}$$

Similarly, the failure distribution becomes

$$F(t) = 1 - e^{-\lambda t}$$

The reliability function is obtained by

$$R(t) = e^{-\lambda t}$$

And the variance is given by

$$\sigma^2 = \frac{1}{\lambda^2}$$

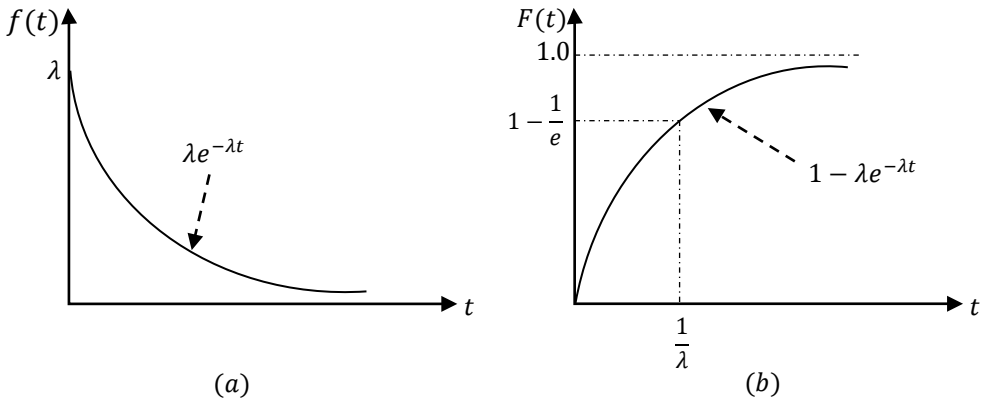

Figure 2-7: The exponential distribution model. (a) Failure density. (b) Failure distribution

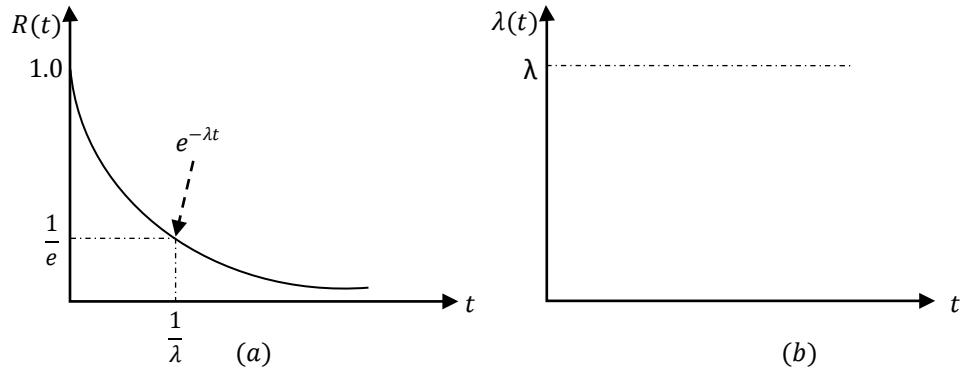

Figure 2-8: The exponential distribution model. (a) Reliability function. (b) Hazard function

2.3 Multi-state System (MSS) Model and Universal Generating Function

Today, there exist a variety of modeling and evaluation methods, be they graphical, probabilistic, or mathematical, or combinations of these, which offer various levels of representation abstracts and different types of measures, and can be useful for performance evaluation exercises of complex systems. In this section, we briefly review the definition of Multi-state Systems (MSS), followed by the Universal Generating Function (UGF) as a mathematical method for studying performance measures of multistate systems using multiple measures. This part represents the basis for the work introduced in Chapter 5.

2.3.1 Multi-state System (MSS) Model

The term MSS is basically used when a system and its components can take an arbitrary finite number of different states of performance levels or rates, ranging from perfect functioning at one side to complete failure at the other side. The fundamental concepts of multi-state system MSS, however, were introduced in the middle of the 1970s [124], [125], [126].

To demonstrate MSS mathematically, consider a system that consists of n units or elements, and assume that any system element i can have k_i different states, mutually exclusive, to encode the discrete scale from complete failure up to perfect functioning. This setting is represented by the set $\boldsymbol{g_i} = \{g_{i1}, g_{i2}, \dots, g_{ik_i}\}$ with related probability of performance for each state $\boldsymbol{p_i} = \{p_{i1}, p_{i2}, \dots, p_{ik_i}\}$, where g_{ij} is the performance rate with associated probability p_{ij} for element i in the state j, $j \in \{1,2, \dots, k_i\}$. In the stochastic domain, let the performance rate at any instance be represented by $G_i(t)$; $0 \le t \le T$, i.e., a random variable taking values from $\boldsymbol{g_i}$, so $G_i(t) \in \boldsymbol{g_i}$ with a probability $P_i(t) \in \boldsymbol{p_i}$ where $\sum_{j=1}^{k_i} p_{ij}(t) = 1$, thus making it a stochastic process for the time interval $[0, T]$ of the MSS operation time [116], [117].

Subsequently, the entire system space can then be seen as having K different states, determined by the states of its individual elements with a total system performance rate $g_i, i \in \{1, \dots, K\}, K = \prod_{j=1}^{n} k_j$. Thus, the MSS performance rate $G(t)$ at any instance t is a random variable too that takes values from the set $\boldsymbol{g} = \{g_1, g_2, \dots, g_K\}$. So, $G(t) \in \boldsymbol{g}$ with related probability of performance for each state $P(t) \in \boldsymbol{p} = \{p_1, p_2, \dots, p_K\}$.

Also, define $\boldsymbol{L^n} = \{g_{11}, g_{12}, \dots, g_{1k_1}\} \times \{g_{21}, g_{22}, \dots, g_{2k_2}\} \times \dots \times \{g_{n1}, g_{n2}, \dots, g_{nk_n}\}$. $\boldsymbol{L^n}$ in this case represents the space of all possible combinations of performance rates for all of the system

elements, and $\boldsymbol{M} = \{g_1, g_2, \dots, g_K\}$ is the space of all possible values of the performance rate for the entire MSS system. The structure function

$$\emptyset\big(G_1(t), G_2(t), \dots, G_n(t)\big): \boldsymbol{L^n} \rightarrow \boldsymbol{M} \tag{2-44}$$

performs a mapping function of the performance rates from the space of all individual elements into the space of the entire system. As a result, the MSS system output performance distribution (OPD) can be defined by the two finite vectors $\boldsymbol{g}$ and $\boldsymbol{p}$ and the system structure function

$$\emptyset\big(G_1(t), G_2(t), \dots, G_n(t)\big), where\ \boldsymbol{p} = \{p_k(t)\} = \mathrm{P}\{G(t) = g_k\}; k \in \{1, \dots, K\} \tag{2-45}$$

MSS representation offers a baseline for defining various measures. Among them is the acceptability function $(G(t), W(t))$, which basically evaluates the ability of MSS to perform a specific task by establishing a relationship between a system performance rate $G(t)$ and some specified performance level, called the demand W. This measure is based on the MSS definition that the space of a system's behaviour is bounded by a set of performance states; thus, one can define two disjoint subsets, defining acceptable and unacceptable system functioning. Practically, MSS performance must exceed the demand to be functioning, and it fails otherwise. So, the function is evaluated as $F\big(G(t), W(t)\big) = G(t) - W(t)$.

The MSS representation provides a generalization of the binary structure function [113], [118], [126]. Various MSS representations and measures for traditional binary-state reliability topologies (such as series, parallel, series-parallel, bridge, etc.) and analysis methods (such as block diagrams, minimal path and cut sets, fault trees, etc.) can be found in [116], [117], [127].

2.3.2 Universal Generating Function (UGF)

UGF is a mathematical technique that allows one to define and solve multi-state system (MSS) output performance distribution using the distributions of its constituting elements. The UGF method was introduced by Ushakov [128], described further in [113], [118], [129], and its applications to reliability analysis can be found in [119], [130]. UGF method is mainly based on using: 1) *z*-transform of discrete random variables, and 2) composition operators, in which the resulting function is called the *u*-function. The method is considered an extension of the widely known moment generating function and thus sometimes called the method of generalised generating sequences [113].

Recall the definition for a discrete random variable X represented by the following probability mass function,

$$P\{X = k\} = p_k, \qquad where\ k = 0,1,2, \ldots$$
$$\sum_{k=0}^{\infty} p_k = 1$$

The *generating function* of the random variable X denoted by $\varphi_X(z)$, as found in [131], can be defined as

$$\varphi_X(z) = \sum_{k=0}^{\infty} p_k z^k \tag{2-46}$$

The coefficients of z^k represent $P\{X = k\}$, and X is a discrete r.v. that can only take integer values $k = 0,1,2, \ldots$, which represent a central restriction to this definition. However, the generating function has the following properties.

To find the distribution of the summation of independent discrete random variables, say $Z = X + Y$, one can simply compute the product of the two generating functions as follows.

$$\begin{aligned} \varphi_Z(z) = \varphi_X(z).\varphi_Y(z) &= \sum_{k_x=0}^{\infty} p_{k_x} z^{k_x} \sum_{k_y=0}^{\infty} p_{k_y} z^{k_y} \\ &= \sum_{k_x=0}^{\infty} \sum_{k_y=0}^{\infty} p_{k_x} z^{k_x} p_{k_y} z^{k_y} \\ &= \sum_{k_x=0}^{\infty} \sum_{k_y=0}^{\infty} z^{(k_x+k_y)} p_{k_x} p_{k_y} \end{aligned} \tag{2-47}$$

This product basically uses the property of polynomials by simply adding the exponents of Z during multiplication procedures, and thus finding $Z = X + Y$. To find the expectation of the *r.v.* X, we write

$$\begin{aligned} \varphi'_X(z)|_{z=1} &= \frac{d}{dz}\varphi_X(z)|_{z=1} \\ &= \left[\sum_{k=1}^{\infty} k p_k z^{k-1}\right]_{z=1} \\ &= \sum_{k=1}^{\infty} k p_k = E[X] \end{aligned} \tag{2-48}$$

Furthermore, for the second derivative of $\varphi_X(z)$ at $z = 1$ we write

$$\begin{aligned}\varphi_X''(z)|_{z=1} &= \left[\sum_{k=1}^{\infty} k(k-1)p_k z^{k-2}\right]_{z=1} \\ &= \sum_{k=1}^{\infty} k^2 p_k - \sum_{k=1}^{\infty} k p_k\end{aligned} \tag{2-49}$$

where the first term is the second moment and the second term is the expectation of *r.v. X*.

The introduction of *moment generating function* $\psi(t)$ [48] removed the restriction of the integer values k of the *r.v. X* on the generating function definition above. Rather, the random variable X is allowed to take all values of t, and can be represented by

$$\psi_X(t) = E[e^{tX}] = \begin{cases} \sum_x e^{tx} p_x, & X \text{ is discrete} \\ \int_{-\infty}^{\infty} e^{tx} f(x)dx, & X \text{ is continuous} \end{cases} \tag{2-50}$$

It also allows higher moments of the random variable to be calculated by successive derivatives of the function $\psi(t)$ and evaluation at $t = 0$. For example, to find $E[X]$, we write

$$\psi_X'(t) = \frac{d}{dt}\psi_X(t) = \frac{d}{dt}E[e^{tX}] = E\left[\frac{d}{dt}[e^{tX}]\right] = E[Xe^{tX}]$$

Hence,

$$\frac{d}{dt}\psi_X(0) = E[X] \tag{2-51}$$

And the generalisation leads to finding $E[X^n]$ by computing the *n*th derivative of $\psi_X(t)$ evaluated at $t = 0$, that is,

$$\frac{d^n}{dt^n}\psi_X(0) = E[X^n], n \geq 1 \tag{2-52}$$

Similarly, the moment generating function of the summation of independent discrete random variables is the product of individual moment generating functions,

$$\psi_{\sum_{i=1}^n X_i}(t) = \prod_{i=1}^{n} \psi_{X_i}(t) \tag{2-53}$$

Subsequently, the *z-transform function* $\psi_x(z)$ of discrete random variable X is defined by substituting $e^t = z$, which can take arbitrary real values in this case for all values of z [116], [117]. It is represented by

$$\psi_X(z) = E[z^X] = \sum_x p_x z^x \tag{2-54}$$

Note that $\psi_X(z)$ inherits the basic properties of the generating function above, particularly,

$$\psi'_X(z)|_{z=1} = E[X] \tag{2-55}$$

and

$$\psi_{\sum_{i=1}^n X_i}(z) = \prod_{i=1}^n \psi_{X_i}(z) \tag{2-56}$$

Now, for complex MSS systems where a large number of components can be connected in different ways, *u-function* is introduced, facilitating the application of more composition operators, i.e., other than the addition of exponents in polynomials, to capture the performance measure of interest. This function represents a subsystem or system as a polynomial, $U(z)$, of a group of smaller components using simple algebraic operations over their individual u-functions, $u_i(z)$, that take the following form

$$u_i(z) = \sum_{j=1}^{k_i} p_{ij}\, z^{g_{ij}} \tag{2-57}$$

Note that in this representation of u-function the coefficients of the terms represent the probabilistic value of some object or state encoded by the exponent of the terms.

For two different components, say g_1 and g_2 with random performance rates $g_1 \in \{g_{11}, g_{12}, \dots, g_{1k_1}\}$ and $g_2 \in \{g_{21}, g_{22}, \dots, g_{2k_2}\}$, their combined $U(z)$ function takes the following form [118], [128], [129],

$$U(z) = \Omega_\omega\big(u_1(z), u_2(z)\big) = \Omega_\omega \left[\sum_{i=1}^{k_1} p_{1i}\, z^{g_{1i}}, \sum_{j=1}^{k_2} p_{2j}\, z^{g_{2j}}\right] = \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} p_{1i}\, p_{2j}\, z^{\omega(g_{1i}, g_{2j})} \tag{2-58}$$

Observe that $u_1(z)$ and $u_2(z)$ represent individual *u*-functions of the components g_1 and g_2, respectively, and $\omega(.)$ function represents the composition operator that reflects the MSS performance measure of interest and relationship between these components. For example, $\omega(.)$ function when defined for capacity measure of an MSS system, $\omega(.)$ of two components equals the sum of the components' capacities when connected in parallel and the minimum when connected in series, denoted by $\omega_p(.)$ and $\omega_s(.)$, respectively, and thus defined by

$$\omega_p(g_1, g_2) = g_1 + g_2 \tag{2-59}$$

and

$$\omega_s(g_1, g_2) = min\,(g_1, g_2) \tag{2-60}$$

Other examples can be found in [118]. In a more general form this representation can be written by

$$U(z) = \Omega_\omega\big(u_1(z), u_2(z), \dots, u_n(z)\big) = \Omega_\omega \left[\sum_{i=1}^{k_1} p_{1i}\, z^{g_{1i}}, \sum_{j=1}^{k_2} p_{2j}\, z^{g_{2j}}, \dots, \sum_{m=1}^{k_n} p_{nm}\, z^{g_{nm}}\right] = \sum_{i=1}^{k_1}\sum_{j=1}^{k_2}\dots\sum_{m=1}^{k_n} \left(p_{1i}\, p_{2j} \dots p_{nm} z^{\omega(g_{1i}, g_{2j}, \dots, g_{nm})}\right) \tag{2-61}$$

Also when $U(z)$ represents the p.m.f. of the *r.v.* X,

$$U_x'(z)|_{z=1} = \frac{d}{dz} U_x(1) = E[X] \tag{2-62}$$

In the case of a *k*-out-of-*n* binary structure, when identical and independent components are considered, one can readily obtain the reliability using the binomial distribution equation in (2-33). When components are not identical, the probabilities of the possible realizations of the structure where the number of functioning components is at least *k* must be summed up. This procedure means

a significant computational cost using the simple enumeration approach of possible states of the structure. Using the UGF method, a straightforward, efficient algorithm, described in [116], [128], [132], can be used as follows.

1. Determine $u_i(z)$'s for the elements of the structure
2. Initialize $R = 0, U_1(z) = u_1(z)$
3. For $j = 2\ to\ n$ do
    a. Obtain $U_j(z) = \Omega_+(U_{j-1}(z), u_j(z))$
    b. If $U_j(z)$ contains a term with z^k, remove this term from $U_j(z)$ and add its coefficient to R

Upon completion of this algorithm, R equals the system's reliability. Note that the function Ω_+ is simply the addition of exponents, i.e., found by the composition operator using $\omega_+(g_1, g_2) = g_1 + g_2$. Also, the elimination of z^k terms occurs because it does not matter how many components are functioning as long as the number is not less than k.

As stated in [116], [117], determining the *u*-function for a system is basically based on a state enumeration approach of its elements, which can be extremely costly in terms of the resources and computations required. To reduce such a cost, first, the like terms for many types of MSS in the *u*-function can be collected as the *u*-function inherits the essential property of the regular polynomial. So, if a *u*-function representing the distribution of a $r.v.X$ contains terms $p_i\ z^{g_i}$ and $p_j\ z^{g_j}$ where $g_i = g_j$, the two terms can be combined into one term $(p_i + p_j)\ z^{g_i}$.

Second, the *u*-function of a higher-level system can be obtained recursively using the *u*-functions of its constituting lower-level subsystems or elements. The recursive determination of the *u*-functions is facilitated by the associative property of the composition operator, which in turn, strictly depends on the structure function having the associative property too in reliability engineering. Further, if the structure function has the commutative property, the composition operator inherits this property, allowing the recursive procedure to contain an arbitrary set of elements with no sense to the order. So,

$$\begin{aligned}&\Omega_\omega\big(u_1(z), \dots, u_k(z), u_{k+1}(z), \dots, u_n(z)\big)\\&\quad = \Omega_\omega\big\{\Omega_\omega\big(u_1(z), \dots, u_k(z)\big),\ \Omega_\omega\big(u_{k+1}(z), \dots, u_n(z)\big)\big\}\end{aligned} \tag{2-63}$$

and

$$\Omega_{\omega}\big(u_1(z),\dots,u_k(z),u_{k+1}(z),\dots,u_n(z)\big) = \Omega_{\omega}\big(u_1(z),\dots,u_{k+1}(z),u_k(z),\dots,u_n(z)\big) \tag{2-64}$$

These two techniques together use the fact that some elements or subsystems have the same performance rates and therefore reduce the length and number of terms of the intermediate elements' *u*-functions. Consequently, they provide a more convenient approach and significant reduction in the computational cost of *u*-functions as opposed to direct calculations of all combinations individually.

Note that the *u*-function differs from regular polynomials in the way that its exponents can be any mathematical objects, as opposed to using scalar variables in polynomials. Furthermore, the *u*-function allows a wider range of $\omega(.)$ operators to be defined over such exponent objects (i.e., not only the product of polynomials used with the ordinary moment generating function or conventional Boolean operators in reliability analysis) [116], [117].

To conclude this section, the UGF method allows one to define performance distribution for a system consisting of multiple levels of smaller components, perform multi-state system performance analysis, and reasonably implement fast optimization procedures. In addition, performance values can be based on various measures such as reliability, availability, speed, or capacity, along with a much more flexible use of composition operators over *u*-functions. Thus, it allows one to capture different topologies, the physical nature of performance, and interactions among system elements [117], [118], [119], [129], [130].

2.4 Asset-Control Graph Evaluation

In this section, we briefly review risk assessment in computing systems, followed by an overview of Bayesian networks. These two subjects represent preliminaries for the proposed evaluation method using the failure relationships among asset and controls in computing systems, which is presented in Chapter 6.

2.4.1 Risk Assessment

Reliability and risk coexist as closely related branches of applied science with a significant overlapping between them. While the first is centered on the analysis of failure and operability, the latter adds to that the study of failure consequences and damage estimates [105].

Risk management methodologies in principle are represented by a series of processes and steps that need to be followed to control risk. These steps are generally centered on risk assessment, risk

mitigation, and evaluation [133]. Risk management methods are, more or less, based on defining the risk as the product of likelihood times impact. In Information Technology (IT), risk assessment methods analyze the combination of assets, threats, and vulnerabilities to calculate the risk level an asset is exposed to. The result of this assessment is fed into the risk mitigation process, in which the most rational, appropriate mitigation measure is selected [133].

Computing paradigms are evolving by nature, leading to what are known today as service-oriented architectures, utility computing, grid computing, virtualization, and cloud computing, among others, representing various types of parallel and distributed systems. While they generally share common conceptual and technological components, they have their own differences too. Whether the change resides mostly at the abstraction level, technological level, service level, or in between, new forms of capabilities and challenges mutate. As such, consistent risk assessment represents one of the main challenges for such evolving paradigms.

Cloud computing, an example on most recent paradigms, has introduced unique sets of computing capabilities and risks at the same time. The “user-centric” perspective of the cloud has redefined how users can control their assets. This paradigm has created a cloud-specific provider-user relationship whereby the provider's computing resources are offered as services to the user based on an on-demand business model.

There exists a wide set of security and privacy issues, not necessarily new, but certainly pertinent to cloud computing. These issues include accountability and auditability boundaries, legal compliance, resource availability and integrity, and data confidentiality and segregation, to name a few [134], [135], [136], [137], [138]. However, we argue that the development of appropriate risk analysis and management methodologies represents one of the areas that necessitate immediate advancement.

The implementation of the cloud paradigm pushes forward the option of risk transference over all other risk mitigation measures: reduction, acceptance, and avoidance. Traditionally, the option of risk transference was set for situations in which the risk has a very high impact and is not easily reduced [139]. However, with regard to the risk transference in the cloud paradigm, we think the transition is heavily influenced by the attractiveness of the economies of scale rather than concerns about the risk manageability. Regardless of whether this observation holds true (or not), appropriate risk assessment methods are much more needed than ever, to manage the anticipated risk of this computing twist.

In light of risk transference, one would want to answer the following questions: how much risk exists in the system? How much risk is being transferred to the cloud? And at what cost or saving to the user? How can we determine the likelihood and impact of risk quantities of resources residing on the cloud provider's side? How can we determine the acceptable probability of failure and its economics between the cloud provider and user?

Most recent work addressing cloud risks and mitigation strategies in particular focuses on mutual regulatory actions between the cloud providers and users, such as those in [134], [135], [136], [137], [140], which are undoubtedly essential and constructive. However, there is still a shortage of literature addressing current risk assessment methods in light of their applicability to various computing paradigms, such as the cloud. For example, the work of [140] recommends stipulating the acceptable risk of failure in the SLA as a predefined threshold for the probability of failure that the cloud provider and user agree on. Yet, finding how to calculate this quantity appropriately remains a research gap.

Earlier, we provided a general review of some work in risk assessment. Several qualitative risk assessment methods, however, are described in [30]. Among them are the Facilitated Risk Analysis and Assessment Process (FRAAP), its variations, and risk assessment matrix. FRAAP is a disciplined process intended to document security-related risks with business operations, conducted for each system, platform, or application at a time. The risk assessment matrix combines the three security objectives, i.e., confidentiality, integrity, and availability, with the two security classes of risk, i.e., accidental acts and deliberate acts. While those methods can be useful in the risk assessment process, they remain qualitative.

The quantitative risk assessment methods, on the other hand, have limitations in common too. For example, the work of [11] presents a model for the probabilistic risk measurement of an enterprise network. To assign cumulative probabilities of successful attacks, it uses attack graphs for model representation and Common Vulnerability Scoring Systems (CVSS) for individual component metrics. The work of [67] also presents a model for quantifying social behavior of an attacker (e.g., skill level, tenacity, financial ability) over the resources of a network as an accumulation of a sequence of steps represented by attack graphs. It then uses Bayesian network (BN)-based analysis to perform vulnerability assessment of the targeted resource. CVSS, however, requires scoring all vulnerabilities in the system to generate the corresponding scores of the impact [62]. In addition, attack graphs require attack templates, configuration files, and attacker profiles as input for each

attack scenario among network nodes [3], [61]. While such models and methods can be useful to the assessment of risk, the enumeration of all possible scenarios with the corresponding likelihood quantities in each configuration remains a complex and too-specific task, or even unattainable, as in the case of a cloud computing setting.

In an attempt to address the aforementioned questions and issues, we propose a quantitative risk assessment method intended to facilitate a better ground for decision makers choosing among alternative risk transference options, and for mitigation measures overall. This method employs Bayesian networks (BNs) to capture and bound the failure dependency among the distinct entities of a computing system. We use a case study involving a cloud computing setting to demonstrate the applicability of the method to various computing paradigms.

2.4.2 Overview of Bayesian Networks

A Bayesian network (BN) is a probabilistic graphical model that represents conditional dependencies among a set of random variables using directed acyclic graphs (DAGs). It is a complete model for the variables and their probabilistic relationships; thus, it can be used for performing various probabilistic inferences that can be very useful to the design and evaluation of a system. The representation of BNs consists of two components. The first component, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, is a directed acyclic graph whose set of vertices[2], $\boldsymbol{V}$, corresponds to the random variables $X_1, X_2, \dots, X_n$, which can be discrete or continuous. Graph edges, $\boldsymbol{E}$, represent the casual dependency relationships among these variables, defining conditional probability statements. The second component, $\boldsymbol{P}$, is the probability distribution over $\boldsymbol{V}$, defining a conditional distribution for each variable, given its parents in $\boldsymbol{G}$ [141].

Consider a BN represented by the finite set $\boldsymbol{X} = \{X_1, X_2, \dots, X_n\}$ of random variables with respect to $\boldsymbol{G}$. Each variable X_i may take value x_i from its domain $Domain(X_i)$[3]. The graph $\boldsymbol{G}$ encodes conditional independence assumptions, which allow the decomposition of any joint distribution into the product form using the chain rule [142], i.e.,

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i / pa(X_i)) \tag{2-65}$$

[2] Vertices and nodes are used interchangeably.
[3] We use capital letters, such as X_1, X_2, for variable names and lowercase letters, such as x_1, x_2, to denote specific values taken by those variables.

$\boldsymbol{X}$ also satisfies the local Markov property, meaning that each variable is conditionally independent of its non-descendants, given its parent variables [142], i.e.,

$$\boldsymbol{X}_v \perp \boldsymbol{X}_{\boldsymbol{V}\setminus dec(v)} / \boldsymbol{X}_{pa(v)}, \qquad for\ all\ v \in \boldsymbol{V} \tag{2-66}$$

Furthermore, BN facilitates many inference forms that are used in the proposed modeling approach, for example, joint distribution queries, which involve calculating the joint probability table between a set of variables. A task of this type is solved by using the chain rule and Markov property, and takes the form,

$$P(x_1 x_2 \dots x_n) = \prod_{i=1}^{n} P(x_i / pa(X_i)) \tag{2-67}$$

Evidence-based queries are used to determine the distribution of non-evidence variables, given some evidence of failure (or non-failure). Inference can be done from children to parents or vice versa. This type takes the form,

$$P(\boldsymbol{X}_{v1} / \boldsymbol{X}_{v2}), \qquad v_1 \neq v_2;\ v_1, v_2 \in \boldsymbol{V} \tag{2-68}$$

In addition, independence check queries are used to discover independency statements among different network nodes. This procedure usually involves adding conditions on some variables to build such independence statements, taking the form,

$$\boldsymbol{X}_{v1} \perp \boldsymbol{X}_{v2} / \boldsymbol{X}_{v3}, \qquad v_1 \neq v_2 \neq v_3;\ v_1, v_2, v_3 \in \boldsymbol{V} \tag{2-69}$$

The choice of graph theory in security evaluation is supported by its abilities to capture complex entity relations and interactions in a simple form [143], [144], along with the availability of well-founded algorithms for the analysis of the model. The choice of BNs approach in particular is supported by its abilities to provide a robust probabilistic method of reasoning under uncertainty and a plausible method for capturing failure dependency relationships in complex systems [65], [142], [145], [146]. To name a few benefits of such an approach, various statistical analysis methods can be engaged on the same system, for example, we can perform different directions of inference (e.g., backward and forward); we can compute the most likely joint distribution of a particular scenario, causal queries, and optimal decisions under uncertainty [32]. Such a direction of research on BNs has even led to their extension to dependability theory, the central field for failure studies. The works of [96], [97] are considered among the earliest studies for estimating reliability using BNs.

BNs have been used to model network vulnerabilities and measure quantitatively network security. The method presented in [58] introduced the idea of using BNs to build attack graphs in a compact form, where nodes represent the state of individual security violations and edges represent the exploitation of vulnerabilities. The goal is to map all underlying potential atomic attack steps exploiting vulnerabilities in a given network. The resultant model is called a Bayesian attack graph. Another work addressing a similar problem was [57], which introduced a quantitative model for network vulnerability assessment, modeling attack graphs as special Dynamic Bayesian Networks (DBNs), and employing CVSS scores for measuring individual vulnerabilities. The overall goal is to derive a measurement of security based on combining various temporal aspects of vulnerabilities such as the availability of exploit codes or patches.

Examples using Bayesian networks in the risk management field, not necessarily in the computing one, can be found in [78], [147], [148]. In [78] in particular, although the goal is to model operational risk in financial institutions using Bayesian networks, a demonstration example of an online business network is presented. The example models various risk factors leading to overall financial loss distribution of the network using random variables, such as application failure, data loss, hacker attack, network failure, server hardware quality, and virus attack. In [147], expert elicitation techniques were used to probabilistically model and forecast nanomaterial risk, as an appropriate dataset including network structure and nodes relationships were not available. Similarly, expert data were used in [148] to build the corresponding BN to model various strategic and legal risks in building important structures.

Chapter 3
The Information Security Maturity Model (ISMM) Extended

3.1 Introduction

This work adopts a system-level approach to address the security quantification problem. To do so, we first introduce a candidate paradigm shift for security modeling that can serves as the foundation for various system-level representations and evaluation techniques. This shift basically starts with what defines a computing system and failure model before any system-level analysis can be established. Using this foundation, we propose two failure-centric model-based quantification approaches, capturing different system characteristics for different evaluation purposes. The first approach addresses the quantification problem considering the set of controls; and the second addresses the problem considering the sets of both assets and controls. Each approach includes a bounding system model, performance measures, and evaluation techniques. To achieve this, we adopt various network and system modelling, reduction, and evaluation techniques available today to represent, quantify, and analyze system-level security. The presented system models, however, are independent from the presented performance measures and evaluation techniques, facilitating a wider range of applications in security evaluation studies.

This chapter covers two main components: 1) the candidate unified abstraction of computing systems and the associated failure model, which represent the foundation upon which the subsequent work throughout this thesis is built; particularly, system models, performance measures, and analysis methods. As such, this part, which is presented in Section 3.4 and Section 3.5, is used throughout the remaining chapters. 2) The proposal of the first system model, which represents a modified version of the original ISMM work. This model is redefined to fit system-level quantitative modeling. The modification covers two main parts: model architecture and maturity propositions so that quantitative maturity can be established, and model structures so that a system-level representation and operational performance measures, mostly from dependability theory, can be established. Using this model, two different groups of evaluation methods are individually presented later in Chapter 4 and Chapter 5.

The overall organization of this chapter is as follows. We start with a brief overview of ISMM-based modeling approach in Section 3.2. We then set out some necessary definitions pertinent to ISMM model extensions in Section 3.3. This is followed by the proposed computing system

abstraction and failure model in Section 3.4 and Section 3.5, respectively. Then the ISMM model architecture is revised accordingly in Section 3.6 and its propositions in Section 3.7 to facilitate the presented quantification approach. This is followed by ISMM-based RBDs, vectors, and structure functions in Section 3.8, along with the properties of quantitative measures and maturity functions in Section 3.9 and Section 3.10, respectively. In Section 3.11, we briefly present a case study demonstrating the contribution to this end, which will then be used to demonstrate analysis using the reliability approach and the Multi-state System (MSS) representation approach using Universal Generating Function (UGF) method in Chapter 4 and Chapter 5, respectively.

3.2 Approach

Security systems today are characterised by their structural complexity. The security element of a computing system is defined by the collective strength of its lower-level constituent controls that are intended to work together in the face of failure. These controls can be in any physical arrangement, such as parallel, series, or meshed, separated sometimes, or even without any systematic aggregation to a particular relevance. Yet, operationally they are interconnected in a particular arrangement with respect to failure, controlling the state and degree of functionality of the overall system. This setting leads to building structural relationships among these controls, which can be identified and analysed; and therefore, can be useful in building more reliable, secure systems. It is remarkable, however, that the relationships between the states of individual controls and the overall system are not black and white. That is, the functionality of a control does not guarantee the functionality of the overall system; failure of a control does not necessary lead to complete system failure either; also, individual controls, and hence the system as a whole, might still continue to perform its intended functions with certain failures at some decreased level of ability, in a way similar to some found in reliability engineering [47].

This work employs these structural relationships among security controls and demonstrates how the original work of the ISMM model can be used to establish a quantitative system-level model. This chapter, however, required the extension and integration of various studies, leading to the steps depicted in Figure 3-1: system-level abstraction[4] into assets and controls, introduced in Section 3.4; the failure model from dependability theory, presented in Section 3.5; the bounding system model

[4] In [151], a system is abstracted into *assets* and *access controls* for the purpose of modeling failure interdependency in access control models. In this work, the abstraction is set into *assets* and *controls* of all types. Thus, it is a generalisation of the earlier work.

using the redefined work of ISMM, presented in Section 3.6 through Section 3.10; and reliability- and MSS UGF-based evaluation techniques, presented separately later in Chapter 4 and Chapter 5. The first two components of this approach, i.e., system abstraction and failure model, have been published in [149], [150], [151] to introduce an asset-control graph and evaluate failure dependency in access control models.

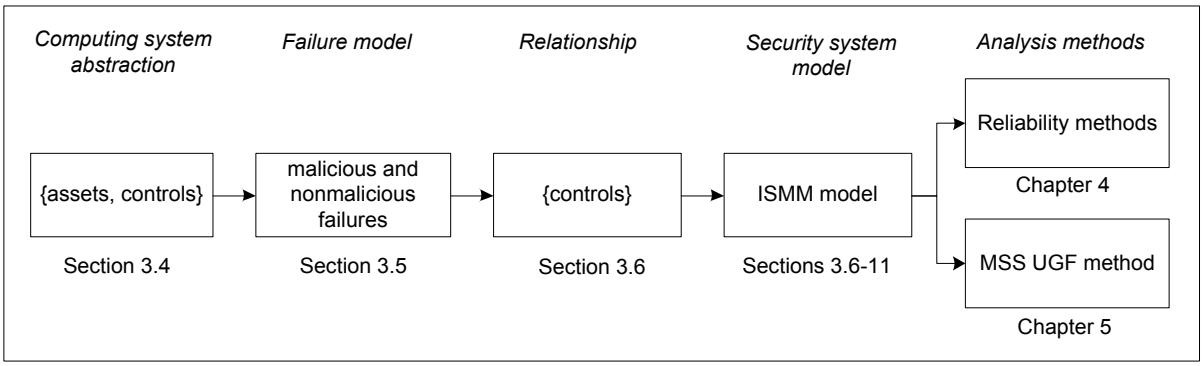

Figure 3-1: Approach of ISMM-based analysis

3.3 Notation and Definitions

This section introduces the extended ISMM notation and definition necessary to this study. A summary of the notation used mainly in Chapters 3, 4, and 5 is outlined in Appendix A.

Assets and controls: these two computer security terms are especially necessary to the abstraction paradigm of a computing system and subsequent evaluation methods [23], [149], [150], [151]. *Assets* can be anything that has value to the organization, its business operations, and their continuity; and *controls* are technical or administrative safeguards or countermeasures to avoid, counteract, or minimize security risks [152]. In short, controls are the protection mechanisms of assets, noting that controls themselves are also part of the total system assets.

Effect and impact: the notion *effect* when used with security controls reflects input to operational security that increases the security strength or level, whereas *impact* is usually used with failures to reflect input that decreases security. These notions are important to distinctly depict the game-theoretic behavior of the security dilemma between defenders and challengers (e.g., attackers or

adversaries, unintentional failures) [22], [23], [149], [151]. These notions are employed to capture the contribution of individual security controls to the state of the security system from both confronting worlds onto the same model foundation.

ISMM-based System: the set of five ordered security layers or subsystems that map the structures of all of the system controls. These layers are structurally connected in a pure series arrangement, defining the conceptual boundaries of the overall security system.

$layer_i$ *or* $subsystem_i$: the set of security controls that are logically interconnected in series, parallel, series-parallel, parallel-series, or mesh structures, or any combination of these, within the logical boundary of layer i.

ISMM structure: any set of controls having a particular logical arrangement with respect to failure. Structures may represent higher-order abstraction such as system- and subsystem-levels or lower-order abstraction such as a collection of constituent components that form a control.

Simple structure: a structure for which a reliability block diagram exists and is reducible to series/parallel form with independent items.

Reliability: the probability that an item will perform its intended function under given conditions for a specified interval of time.

Binary system: a system and its components that are only allowed to take two possible states: either working or failed.

$MTTF$: mean time to failure, which is the expected value of a control's failure-free operation time. Its unbiased (empirical) estimate can be found by $\overline{MTTF} = (t_1 + t_2 + \cdots + t_n)/n$, where $t_1, t_2, \ldots, t_n$ are observed failure-free operating times of statistically identical items. For constant failure rate λ, $MTTF = 1/\lambda$.

Mission time: the stated working time of a component, control, or subsystem.

Failure: deviation of a component, control, subsystem, or system from its correct service, not necessary the specification. Failures are events that are considered with respect to their impact (actual occurrences), not the underlying mechanisms (such as faults and errors, which are states), thus always appearing in time although other variables such as effort can be used.

Item: any component, control, subsystem, or system that can be represented as a structural unit. It may consist of hardware, software, policies, procedures, human resources, or any combination of these.

Redundancy: existence of more than one means for performing a required function in an item. Active (hot, parallel) redundancy is considered in this work, not warm (partially loaded) or standby redundancy.

Multi-state system: a system and its components that are allowed to take more than two possible states, e.g., completely working, partially working (or failed), and completely failed.

Measures: ways to quantitatively represent important performance facets of the actual security state or behaviour using a mathematical abstraction of the problem. A measure, however, cannot capture all the properties of interest. Therefore, the art is to define a suitable subset of those properties in such a way that is simple, focused and yet large enough. As such, many researchers consider that performance evaluation, to a large extent, is an art [36].

Similarities can be found between a number of terminologies used in conventional reliability and security [7], [8], [9], [12], [21], [100]. In this work, however, the following concepts are used analogously: *security breach* and *failure*; and *control* or *security control* and *component*.

3.4 System Abstraction

As described in [153], computing systems are, per design, built based on multiple levels of abstraction, a property that is essential for their dependability while in operation. When descending recursively down the components of a computer, components enlarge themselves to another level of abstraction with a lower-order set of constituting components until reaching the lowest level, perhaps individual transistors. In this sense, failures can usually be tracked down to a single component at some level, which might just represent an error at a higher-level of abstraction.

The essence of the presented modeling and evaluation approach is centered on analyzing the failure relationship between assets and controls when both are seen as the distinct abstractions of the system under study. This level of abstraction represents the infrastructural paradigm facilitating the bounding of system model, and the establishment of their structural relationships with respect to failure, performance measures, and associated functions.

Thus, components of a computing system are abstracted into two main classes: components whose primary purpose is to perform system functions or tasks, denoted as *assets*; and components whose primary purpose is to implement system countermeasures, denoted as *controls* [149], [150], [151]. The controls themselves are also part of the total system assets as per their primary definitions in Section 3.3. Assets can be low-level objects such as data files or high-level ones such as database and application platforms. Similarly, controls can be low-level (i.e., integrated) mechanisms such as authorization modules or high-level (i.e., standalone) ones such as firewalls. Figure 3-2 demonstrates this relationship.

The relationship between assets and controls is unavoidable for any system, and its importance cannot be overstated. In principle, security controls define the domain and nature of countermeasures in the system. The failure of such controls, however, leads to the exposure of system assets consequent to the unavoidable dependency between system assets and controls, and thus could lead to catastrophic system damages. Furthermore, the failure behaviour itself can be very complicated. Controls may fail separately or jointly in various forms due to malicious or nonmalicious causes. For example, an authorization module (perhaps as an access control) on a particular database platform (as an asset) may fail through design flaws (as a nonmalicious failure) or brute force attacks (as a malicious failure), with or without corrupting the database itself, and may be detected or remain undetected.

Nevertheless, the failure of any system component, or a set of components, could extend to affect other components as a result of the interdependencies involved with those components. Therefore, the identification and examination of these interactions (i.e., control-control and asset-control relationships) are crucial to understanding and controlling the security behaviour of a system. Observe that we employ the same long-used classical taxonomy of these security terms but from a different perspective, thus only changing the paradigm of their use.

To the best of our knowledge, there is still a shortage of literature addressing such relationships quantitatively. The failure definition, however, must be refined more appropriately so that this relationship can be captured and analyzed.

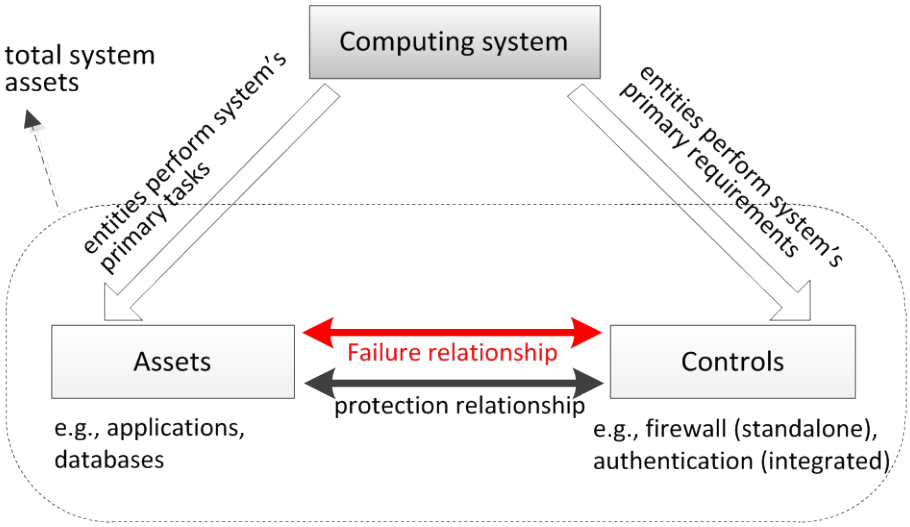

Figure 3-2: Typical entity abstracts in computing systems

3.5 Failure Model

Following system abstraction, the definition of failure (or breach) becomes imperative. The identification and evaluation of failure is a key component in all security-related studies, including risk and extensions from dependability studies [7], [8], [9], [19], [47], [75], [98], [108], [143], [154], [155]. The definition is usually based on either the impact of security failure or the underlying failure mechanisms. The tendency, however, in most of the current quantitative security models is toward addressing underlying failure mechanisms (such as those surveyed in [7]), thus failing to generalise to wider system-level analysis. Regardless of that, appropriate failure models lead to building appropriate security measures, including technical-based and economics-based ones, which in turn, lead to building appropriate quantification and evaluation methods.

The failure definition adopted in this work, which first appeared in [149], [151], is a representation analogous to failure in conventional reliability: the deviation from correct service [8], [98]. Two key aspects, however, are added to this definition: first, it includes failures from normal operational use and malicious activities on security systems [9], [98], [108]; and second, the level of abstraction of failure is based on the impact, not on the underlying failure details [7], [23]. The correct service is delimited by meeting the principle goals of security: confidentiality, integrity, and availability. This

definition is central to the evaluation methods of this research. It is inclusive in terms of its failure domain, and it reflects the consequence or impact in terms of its failure level of abstraction.

Studying failure impact as opposed to the underlying details adds great benefits to many system-level security analysis exercises and to this direction of research in particular (see Figure 3-3). It allows us first to shift the problem domain from modeling failure details at the failure source (e.g., an attacker's mechanisms, accidental fault scenarios; corresponding probabilities of threats, vulnerabilities, exploits, and associated risks; etc.) to the domain of failure consequences or impacts as seen by the system owner or protector (e.g., frequencies and classes of failure, corresponding risk paths and associated impact on assets and controls, etc.). This shift leads to a more plausible, less-complex statistical problem as a result of the bounded view of failure. Second, modeling attack effect as opposed to attack details themselves allows us to incorporate more failures into the model, covering larger classes of attacks, including zero-day attacks [7]. Third, modeling actual consequences of failures, including attacks, facilitates a direct linkage of actual impact of risks with associated economics of security [19], [98], whereby both protection incentives and the failure suffer (i.e., incentives to protecting a system and the suffering from its failure [143]) can be aligned more efficiently and effectively. This facilitation occurs because risks are risks regardless, whether caused by malicious activities or traditional failures. In the end, a failure will have an impact on operational security, and that impact is what needs to be controlled [154]. Thus, attacks could, arguably, arrive at a random point in time from the system owner's perspective, and be manifest as traditional failures [155]. Fourth, in practice, with the current transition toward cloud computing, attaining knowledge or control of the underlying failure dynamics, including failure topology and threat models, will become far more difficult than assessing the impact of failure (e.g., failure statistics). This transition makes it imperative to have models that analyze and evaluate security using this level of abstraction, that is, the impact of failure. Fifth, adopting impact of failure allows us to depart from the ongoing debate in the security community, described in [7], [8], [9], about the plausibility of modeling the class of intentional attacks using probability theory.

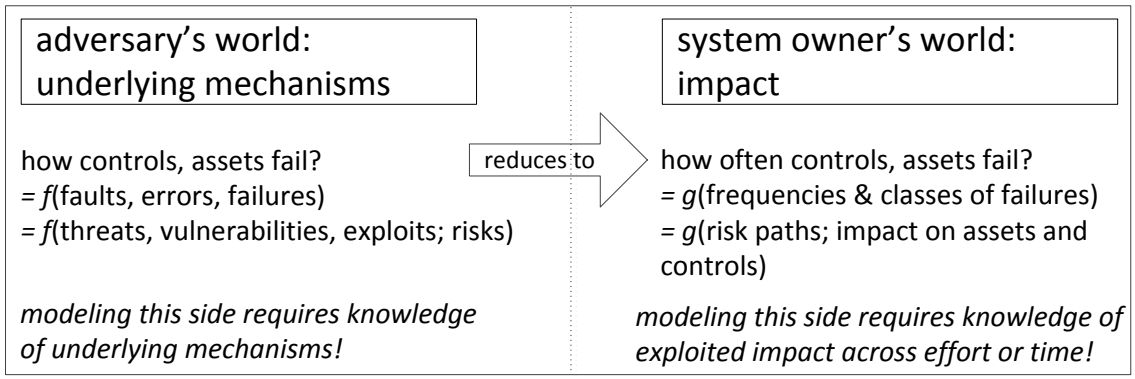

Figure 3-3: The transformation of failure space from underlying mechanisms into impact

While system failure in conventional reliability is commonly modelled as a function of time, as components operate and fail over time, reliability in the security context can be more complicated. When addressing security problems by applying probabilistic techniques to underlying mechanisms of malicious scenarios, beside the direct use of the variable *time*, the variable *effort* is sometimes used to represent the amount of effort expended to breach the system [3], [9], [10], [54], [64], which eventually leads to constructing the analysis model over the variable time. This approach, however, is because the notion of effort is seen to be more inclusive of malicious characteristics such as the required attack tools, time, and computational power [3], [9], [54]. On the other hand, when impact of failure is considered, defining failure events over the variable time becomes a more rational choice because that impact of failure implies the manifestation of failure events, as opposed to underlying failure mechanisms, which implies failure behavioural characteristics. For convenience, we will use the variable *time* to define the failure process, i.e., time taken for a failure event to occur. However, this is not a restriction to the application of this work as effort-dependent random variables can be used in a similar manner. Figure 3-4 illustrates this failure process.

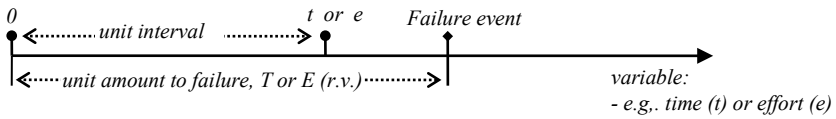

Figure 3-4: The failure process in security context

3.6 ISMM Model Architecture

Maturity as a notion is connected to staged growth with respect to a particular reference of interest. The concept of the maturity model was first developed in 1943 by Maslow in his theory of human motivation [156]. In the computing field, the pervasive use of computers and information systems started in the 1960s made the software development process a real challenge for many IT projects to succeed. Subsequently, the concept of the maturity model was applied in IT by the development of the stages-of-growth model in [157], quality management maturity grid in [158], followed by the software development process in [159]. The concept was later adopted more extensively by the Software Engineering Institute (SEI), funded by the U.S. Department of defense, at Carnegie Mellon University through the development of the Capability Maturity Model (CMM) [85], [160]. Many derivatives then found their way into other processes of various engineering fields and application domains, including the Systems Security Engineering Capability Maturity Model CMM (SSE-CMM/ ISO/IEC 21827).

The connection, however, between the original maturity model of Maslow and subsequent maturity models we have today lies in the context of the primitives used in these maturity models, making them a sort of a mutation from the original work of Maslow's theory. In particular, these primitives include the boundedness of the needs or requirements of the subject, the conceptual layers, the order of needs, maturity promotion and demotion, and self-actualisation.

It is remarkable that the maturity models we have today have always been of a subjective nature, using qualitative measures along the evaluation process. They measure how good a process of interest is [157], [160], as opposed to the actual delivery of that process. In this work, we address maturity both qualitatively and quantitatively all together, showing how maturity concepts and principles can be further extended and formulated mathematically. Thus they can become useful to provide a consistent and integrated approach of quantitative evaluation, through application in information security.

ISMM model semantics: The original work of ISMM was first published in 2003 [161], when the maturity model as a qualitative one was developed and validated. This work was then summarised in [2] and reprinted in [22]. Later, the work was revised in [23] to model structures of security controls, map controls into a sort of layered security architecture, and facilitate the quantification approach afterwards. The work of ISMM herein is further extended to make it a system-level quantification model suitable for establishing various performance measures of the security element. This extension

is mostly connected with the application of dependability theory in security studies. The connection manifests in many ISMM components such as model notation, definitions, and assumptions, and reliability analysis methods and techniques such as structure functions, reliability block diagrams (RBDs), and minimal path and cut sets.

ISMM is a security-centric model, primarily developed to classify and evaluate the maturity levels of information security from a qualitative perspective, which is affected by people, process, and technology, in any computing environment [2], [22], [23], [161]. It provides a means to analyze at which layers and to what extent the integrity of the three main protection processes of security—prevention, detection, and recovery—are realized, and what functions of information security—confidentiality, integrity, and availability—are implemented. Figure 3-5 demonstrates the five layers of the model, indicating their precedence and relative visibility and sophistication of controls across the hierarchy. In this work, we extend the model to quantitatively bound security systems and measure performance using well-founded analysis techniques mostly adopted from reliability theory. The resultant work makes the model a formal, systematic, and comprehensive approach to qualitatively study a system's behavior, strengths, and vulnerabilities and to quantitatively measure the system's operational performance.

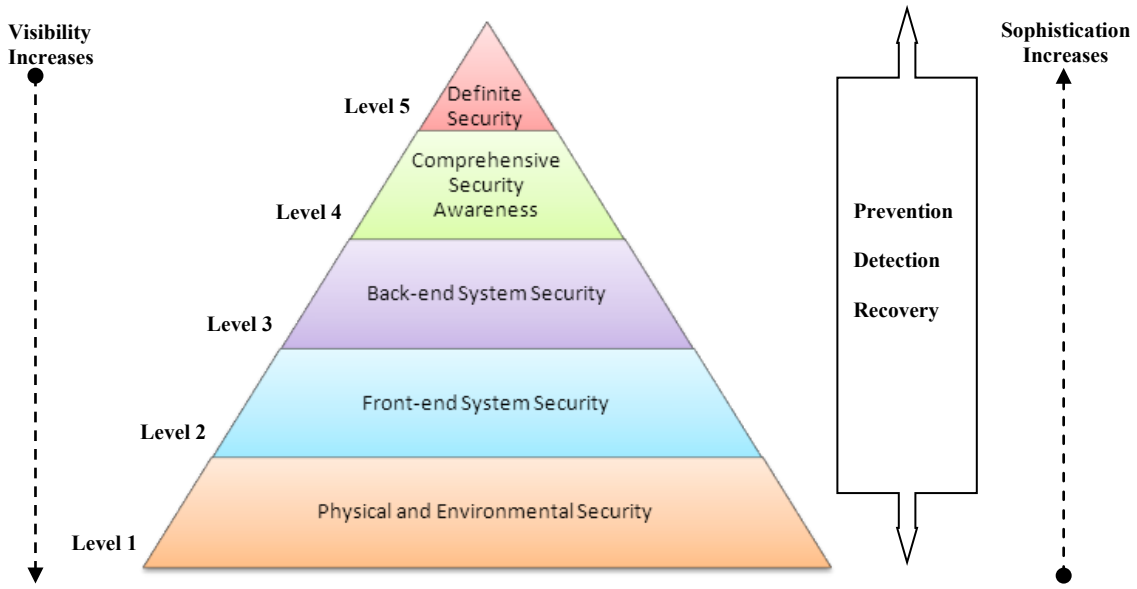

Figure 3-5: The Information Security Maturity Model (ISMM) [2], [161]. Source: ©2006, *Information Systems Control Journal*, vol. 3, p36. All rights reserved. Used by permission.

The benefits and uses of the model can be better described from three different perspectives, or dimensions, as follows

The layering dimension: A layer is basically a conceptual security boundary, or domain, which consists of a collection of controls that preserve a particular structure of failure relationships, serving common layer-level security functions and sharing common layer-level security goals. The layering dimension is demonstrated on the model by five consecutive layers that capture the conceptual boundaries of security systems, including social and technological aspects. Together the five different, yet interrelated, security layers form a particular hierarchy of system-level security functions and goals, starting from Physical and Environmental Security at the bottom to Definite Security at the top, as summarised in Table 3-1. Accordingly, a closed form representation of a security system can be established based on these ordered layers of interrelated controls. These layers are further described as follows.

Layer 1: Physical and Environmental Security. This layer represents physical controls that aim to prevent unauthorized access or interference with the organization or ICT equipment and information assets (e.g., physical access controls and CCTV systems). The maturity at this layer is measured by the extent to which its physical and environmental boundary is protected. These controls generally require less technical knowledge and sophistication and tend to be more visible compared to the other higher layers. Therefore, the sophistication index is considered the lowest, and the visibility index is the highest at this layer.

Layer 2: Front-end System Security. This layer bounds the application-level system functions and integrity requirements between the application component itself and end-user interface. Controls here aim to protect application data from any potential threats that might cause loss, damage, or unauthorized access by either internal or external users (e.g., application access controls). The maturity at this layer is measured by the extent to which front-end information is protected. The visibility index decreases here compared to level-1, as the number of people who can be exposed to this boundary is less. In addition, the control complexity is more, and thus its index is higher.

Layer 3: Back-end System Security. Layer 3 delimits any resource (e.g., hardware, software, or process) that is beyond application-level components. The underlying network infrastructure and internal and external communication devices constitute the major components at this boundary. Controls aim to provide adequate protection of such resources (e.g., data and communication cryptographic mechanisms). The layer maturity is measured by the extent to which back-end

information is protected. This setting suggests that the visibility and exposure element of such controls decreases compared to the lower levels, as equipment is presumably placed in less accessible areas. The required depth of knowledge is also usually more than the requirement at the lower layers, Therefore, the sophistication index increases.

Layer 4: Comprehensive Security Awareness. This layer captures the awareness element of people, which is basically manifested in their practices and behaviours. It includes all technical and nontechnical controls that persuade the organization to operate in a security-conscious culture in all technologies deployed across all lower layers, as the right awareness leads to the right practice. The maturity of this layer is measured by the extent to which security behaviour is protected. Since the awareness of people is an unseen element (it is being demonstrated by their behaviours), ISMM suggests that the visibility index is the lowest compared to the lower layers. Moreover, as this level is solely about people and people alone, the knowledge and management requirements are greater; therefore, the sophistication index is higher than that of the lower layers.

Layer 5: Definite Security. This layer captures the subjectivity issue of information security and its total and continuous resilience, optimization, and improvement against threats across all the lower layers. As is known, there is no such system that is absolutely secure (i.e., 100 percent secure). Therefore, this layer is about convergence of lower-layer controls, which is realised through the depth of applied security knowledge, manifested culture of accountability, and confidence in the organization. Its maturity is measured by the extent to which reliable cross-layer, integrity-specific controls are sustained. As a result, it is never fulfilled completely and works as a ceiling for any security system. Example controls are risk management, business continuity planning (BCP), and disaster recovery planning (DRP) processes. The risk management process, for instance, is not a single function of level-1 only; ideally, it is rather required to be deployed to cover the other levels in the hierarchy. Such proposition helps to define exactly the relevance and scope of applicability of such a requirement at individual layers and collectively, avoiding the false impression of its completeness when that is not the case. Obviously, the sophistication index is the highest and the visibility index is the lowest among all the other layers.

The idea of this layering concept in studying security systems is significant for many reasons:

1) It is used to define an overall boundary of a security system in a hierarchy, reflecting the effect of controls and impact of failures (contribution) at every level of abstraction we examine, i.e., control level, subsystem level, and system level. A key advantage to introducing the notions of

effect of security controls and *impact* of failures is to further augment the boundary drawn on the operating environment of the security system, as in conventional reliability theory. This addition allows all possible inputs of control effects, i.e., a control's input space, and all possible inputs of variable failure classes, i.e., failures input space, to be captured on the same model foundation, regardless of their underlying details. Thus, it allows conceivable structural relationships of controls to be constructed, transforming the complex statistical domain of underlying protection and failure mechanisms into the less complex domain of caused effect and impact, respectively.

2) It is employed to build the notion of precedence among controls, reflecting a rational order of protection and failure of the system from both defenders and challengers.

3) The layers are used to build accordingly logical structures of controls, in which quantifier functions for performance measures of interest and the consequent maturity function can be defined.

4) The systematic aggregation of controls allows us to study consistency and adequacy requirements of comparable strengths in security implementations and to detect their deficiencies, covering both theoretical specification and operated configuration. This is important because information about how security controls might fail in both theory and practice is essential to security designers [18]. NIST, for instance, has published a series of *Special Publications* that give guidance for the strength of various cryptographic algorithms and key sizes when implementing cryptographic primitives. SP 800-57 guidelines give recommendations for a select set of applications, among them, PKI, IPsec, TLS and S/MIME, which are implemented at various layers on the TCP/IP protocol stack and can be used by many platforms and applications. The guidelines state that algorithm suites that use non-comparable strength algorithms are not recommended. The weakest link, i.e., weakest algorithm and used key size, determines the strength of the resultant protection of the system [5]. Clearly, this emphasis for comparable security strengths by NIST SP800-57 extends the requirements set for such cryptographic algorithms beyond their design to operational configuration among individual crypto primitives, that is, strength while in operation. A key question that arises is how to meet such a requirement for such algorithms that might be running on different TCP/IP layers, platforms, or applications of the security system? A more generic question would be how to meet the requirement of comparable security strengths across all controls in the security system? These questions lead to realising the importance of system-level, layered security architecture that captures the logical

dynamics among controls with respect to both their strengths and failures, and for both designed and operational security.

5) Furthermore, this work facilitates the segregation of expertise and analysis methods and tools required for each security boundary. That is, physical security engineers and attackers need knowledge and analysis tools mostly focused on the physical-specific domain to address issues such as hardware architectures and side-channel and differential attacks. The front-end boundary requires more expertise and tools in software architectures. The back-end boundary requires more knowledge of cross-layer messages in communication protocols. The comprehensive-security awareness boundary requires more tools and knowledge in analyzing the behavior of people and social engineering, and finally the Definite layer suggests that more exploitation is required into integrity elements among all types/classes of controls, including physical, technical, and administrative, and involving software, hardware, or people.

The process dimension: This dimension is used to reflect the three fundamental security protection processes, i.e., prevention, detection, and recovery, on every abstraction of the system. In other words, controls can be classified according to these three processes and are assumed to be adequate on all relevant layers before a particular maturity level can be reached. This perspective is useful when examining the adequacy and deficiency of controls during security assessment exercises.

The human dimension: This dimension is employed to reflect people's interactions across security boundaries with respect to sophistication and visibility indices. The visibility index is used to depict the scope of exposure of such security controls on people, which generally increases as one moves down the hierarchy. The sophistication index represents the depth of knowledge, total cost of ownership (TCO), and management requirements, which increase moving up the hierarchy. Both the process and people dimensions are used to further explain the insights of the qualitative part of the model.

Table 3-1: The ISMM model security boundaries and protection goals

Conceptual boundary	Layer	Subsystem	Protection goal
definite	Definite security	$Subsystem_5$	convergence
awareness	Comprehensive security awareness	$Subsystem_4$	Behaviour
logical/technical	Back-end security	$Subsystem_3$	back-end information
	Front-end security	$Subsystem_2$	front-end information
physical	Physical and environmental security	$Subsystem_1$	physical assets

3.7 ISMM Model Propositions

In light of previous work and as part of our revision of the ISMM model, we revisit the model propositions so that maturity qualification can be represented by a mathematical figure in agreement with maturity concepts and common security principles. Doing so allows us, as shown later, to establish measurement methods extended from well-founded theories. The revised propositions are summarized in the following points.

- There is an implicit layer at the bottom of the hierarchy, called ad-hoc security or level 0, meaning lack of any recognition of security issues and respective controls. This layer works as the floor for any security system.
- Reachability (promotion) criteria: maturity at a given layer is reached by implementing adequate and quality prevention, detection, and recovery controls on that layer and its preceding layers, if they exist (principle of effectiveness). As a result, skipping levels violates maturity promotion criteria.
- Diminishability (demotion) criteria: maturity at a given layer is compromised by any failure at its perimeter of information security, i.e., confidentiality, integrity, or availability, on that layer or its preceding layers (principle of easiest penetration).
- If a lower-level security subsystem failure occurs for a given scenario, the maturity score immediately decreases to that level, or lower, according to its new measurement qualification (principle of weakest link). For instance, if the Layer-1 subsystem fails (perhaps because of failure of a non-redundant physical access control) in a system with level-3 maturity, then its

maturity is immediately set to level-1 or lower (based on its new eligibility) until recovery processes complete and appropriate countermeasures are implemented.

- Maturity promotion and demotion are based on the performance measures of choice and underlying maturity function, and defined mathematically accordingly.
- The provided logical order does not necessarily mean a certain level requirement must be met fully before a subsequent level requirement arises.
- Definite security is a continuous process, and therefore, can never be achieved completely.
- Investment of resources on each security control, layer, and the system overall is optimized when balanced with the harm likely to result from relevant failure classes (principle of adequate protection).

3.8 ISMM Mapping: Reliability Block Diagrams, Vectors, and Structures Functions

In order to exploit the relationships between a security system and its individual constituents, i.e., security controls, the security system is first mapped into the context of the ISMM model. This mapping makes a transformation from the physical arrangement of the security system, represented by its set of controls, into the logical arrangement of the system with respect to failure, represented by five sets of controls, corresponding to the five ISMM subsystems. The output of this process leads to building ISMM reliability block diagrams, vectors, and structure functions, adopted from reliability theory. This transformation process represents a necessary step before subsequent mathematical representations and evaluation techniques, demonstrated in Chapter 4 and Chapter 5, can take place.

The representation methods in particular are extended into our work for two main purposes: first, as applied in conventional reliability, these methods are used to show and analyse the effects of failure, and success, of any component in any ISMM structure. They can show how different controls' failures (or successes) combine to cause ISMM-based structures, subsystems, and system failures (or successes). Thereby, they can be used for establishing reliability and availability measures of security systems. Second, they provide compact visual and mathematical representation suitable to reflect the ISMM-based logical hierarchy of a security system, including the arrangement of its controls with respect to the reference of interest, allowing the identification and enumeration of all possible states' pathways. This representation is a multiple-level one that allows top-level RBDs, vectors, and

structure functions to be decomposed into smaller items repeatedly, until they reach the level of abstraction of interest. Using this representation, various quantifier functions of other performance measures, beyond conventional reliability, can be defined, as explained later.

The distinction between physical and logical arrangements of a system is important in deriving the proper mathematical representation used in the analysis. The physical arrangement is meant to reflect the actual arrangement between different security controls or components, whereby the logical arrangement is drawn with respect to the behaviour of the failure process of controls. This distinction results in cases where the logical arrangement of a system is not necessary the exact match to its physical arrangement. Translating the physical arrangement of security controls into logical network, however, requires a solid understanding of the system's physical arrangement, its security functional requirements, and operational behaviour among its controls. For a given set of security controls, $\boldsymbol{C} = \{C_k, for\ all\ k = 1, \dots, n\}$, the actual functionality of each control C_k determines its logical boundary domain, i.e., the corresponding layer on the ISMM model. Further technical details on each control, such as the kind of protection mechanism (prevention, detection, and/or recovery) along with the type of that control C_i, determines its logical arrangement within that layer. Analogical to defining logical arrangement in reliability theory, the security system is modelled as an interconnection of smaller parts, or controls, in series and parallel based on the failure model defined earlier. The following rules are used to define the logical arrangement between a structure of two controls, and consequently to build up the structures of individual subsystems and the overall system along the ISMM hierarchy:

1. If failure of one control leads to the failure of the functions of the combination, the two controls are considered to be operating in a series arrangement.
2. If failure of one control leads to the other control taking over the functions of the failed control, that is, the combination of functions continues, the two controls are considered to be operating in a parallel arrangement.

Implementing this procedure on all controls of the security system, we obtain

$$\begin{aligned} C_k, k = 1, \dots, n &\xrightarrow{maps\ to} C_{i,j} \\ i &= layers\ 1, \dots, 5\ on\ ISMM, \\ j &= controls\ 1, \dots, n_i\ at\ layer\ i, \end{aligned} \tag{3-1}$$

n_i *total number of layer i controls,*

$n = \sum_{i=1}^{5} n_i$ *, total number of sysem controls.*

This step leads to building the ISMM model as depicted in Figure 3-6.

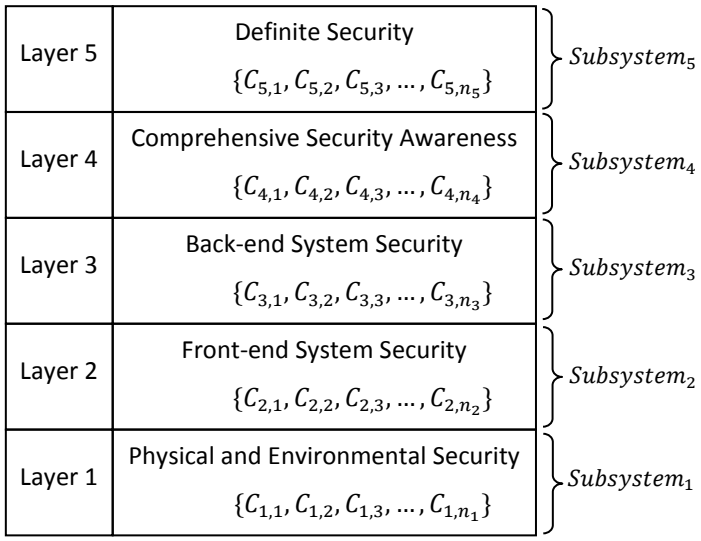

Figure 3-6: Mapping security controls on ISMM Model

Observe that the abstraction of a *structure* is employed to capture the logical interconnectivity between a set of controls at any level. At system-level, the security system is seen as a structure of order five, containing all $subsystem_i's$ structures, each of which is another structure of order $n_i, i = 1, \dots, 5$; n_i denotes the number of security controls at $subsystem_i$. Further, the security controls can themselves be structures of smaller components and so forth until the smallest identifiable component level. This multiple-level decomposition is represented by

$security\ system_{ISMM} = \{\boldsymbol{C_i}\ or\ subsystem_i, for\ all\ i = 1, \dots, 5\}$

$subsystem_i\ or\ layer_i = \{C_{i,j}, for\ all\ i = 1, \dots, 5\ and\ j = 1, \dots, n_i\}$

$C_{i,j} = \{C_{i,j,k}, for\ all\ k = 1, \dots, n_{i,j}\}$,

and so on as moving into smaller levels of abstraction.

To study the relationships between individual controls and subsystems, and their effect on the whole ISMM-based security system, one has to know how performance or failure of smaller components affects performance or failure of larger ones along the hierarchy of all structures of the system. To show this, we use binary state representation, where the component state equals 1 when it is functioning and 0 when it is non-functioning, allowing state vectors and structure functions to be constructed.

We first introduce the so-called indicator, or Boolean, function $x_{i,j}$ for control j at layer i, $i = 1, \dots, 5$ and $j = 1, \dots, n_i$, as used in reliability theory [48], [49], [104], [105], [113], [114], according to the following definition

$$x_{i,j} = \begin{cases} 1, & if\ the\ jth\ control\ at\ ith\ layer\ is\ functioning \\ 0, & otherwise \end{cases} \tag{3-2}$$

The vector $\mathbf{x_i} = (x_{i,1}, x_{i,2}, \dots, x_{i,n_i})$ is called the *state vector* with n_i coordinates for layer i or $subsystem_i$. It shows the status of each security control at layer i as either working or failed. So, $subsystem_i$ has 2^{n_i} different states determined by the status of individual controls at layer i. Some of these 2^{n_i} vectors will place the structure $\mathbf{x_i}$ into a functioning state, and others will make it fail. To denote the states of structures, we further introduce the *structure function* $\emptyset(\mathbf{x_i})$ of the vector $\mathbf{x_i}$ for the structure or $subsystem_i$ as a whole,

$$\emptyset(\mathbf{x_i}) = \begin{cases} 1, & if\ subsystem_i\ is\ functioning \\ 0, & otherwise \end{cases} \tag{3-3}$$

Similar to the features of (2-1) and (2-2), $\emptyset(\mathbf{x_i})$ is a Boolean function of $x_{i,j}$. In addition, $\emptyset(\mathbf{x_i})$ is a monotonic increasing function of the vector $\mathbf{x_i}$, which means if $x_{i,j} \le y_{i,j}, i = 1, \dots, 5\ and\ j = 1, \dots, n_i, then\ \emptyset(\mathbf{x_i}) \le \emptyset(\mathbf{y_i})$.

A *series* system, or structure, will function if and only if all of its controls are functioning, whereas a *parallel* system needs only one of its controls to be functioning. For a pure series structure of $subsystem_i$, the structure function is written as

$$\emptyset(\mathbf{x_i}) = min(x_{i,1}, x_{i,2}, \dots, x_{i,n_i}) = \prod_{j=1}^{n_i} x_{i,j} \tag{3-4}$$

The pure series structure can be represented using the reliability block diagram as shown in Figure 3-7.

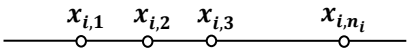

Figure 3-7: Subsystem$_i$ representation of a series structure

On the other hand, for a pure parallel structure for $subsystem_i$, the structure function is written as

$$\emptyset(\mathbf{x_i}) = max\left(x_{i,1}, x_{i,2}, \dots, x_{i,n_i}\right) = 1 - \prod_{j=1}^{n_i}\left(1 - x_{i,j}\right) \tag{3-5}$$

And, the structure can be represented using a reliability block diagram, shown in Figure 3-8.

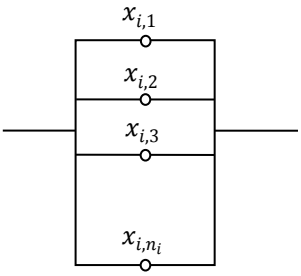

Figure 3-8: Subsystem$_i$ representation of a parallel structure

Observe that both parallel and series structures are special cases of *k*-out-of-*n* structures (or voting systems). Such a structure functions if and only if at least *k* controls are functioning. Since $\sum_{j=1}^{n} x_{i,j}$ equals the number of functioning controls in $subsystem_i$, the structure function of the *k*-out-of-*n* structure is written by

$$\emptyset(\mathbf{x_i}) = \begin{cases} 1, & \sum_{j=1}^{n_i} x_{i,j} \geq k \\ 0, & otherwise \end{cases} = I\left(\sum_{j=1}^{n_i} x_{i,j} \geq k\right) \tag{3-6}$$

However, mixed structures are used to represent systems of mixed security controls as it is unlikely for systems in practice to consist only of either purely series or purely parallel structures. As

explained before, parallel-series structures are solved first for series structures, and then results are plugged into a parallel structure (Figure 3-9). For all n components in each serial line, we first find

$$O_{i,m} = min\left(x_{i,1_m}, x_{i,2_m}, \dots, x_{i,n_m}\right) = \prod_{j=1}^{n} x_{i,j_m} \tag{3-7}$$

Then we calculate higher-order subsystem structure function, $\boldsymbol{S_i}$, as

$$\emptyset(\mathbf{S_i}) = max\left(O_{i,1}, O_{i,2}, \dots, O_{i,m}\right) = 1 - \prod_{j=1}^{m}\left(1 - O_{i,j}\right) \tag{3-8}$$

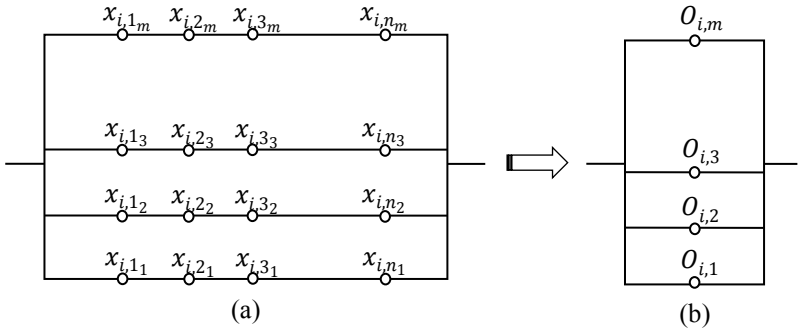

Figure 3-9: Subsystem$_i$ representation of a parallel-series mixed structure

Series-parallel structures are solved first for parallel structures, and then results are plugged into a series structure (Figure 3-10). For all m components in each parallel line we first find

$$O_{i,n} = max\left(x_{i,1_n}, x_{i,2_n}, \dots, x_{i,m_n}\right) = 1 - \prod_{j=1}^{m}(1 - x_{i,j_n}) \tag{3-9}$$

Then we calculate higher-order subsystem structure function, $\boldsymbol{S_i}$, as

$$\emptyset(\mathbf{S_i}) = min\left(O_{i,1}, O_{i,2}, \dots, O_{i,n}\right) = \prod_{j=1}^{n} O_{i,j} \tag{3-10}$$

So, the general rule is that structures are reducible to smaller entities based on the same techniques used in reliability theory in order to make mathematical computations more compact and easier.

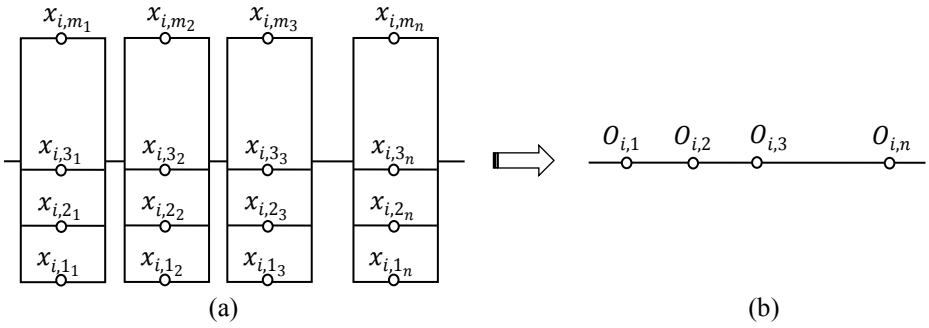

Figure 3-10: Subsystem$_i$ representation of a series-parallel mixed structure

While the above equations are presented for subsystem structures, they can be similarly used to encode smaller levels of structural abstractions. Also, n and m need not to always be equal in each row and column, respectively, in both mixed structures. However, the structure function $\emptyset(\mathbf{x_i})$ for each ISMM subsystem is required to be equal to 1 in order for that subsystem to be functioning.

Additionally, following the logical connectivity rules mentioned earlier, the ISMM-based system-level structure is defined by a series arrangement among its individual subsystems. In order for the security system, $\boldsymbol{S}$, to be functioning, all subsystems $\boldsymbol{S_i}'s, i = 1\ to\ 5$, need to be functioning. This arrangement makes a suitable relationship as security is measured by its weakest link. We call the vector $\mathbf{x} = \big(\emptyset(\mathbf{x_1}), \emptyset(\mathbf{x_2}), \emptyset(\mathbf{x_3}), \emptyset(\mathbf{x_4}), \emptyset(\mathbf{x_5})\big)$ the *state vector* of ISMM-based security system $\boldsymbol{S}$, representing the individual structure functions of the subsystems, showing the status of each $subsystem_i$ as either working or failed. All possible states of the 5-tuple $\mathbf{x}$ are determined by the individual values of its vertices. So, if we let $\emptyset(\mathbf{x})$ represent the structure of the whole security system, then

$$\emptyset(\mathbf{x}) = \begin{cases} 1, & if\ security\ system\ is\ functioning \\ 0, & otherwise \end{cases} \tag{3-11}$$

Moreover, since the structure of the whole system is represented by a pure series arrangement of individual subsystems, as illustrated in Figure 3-11, $\emptyset(\mathbf{x})$ is written as follows:

$$\emptyset(\mathbf{x}) = min\big(\emptyset(\mathbf{x_1}), \emptyset(\mathbf{x_2}), \emptyset(\mathbf{x_3}), \emptyset(\mathbf{x_4}), \emptyset(\mathbf{x_5})\big) = \prod_{i=1}^{5} \emptyset(\mathbf{x_i}) \tag{3-12}$$

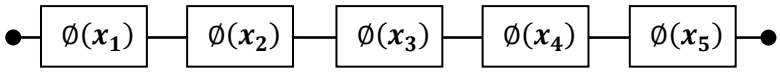

Figure 3-11: ISMM-based security system logical representation

Observe that the use of structure functions is not limited to defining the state of ISMM structures, subsystems, or system, but rather can be the basis for defining quantifier functions of ISMM-based performance measures.

3.9 ISMM Measures

The structure function $\emptyset(.)$ is central to the ISMM approach as it defines the relationships among controls that constitute every subsystem and consequent system overall. All quantifier functions for each performance measure of interest are defined subsequently based on this structure. Recall that structure functions deal with the study of the state of a control, subsystem, and system overall (e.g., functioning or non-functioning in binary systems), whereas quantifier functions deal with performance measures such as reliability and availability. We present specific properties of the structure function that are based on common properties found in traditional reliability engineering [48], [113], [114], [115] and multi-state systems analysis [118], [119], [125], [128], [129]. These properties are conditions necessary to ensure proper definition and application of ISMM quantifier functions, and are summarised into the following:

1. $\emptyset(.)$ is commutative, that is,

$$\emptyset\left(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,j}, \mathrm{x}_{i,j+1}, \dots, \mathrm{x}_{i,n_i}\right) = \emptyset\left(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,j+1}, \mathrm{x}_{i,j}, \dots, \mathrm{x}_{i,n_i}\right)$$

2. $\emptyset(.)$ is associative, that is,

$$\emptyset\left(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,j}, \mathrm{x}_{i,j+1}, \dots, \mathrm{x}_{n_i}\right) = \emptyset\left(\emptyset\left(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,j}\right), \emptyset\left(\mathrm{x}_{i,j+1}, \dots, \mathrm{x}_{n_i}\right)\right)$$

Note that commutativity and associativity are necessary to guarantee that changing the order or grouping of operands does not change the end product. $\emptyset(.)$ is commutative and associative across all abstraction levels.

3. $\emptyset(.)$ is *coherent* [114], which is translated in our work by stating that the security system is coherent if and only if 1) the structure function $\emptyset(.)$ is nondecreasing in each argument $x_{i,j}$ for

$i = 1, \dots, 5$ and $j = 1, \dots, n_i$; and 2) every control is relevant. This property can be further decomposed into the following conditions. At the subsystem's abstraction level,

a. $\emptyset(\mathbf{0_i}) = 0$, meaning $subsystem_i$ is in a failure state when all controls fail

b. $\emptyset(\mathbf{1_i}) = 1$, meaning $subsystem_i$ works when all controls work

c. If $\mathbf{x_i} \leq \mathbf{y_i}$ then $\emptyset(\mathbf{x_i}) \leq \emptyset(\mathbf{y_i})$, meaning $\emptyset(\mathbf{x_i})$ in nondecreasing in each argument $x_{i,j}$

d. There exists a vector $\mathbf{x_i}$ such that $0 = \emptyset(0_{i,j}, \mathbf{x_i}) < \emptyset(1_{i,j}, \mathbf{x_i}) = 1$. This means that for every control $C_{i,j}$ in $subsystem_i$, there exists a control state vector such that the state of the control dictates the state of the subsystem $\boldsymbol{C_i}$,

Note that the same conditions above apply across the hierarchy, e.g., $\boldsymbol{C_i}'s$ are relevant to $\boldsymbol{C}$ and its structure function $\emptyset(.)$ is nondecreasing in each argument x_i.

4. The system controls are logically homogenous, meaning that all controls and the overall system itself have the same number of distinguished states with respect to the measure of interest. This is important to ensure that the same quantifier functions can be applied to all controls.

5. System configuration is based on simple structures, which are structures that can be reduced to any combination of series and parallel models with independent controls.

   The properties (1) to (3) above are based on common properties assumed in most of studies on structure functions in reliability theory and the generalization in MSS and UGF [117], [119], [130]. The assumptions (4) and (5) are necessary to unify the demonstrated models and simplify calculations, and so ensure that the mathematical representation demonstrated here applies directly without the need for any additional transformation steps. Thus, the examples demonstrate binary systems in addition to providing model representation for multi-state systems. Analysis of complex structures, however, such as bridge and delta structures, can be found in [47], [115], [116] using more complicated techniques, which fall outside the contribution of this work. Observe that such techniques used in reliability theory in general can be readily adoptable to this work.

As used in reliability and MSS analysis, a proper quantifier function must unambiguously define the rules for estimating the performance of the measure of interest. The definition of such a function strictly depends on the logical arrangement between controls in the system and the physical nature of the performance measure of interest.

3.10 Maturity Function

The maturity notion is extended in this work at many levels; in particular, 1) it allows the integrated quantification of both qualitative aspects and quantitative measures to be captured on the same model foundation. 2) The precedence introduced is different than the order used in traditional maturity models: in the latter, layers are used to define the order of fulfillment in an abstract way; while in our work it is more of a precedence defined by the effect of existence of controls and impact of their absence or failure. This property pictures the security phenomena into a game-theoretic behavior between defenders and opponents, thus signifying the priority of fulfillment with respect to the two worlds. 3) Structures are introduced to add another dimension of scalability and application, representing logical relationships among the set of items or objects in every layer. 4) The boundedness property applies to controls, subsystems, and the overall system individually and collectively.

The qualification of maturity to a certain ISMM-based security level, as mentioned in the main propositions of the ISMM model, is determined by the *consecutive adequacy* from the lowest layer in the hierarchy of precedence to the layer of maturity. Adequacy means security controls are resilient and effective at the qualified maturity level and all its preceding layers, so that the logical precedence and aggregation of security boundaries is preserved. The subjectivity aspect of the notion *adequate* is modeled quantitatively according to the performance measure(s) of interest, so it can be well identified, evaluated, and measured in a consistent manner.

This setup mathematically means that the performance measures need to be equal or higher than the predefined adequacy lower bounds. Examples of such measures include, but are not limited to, reliability and availability measures. In the case of maturity based on multiple measures, however, various adequacy functions can be applied; for instance, the acceptance test method, where the maturity test is defined based on whether every measure is equal or more than its minimum bound; the weighted average method, where the average of measures is checked against a combined average minimum bound; the voting method, where the number of successful checks is compared to a minimum bound on successful votes; or the geometric mean, where the central tendency among the set of measures is computed, thus making a good measure for finding a single outcome out of several heterogeneous sources, which is commonly applied in various evaluation exercises in the computing field [153].

In either case, maturity is a composite value determined by the results of the performance measures in use, taking the general form

$$\mathcal{M}_{ISMM} = \mathcal{F}(\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_l) \tag{3-13}$$

where each maturity measure $\mathcal{M}_k$ satisfies

$$\begin{aligned} &\mathcal{M}_k = \max(m)\ s.t.\ \mathrm{M}_{i_k} \geq threshold\left(\mathrm{M}_{i_k}\right), for\ all\ layers\ i = 1, \dots, m; \\ &m = 1, \dots, 5; k = 1, \dots, l. \end{aligned} \tag{3-14}$$

Alternatively, using the acceptability function,

$$\mathcal{M}_k = \begin{cases} \max(m), & \left\{\prod_{i_k=1}^{m} I\left(\mathrm{M}_{i_k} \geq threshold\left(\mathrm{M}_{i_k}\right)\right)\right\} = 1 \\ 0, & otherwise \end{cases} \tag{3-15}$$

The following algorithm can be used to find the maturity $\mathcal{M}_k$ for every measure k

1. Input: M_{i_k} for all $i = 1, \dots, 5$
2. Output: $\mathcal{M}_k$
3. Initialization: $\mathcal{M}_k = 0$
4. for m=1 to 5 do
5. if $\prod_{i_k=1}^{m} I\{\mathrm{M}_{i_k} \geq threshold\left(\mathrm{M}_{i_k}\right)\}$
6. then $\mathcal{M}_k \leftarrow m$
7. else return $\mathcal{M}_k$
8. end for
9. return $\mathcal{M}_k$

The condition in the maturity test above is necessary to fulfill the logic of the essential layering principle of the ISMM model, which basically reflects the precedence of logical security boundaries and associated dimensions and propositions. Observe that the merit in mentioning multiple adequacy functions is to indicate that the ISMM approach facilitates different kinds of measures so that the most suitable yet consistent one is applied, especially in the case of multiple performance measures, as they are meant to reflect different performance capabilities of the system.

The essence of this work, however, is based on the reliability measure due to its realised significance in reliability engineering for decades as a key performance measure for evaluating systems' operational capability and its useful applications in the security field. Therefore, to demonstrate maturity quantification and its application in security engineering, we approach the reliability measure through common methods and techniques found in reliability engineering and MSS evaluation using the UGF method. We restrict ourselves to the reliability measure in the definition of maturity function.

The use of the reliability measure means, regardless of the approach used, the computed reliability of security controls at every qualified layer, i.e., subsystem's reliability R_i, is required to be equal or higher than the predefined adequacy or lower bound r_i to meet the maturity condition for that layer. In mathematical form,

$$\mathcal{M}_{ISMM} = \mathcal{M}_R = \max(m)\, s.t.\, R_i \geq threshold(R_i) = r_i \quad for\ all\ layers\ i = 1, \dots, m; m = 1, \dots, 5. \tag{3-16}$$

We believe maturity for a given system is a relative aspect that defines the relationship between a particular capability of the system and its respective demand (i.e., the intended performance goal). For a reliability measure as an example, we define maturity as the relative distance between the actual reliability value and the expected reliability demand of the system, representing the intended goal we are willing to accept (i.e., minimum acceptable bound of reliability). This view applies analogically to other performance measures.

3.11 Case Study

3.11.1 Evaluation Process

To demonstrate the ISMM modeling approach, including the application of reliability engineering presented in Chapter 4 and the Multi-state System (MSS) representation and Universal Generating Function (UGF) presented in Chapter 5, we set up a case study that represents a common scenario of security systems protecting enterprise-level computing environments. The selected controls in this example, however, are mostly compiled from best practices and recommendations for security controls, such as those published by the SANS institute [162] and NIST SP 800-53 publication [163].

The evaluation process follows the general procedure used in reliability engineering, such as the one described in [47], besides the addition to accommodate the ISMM approach. It can be broken into three main steps:

1. Construction phase. Three main tasks are performed here: i) the problem of interest is described including the failure profile. ii) The ISMM-based system model is built, mapping a system's physical arrangement into its logical arrangement, including state vectors, reliability block diagrams, and structure functions. iii) The corresponding maturity bounds are established. Task (i) is presented in Section 3.11.2, and (ii) and (iii) are presented in Section 3.11.3.
2. Processing or analysis phase. The values of the selected performance measures are calculated, which might involve some solution tools or approximations, particularly for complicated scenarios. This step is an approach-specific one. Thus, in Chapter 4 the following combinatorial methods are presented: minimal path set, minimal cut set, modeling based on random events, modeling based on time-independent random variables, and modeling based on time-dependent random variables. In Chapter 5, MSS representation and the corresponding UGF method are presented.
3. Evaluation and interpretation phase. The results of the analysis phase are validated using statistical techniques and comparisons with previous states of the system or other similar systems, allowing possible improvements of the system to be concluded. The analysis, based on the reliability approach and MSS UGF method, is presented in Chapter 4 and Chapter 5, respectively.

3.11.2 System Description

Consider a computing environment that is based on a web-based application model, consisting of multiple web servers, applications, and database platforms secured in one premise. The protection of this computing environment requires the implementation of a security system that consists of multiple types of security controls, including physical, technical, and administrative controls, totalling thirty-three controls. The physical layout of this security system is depicted in Figure 3-12.

For computer security of application-level access, a redundant control is implemented for application software, a dedicated control for protecting access from wireless devices, and 2-out-of-3 redundant controls to protect web servers and meet their demand threshold before they can be considered in a failure state. For computer security of underlying networking infrastructure (i.e., back-end resources), two external firewalls and one internal firewall are used to set up the DMZ zone

over the web servers. A redundant database-level security control is implemented over the database platforms. Two administrative workstations are dedicated to monitoring and controlling access privileges of the system resources working in active redundancy mode. In addition, the underlying network traffic is protected by a control for malware defences and Intrusion Prevention/Detection Systems (IPS/IDS) for the protection of inbound and outbound network traffic. For physical security of the overall system, assume redundant controls in the form of surveillance systems (i.e., CCTVs), redundant locks, redundant mantraps, and redundant fences to protect the physical boundary of the entire premises. For security of the human-factor, the organization implements a consolidated social engineering awareness program, continuous training for users, and a dedicated personnel security control (i.e., screening, termination, third-party access agreement, etc.), in addition to the governance of organization-wide security policies and procedures. For other organization-wide governance and assurance controls, the organization performs vulnerability assessment and penetration testing across all controls of all types, in addition to sustaining risk assessment controls, compliance procedures, incident response and management controls, and accounting and auditing controls.

Assume independence among all the controls and the active redundancy mode for redundant controls. The complete set of controls classified according to their type is outlined in Table 3-2.

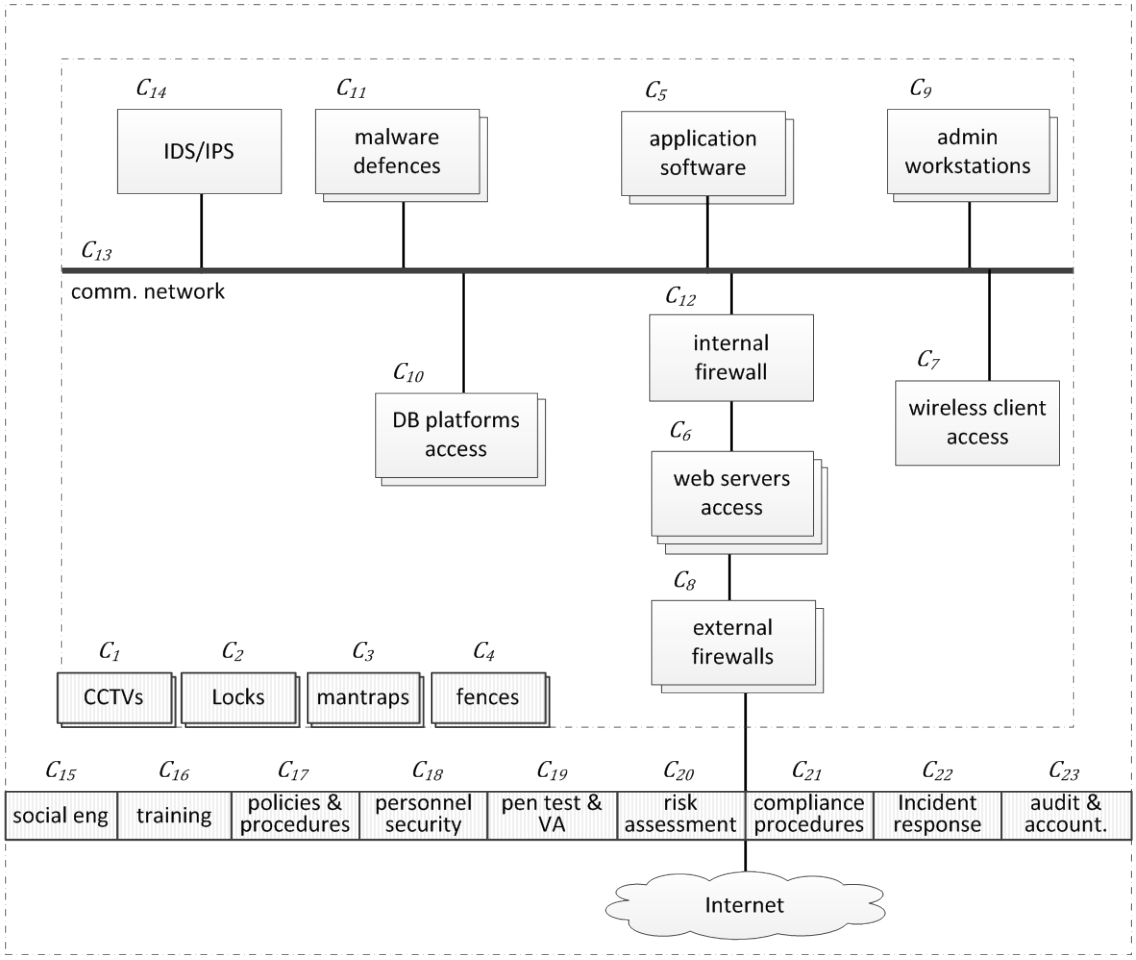

Figure 3-12: Physical layout of the ISMM case study[5]

Table 3-2: Security controls of the ISMM case study

Control No.	Name	Type	Redundancy	ISMM Security boundary
C_1	CCTVs	Physical	double	Physical and environmental
C_2	Locks	Physical	double	Physical and environmental

[5] Note that the physical layout reflects technical controls inside the inner dotted box, physical controls surrounding the technical controls, and administrative controls surrounding the whole computing environment in the outer dotted box. There is no particular arrangement for both types of physical and administrative controls in the physical sense.

C_3	Mantraps	Physical	double	Physical and environmental
C_4	Fences	Physical	double	Physical and environmental
C_5	Application software	Technical	Double	Front-end
C_6	Web servers	Technical	2-out-of-3	Front-end
C_7	Wireless client access	Technical	None	Front-end
C_8	External firewalls	Technical	Double	Back-end
C_9	Administrative workstations	Technical	Double	Back-end
C_{10}	Database platforms access	Technical	Double	Back-end
C_{11}	Malware defences	Technical	Double	Back-end
C_{12}	Internal firewall	Technical	None	Back-end
C_{13}	Network	Technical	None	Back-end
C_{14}	IPS/IDS	Technical	None	Back-end
C_{15}	Social engineering	Administrative	None	Comprehensive awareness
C_{16}	Training	Administrative	None	Comprehensive awareness
C_{17}	Policies & procedures	Administrative	None	Comprehensive awareness
C_{18}	Personnel security	Administrative	None	Comprehensive awareness
C_{19}	Penetration testing & vulnerability assessment (VA)	Administrative	None	Definite security
C_{20}	Risk assessment	Administrative	None	Definite security
C_{21}	Compliance procedures	Administrative	None	Definite security
C_{22}	Incident response & management	Administrative	None	Definite security
C_{23}	Auditing & accountability	Administrative	None	Definite security

3.11.3 ISMM Mapping: RBDs, Vectors, and Structure Functions

Before proceeding to any mathematical modeling, we need to establish some definitions, assumptions, and bounds as common grounds for subsequent reliability and maturity analysis. First,

the failure definition is central to building reliability function regardless of the application. Recall that complete failure is not a necessary condition for considering a control in failure state; rather, not performing the intended function is what determines the failure state. Similarly, this applies to the context of security controls using impact of failure statistics explained in Section 3.5. For instance, the probability of failure for CCTVs control C_1 can be defined by

$$P(failure\ of\ C_1)$$

$$= P(\text{"}\textit{inability of CCTVs to perform the intended surveillance function}\text{"})$$

And as an example of technical controls, say web servers C_6

$$P(failure\ of\ C_6)$$

$$= P(\text{"}\textit{failure of web servers access controls}\text{"})$$

And as an example of administrative control, say incident response and management C_{21}

$$P(failure\ of\ C_{21})$$

$$P(\text{"}\textit{failure of IR procedures to control attacks within the scope of the organization threats and vulnerabilities}\text{"})$$

A common method, however, that is widely used in reliability statistics to find the mean time to failure $MTTF$ of an item is the sample mean (empirical mean), which can be calculated as

$\overline{\text{MTTF}} = (t_1 + t_2 + \cdots + t_n)/n$, where $t_1, t_2, \ldots, t_n$ are observed failure-free operating times of statistically identical items. Observe that for constant failure rate λ, $MTTF = 1/\lambda$.

Traditionally, when the failure model is being built, failure datasets are estimated in different ways: i) statistical data about historical failures of components, ii) government and commercial data, iii) field datasets, iv) experimental data, or v) expert knowledge and reasonable assumptions. In this work, for the reasons explained in Section 2.1.5, we consider expert knowledge compiled from best practices and reasonable assumptions in outlining controls and building the case study.

Second, the specification of a time interval for this scenario is assumed to be one week. Thus, the probability functions and physical interpretations are based on this period. Third, the environmental conditions of the system under consideration such as humidity and temperatures are not specified as we do not have failure datasets of individual controls, which are commonly available in traditional reliability analysis of electronic systems. Fourth, the ISMM-specific bounds are established to reflect

the performance measure of interest: reliability. Observe that these bounds can be design parameters of the system under study, and hence may differ accordingly. They can be used in a way similar to that of nines in engineering; for instance, the term "three nines" means reliability figure of 0.999 and "five nines" means 0.99999. The use of such terms, however, has progressed to establish commonly-accepted benchmark figures for various systems. In this case study, we assume that the individual subsystems have the minimum reliability bounds outlined in Table 3-3.

Table 3-3: ISMM-based reliability minimum bounds

Performance measure at $subsystem_i$	**Minimum bound** r_i
R_1	$r_1 = 0.97$
R_2	$r_2 = 0.93$
R_3	$r_3 = 0.90$
R_4	$r_4 = 0.86$
R_5	$r_5 = 0.80$

The information presented above, i.e., the case study description and associated conditions and bounds, can be compiled during security assessment tasks of the system under study, perhaps as part of a risk assessment exercise. This information, however, facilitates the mapping of the problem from its physical arrangement into the ISMM-based logical arrangement. Such mapping facilitates defining state vectors, reliability block diagrams, and structure functions, as follows.

A total of eight controls, represented by four different classes of controls exhibiting double active redundancy in a binary system, are mapped to Layer-1, which is the physical and environmental security boundary. As such, Subsystem_1 can have up to 2^8 different permutations of control-level states.

The corresponding reliability block diagram is depicted in Figure 3-13. Thus, Subsystem_1 is represented directly by the state vector $\mathbf{x_1} = (x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5}, x_{1,6}, x_{1,7}, x_{1,8})$, and the structure function is written as

$$\emptyset(\mathbf{x_1}) = \max(x_{1,1}, x_{1,2}) \max(x_{1,3}, x_{1,4}) \max(x_{1,5}, x_{1,6}) \max(x_{1,7}, x_{1,8})$$
$$= [1 - (1 - x_{1,1})(1 - x_{1,2})][1 - (1 - x_{1,3})(1 - x_{1,4})][1 - (1 - x_{1,5})(1 - x_{1,6})][1 - (1 - x_{1,7})(1 - x_{1,8})]$$
$$= (x_{1,1} + x_{1,2} - x_{1,1}x_{1,2})(x_{1,3} + x_{1,4} - x_{1,3}x_{1,4})(x_{1,5} + x_{1,6} - x_{1,5}x_{1,6})(x_{1,7} + x_{1,8} - x_{1,7}x_{1,8})$$

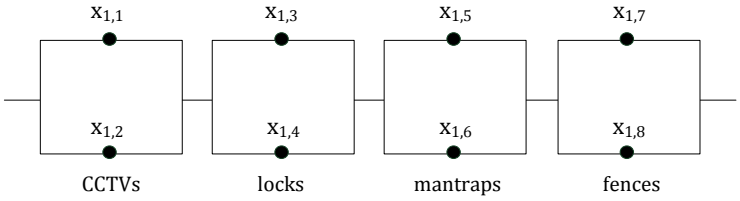

Figure 3-13: RBD for $subsystem_1$

Similarly, the reliability block diagram for Subsystem_2 is established in Figure 3-14 and represented by the vector $\mathbf{x_2} = (x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}, x_{2,6})$. The corresponding structure function is written as

$$\emptyset(\mathbf{x_2}) = \max(x_{2,1}, x_{2,2}) \max(x_{2,3}x_{2,4}, x_{2,3}x_{2,5}, x_{2,4}x_{2,5})\, x_{2,6}$$
$$= [1 - (1 - x_{2,1})(1 - x_{2,2})][1 - (1 - x_{2,3}x_{2,4})(1 - x_{2,3}x_{2,5})(1 - x_{2,4}x_{2,5})]x_{2,6}$$
$$= (x_{2,1} + x_{2,2} - x_{2,1}x_{2,2})(x_{2,3}x_{2,4} + x_{2,3}x_{2,5} + x_{2,4}x_{2,5} - 2x_{2,3}x_{2,4}x_{2,5})\, x_{2,6}$$

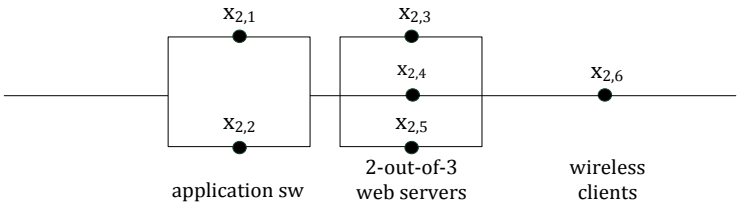

Figure 3-14: RBD for $subsystem_2$

The reliability block diagram for Subsystem_3 is depicted in Figure 3-15, and represented by the vector $\mathbf{x_3} = (x_{3,1}, x_{3,2}, x_{3,3}, x_{3,4}, x_{3,5}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10})$. Its structure function is written as

$$\emptyset(\mathbf{x_3}) = \max(x_{3,1}, x_{3,2}) \max(x_{3,3}, x_{3,4}) \max(x_{3,5}, x_{3,6})\, x_{3,7} x_{3,8}, x_{3,9} x_{3,10}$$

$$= [1 - (1 - x_{3,1})(1 - x_{3,2})][1 - (1 - x_{3,3})(1 - x_{3,4})][1 - (1 - x_{3,5})(1 - x_{3,6})]\, x_{3,7}\, x_{3,8} x_{3,9} x_{3,10}$$

$$= (x_{3,1} + x_{3,2} - x_{3,1}x_{3,2})(x_{3,3} + x_{3,4} - x_{3,3}x_{3,4})(x_{3,5} + x_{3,6} - x_{3,5}x_{3,6}) x_{3,7} x_{3,8} x_{3,9} x_{3,10}$$

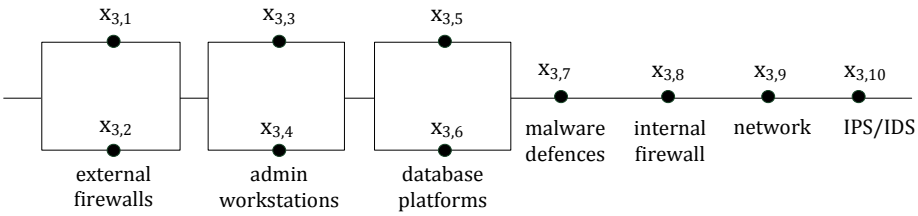

Figure 3-15: RBD for $subsystem_3$

The reliability block diagram for Subsystem_4 is shown in Figure 3-16, and is represented by the vector $\mathbf{x_4} = (x_{4,1}, x_{4,2})$, where its structure function is written as

$$\emptyset(\mathbf{x_4}) = \min(x_{4,1}, x_{4,2}, x_{4,3}, x_{4,4}) = x_{4,1} x_{4,2} x_{4,3} x_{4,4}$$

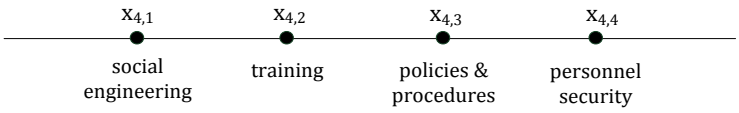

Figure 3-16: RBD for $subsystem_4$

And finally the reliability block diagram for Subsystem_5 is shown in Figure 3-17, and is represented by the vector $\mathbf{x_5} = (x_{5,1}, x_{5,2}, x_{5,3}, x_{5,4}, x_{5,5})$. The corresponding structure function is written as

$$\emptyset(\mathbf{x_5}) = x_{5,1} x_{5,2} x_{5,3} x_{5,4} x_{5,5}$$

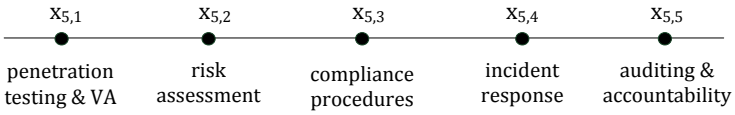

Figure 3-17: RBD for $subsystem_5$

Now, to reflect these functions at the ISMM-based system level, recall the vector $\mathbf{x} = \big(\emptyset(\mathbf{x_1}), \emptyset(\mathbf{x_2}), \emptyset(\mathbf{x_3}), \emptyset(\mathbf{x_4}), \emptyset(\mathbf{x_5})\big)$, which represents the *state vector* of the ISMM model for the whole security system and is formulated by

$$\emptyset(\mathbf{x}) = min\big(\emptyset(\mathbf{x_1}), \emptyset(\mathbf{x_2}), \emptyset(\mathbf{x_3}), \emptyset(\mathbf{x_4}), \emptyset(\mathbf{x_5})\big)$$

$$= \prod_{i=1}^{5} \emptyset(\mathbf{x_i}) = \big(x_{1,1} + x_{1,2} - x_{1,1}x_{1,2}\big)\big(x_{1,3} + x_{1,4} - x_{1,3}x_{1,4}\big)\big(x_{1,5} + x_{1,6} - x_{1,5}x_{1,6}\big)\big(x_{1,7} + x_{1,8} - x_{1,7}x_{1,8}\big)\big(x_{2,1} + x_{2,2} - x_{2,1}x_{2,2}\big)\big(x_{2,3}x_{2,4} + x_{2,3}x_{2,5} + x_{2,4}x_{2,5} - 2x_{2,3}x_{2,4}x_{2,5}\big)x_{2,6}\big(x_{3,1} + x_{3,2} - x_{3,1}x_{3,2}\big)\big(x_{3,3} + x_{3,4} - x_{3,3}x_{3,4}\big)\big(x_{3,5} + x_{3,6} - x_{3,5}x_{3,6}\big)x_{3,7}x_{3,8}x_{3,9}x_{3,10}x_{4,1}x_{4,2}x_{4,3}x_{4,4}x_{5,1}x_{5,2}x_{5,3}x_{5,4}x_{5,5}$$

Mapping all subsystems onto the ISMM model leads to building the overall system structure as depicted in Figure 3-18. Observe that the system structure can have up to 2^5 different permutations of subsystem-level states, consisting of another lower-level abstraction of 2^{33} different permutations of control-level states.

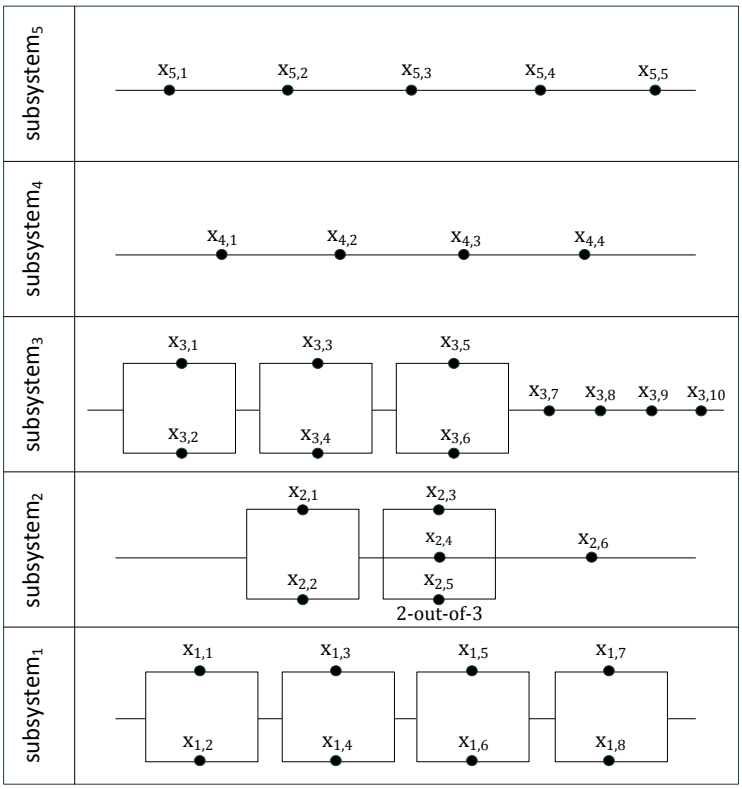

Figure 3-18: ISMM-based system-level RBD of the case study

3.12 Summary

In this chapter, we have introduced the abstraction of a computing system into assets and controls, a conceptual perspective that can be employed for various studies involving the interaction between these two classes. The abstraction is useful in examining the security element from both economical and operational perspectives. Second, we have introduced a failure model based on the impact of failures consequent to malicious and/or nonmalicious causes. We have also discussed the importance of addressing the impact of failure in defining failure models for security studies, which can lead to building a plausible statistical domain of the security failure problem. Third, we have presented the revised version of the ISMM model, explaining its architecture, propositions, and mapping process into RBDs, vectors, and structure functions. We have then presented common properties of model measures and how to build the maturity function quantitatively. The model allows us to identify and capture conceivable structural relationships of security controls, and so build the logical network of the security system.

Chapter 4
Reliability-theoretic ISMM-based Analysis

4.1 Introduction

Over the past decade or so, new research has extended concepts and models from dependability theory into security studies [7], [8], [9], [20], [20], [21], [27], [73], [86], [98], [99], [100]. Following the representation approach of the security system presented in the previous chapter, we present herein several reliability-theoretic analysis methods to study dependability and maturity of security systems. In particular, we extend the application of minimal path method, minimal cut method, and reliability analysis, and then show how to establish the maturity function accordingly.

The adoption of such methods is mainly based on earlier reliability studies, such as those in [48], [49], [80], [104], [105], [113], [114], [120]. We use SHARPE tool [68] to build and compute the resulting numerical values. More complicated techniques, however, such as reliability heuristics and approximation methods [47], [104], [113], [115], [120], and simulation (such as those explained in [34], [36]) can be used for complex scenarios. Such methods should be applied naturally, observing that the purpose is not to advance reliability analysis methods themselves, but rather to advance security analysis by adopting such methods.

The remainder of this chapter is organized as follows. Section 4.2 analyzes the security system using minimal path and cut methods, showing which sets lead to the functioning and failure states of the security system. For completeness, reliability modeling starts in Section 4.3 based on basic random events of controls failures. Section 4.4 then presents reliability analysis based on random variables, considering cases of both static probabilities of failure and dynamic, i.e., time-dependent, probabilities of failure along with maturity analysis. In each method, the model formulations and demonstrative examples use the case study in Section 3.11.

4.2 Minimal Path and Minimal Cut Sets

4.2.1 Model Formulation

In a similar manner to their use in traditional reliability analysis, minimal path and minimal cut sets are adopted in ISMM-based reliability analysis. We define a minimal path set for $subsystem_i$ as the minimum set of security controls required to ensure the functioning state of $subsystem_i$.

Consequently, a minimal cut set for $subsystem_i$ is the minimum set of security controls whose failure leads directly to the failure of the whole $subsystem_i$. Minimal path sets and minimal cut sets allow us to represent a security system as a parallel-series structure and a series-parallel structure, respectively. In this part, we demonstrate the extended mathematical representation at the subsystem level first, noting that it is applicable to encoding smaller levels of structural abstraction. Then we demonstrate the representation at the ISMM-system level.

For a state vector $\mathbf{x_i}$ to be called a *path vector* for layer $i, i = 1,\ldots,5$, it is required to have $\emptyset(\boldsymbol{x_i}) = 1$. It is also said to be a *minimal path vector* if $\emptyset(\boldsymbol{y_i}) = 0$ for all $\boldsymbol{y_i} < \boldsymbol{x_i}$ $\left(y_{i,k} \le x_{i,k}, k = 1, \ldots n_i \text{ and } y_{i,k} < x_{i,k} \; for \; some \; k\right)$. The set $\boldsymbol{A_{i,k}} = \{k: x_{i,k} = 1\}$ is also called a *minimal path set*. Now, for a given $subsystem_i$, let $\boldsymbol{A_{i,1}, A_{i,2}, \ldots, A_{i,s}}$ denote the minimal path sets. We define a new Boolean function $\propto_{i,j} (\boldsymbol{x_i})$ of the *j*th minimal path set for $subsystem_i$ as follows,

$$\begin{aligned} \propto_{i,j} (\boldsymbol{x_i}) &= \begin{cases} 1, & if \; all \; controls \; of \; \boldsymbol{A_{i,j}} \; set \; are \; functioning \\ 0, & otherwise \end{cases} \\ &= \prod_{k \in \boldsymbol{A}_{i,j}} x_{i,k} \end{aligned} \tag{4-1}$$

Thus, to ensure that $subsystem_i$ is functioning, we need at least one of its minimal path sets where all the controls are functioning. This leads to conclude the following relationship.

$$\begin{aligned} \emptyset(\boldsymbol{x_i}) &= \begin{cases} 1, & if \;\; \propto_{i,j} (\boldsymbol{x_i}) = 1 \; for \; some \; j \\ 0, & if \;\; \propto_{i.j} (\boldsymbol{x_i}) = 0 \; for \; all \; j \end{cases} \\ &= max_{i,j} \propto_{i,j} (\mathbf{x_i}) \\ &= max_{i,j} \prod_{k \in \boldsymbol{A}_{i,j}} x_{i,k} \end{aligned} \tag{4-2}$$

Furthermore, a state vector $\mathbf{x_i}$ is called a *cut vector* for layer $i, i = 1,\ldots,5$, if $\emptyset(\boldsymbol{x_i}) = 0$ and a *minimal cut vector* if $\emptyset(\boldsymbol{y_i}) = 1$ for all $\boldsymbol{y_i} > \boldsymbol{x_i}$. The set $\boldsymbol{C_{i,k}} = \{k: x_{i,k} = 0\}$ is then called a *minimal cut set*. Now, for a given $subsystem_i$ let $\boldsymbol{C_{i,1}, C_{i,2}, \ldots, C_{i,q}}$ denote the minimal cut sets. We introduce a new Boolean function $\beta_{i,j}(\mathbf{x_i})$ of the *j*th minimal cut set for $subsystem_i$ as follows,

$$\begin{aligned} \beta_{i,j}(\boldsymbol{x_i}) &= \begin{cases} 1, & if \; at \; least \; one \; control \; of \; \boldsymbol{C_{i,j}} \; set \; is \; functioning \\ 0, & otherwise \end{cases} \\ &= max_{k \in \boldsymbol{C}_{i,j}} x_{i,k} \end{aligned} \tag{4-3}$$

As a result, to ensure that $subsystem_i$ is functioning using cut sets, we need at least one control from each of its minimal cut sets to be in a functioning state. In other words, $subsystem_i$ is not functioning if and only if all the controls of at least one minimal cut set are not functioning. This leads us to the following relationship.

$$\begin{aligned} \emptyset(\boldsymbol{x_i}) &= \prod_{j=1}^{q} \beta_{i,j}(\boldsymbol{x_i}) \\ &= \prod_{j=1}^{q} max_{k \in \boldsymbol{C}_{i,j}} x_{i,k} \end{aligned} \tag{4-4}$$

Applying the above definitions at ISMM-system level, we conclude that the set of individual layer subsystems together represents the only existing minimal path set, whereas each individual subsystem represents a minimal cut set, consisting of one element. This conclusion is because of the pure series arrangement assumption among these subsystems (subsystems 1 to 5) to ensure the overall security system in a functioning state. In other words, the security system is not functioning in the case of failure of any of its subsystems. The minimal path set is represented by

$$\boldsymbol{A_1} = \{\emptyset(\mathbf{x_1}), \emptyset(\mathbf{x_2}), \emptyset(\mathbf{x_3}), \emptyset(\mathbf{x_4}), \emptyset(\mathbf{x_5})\} \tag{4-5}$$

And the five single-element minimal cut sets are

$$\boldsymbol{C_1} = \{\emptyset(\boldsymbol{x_1})\}, \boldsymbol{C_2} = \{\emptyset(\boldsymbol{x_2})\}, \boldsymbol{C_3} = \{\emptyset(\boldsymbol{x_3})\}, \boldsymbol{C_4} = \{\emptyset(\boldsymbol{x_4})\}, \boldsymbol{C_5} = \{\emptyset(\boldsymbol{x_5})\} \tag{4-6}$$

As a result, $\emptyset(\mathbf{x})$ for system-level in terms of minimal path set representation is given by

$$\begin{aligned} \emptyset(\boldsymbol{x}) &= \prod_{i=1}^{5} \left(max_{i,j} \propto_{i,j} (\boldsymbol{x_i}) \right) \\ &= \prod_{i=1}^{5} \left(max_{i,j} \prod_{k \in \boldsymbol{A}_{i,j}} x_{i,k} \right) \end{aligned} \tag{4-7}$$

or

$$\begin{aligned} \emptyset(\boldsymbol{x}) &= max\{min(\boldsymbol{A_1})\} \\ &= min\{\emptyset(\boldsymbol{x_1}), \emptyset(\boldsymbol{x_2}), \emptyset(\boldsymbol{x_3}), \emptyset(\boldsymbol{x_4}), \emptyset(\boldsymbol{x_5})\} \end{aligned} \tag{4-8}$$

Also, $\emptyset(\mathbf{x})$ in terms of minimal cut set representation is given by

$$\begin{aligned}\emptyset(\boldsymbol{x}) &= \prod_{i=1}^{5}\left(\prod_{j=1}^{q}\beta_{i,j}(\boldsymbol{x_i})\right) \\ &= \prod_{i=1}^{5}\left(\prod_{j=1}^{q} max_{k\in \boldsymbol{C}_{i,j}}\, x_{i,k}\right)\end{aligned} \tag{4-9}$$

This also leads to

$$\begin{aligned}\emptyset(\boldsymbol{x}) &= min\{max(\boldsymbol{C_1}), max(\boldsymbol{C_2}), max(\boldsymbol{C_3}), max(\boldsymbol{C_4}), max(\boldsymbol{C_5}) \\ &= min\{\emptyset(\boldsymbol{x_1}), \emptyset(\boldsymbol{x_2}), \emptyset(\boldsymbol{x_3}), \emptyset(\boldsymbol{x_4}), \emptyset(\boldsymbol{x_5})\}\end{aligned} \tag{4-10}$$

Minimal path and cut sets are useful tools for analyzing the critical security controls in the overall system structure where sensitivity analysis can be performed more efficiently. Minimal path sets and minimal cut sets representations can provide information about the lower bounds of controls required to ensure ISMM structures' functionality state and failure state, respectively. Both minimal sets can also be useful in constructing the system structure backward, as they provide an easy way to describe the behaviour of the system. Therefore, the adoption of such sets is deemed beneficial to studying security systems in that they help in identifying the weakest link in the chain more efficiently.

4.2.2 Example

Recall that a minimal path set for $\text{subsystem}_\text{i}$ is defined as the minimum set of security controls required to ensure the functionality state of $\text{subsystem}_\text{i}$. A minimal cut set for $\text{subsystem}_\text{i}$ is the minimum set of security controls whose failure directly causes the failure of the whole $\text{subsystem}_\text{i}$. These sets are applied to individual ISMM subsystems and the system as a whole. We use the case study presented in Section 3.11 to demonstrate this method. We first present a subsystem-level minimal path and cut sets and associated structure functions, followed by the consequent system-level representation.

Subsystem_1 has sixteen minimal path sets, namely,

$$\boldsymbol{A_{1,1}} = \{x_{1,1}, x_{1,3}, x_{1,5}, x_{1,7}\}, \boldsymbol{A_{1,2}} = \{x_{1,1}, x_{1,3}, x_{1,5}, x_{1,8}\}, \boldsymbol{A_{1,3}} = \{x_{1,1}, x_{1,3}, x_{1,6}, x_{1,7}\},$$
$$\boldsymbol{A_{1,4}} = \{x_{1,1}, x_{1,3}, x_{1,6}, x_{1,8}\}, \boldsymbol{A_{1,5}} = \{x_{1,1}, x_{1,4}, x_{1,5}, x_{1,7}\}, \boldsymbol{A_{1,6}} = \{x_{1,1}, x_{1,4}, x_{1,5}, x_{1,8}\},$$
$$\boldsymbol{A_{1,7}} = \{x_{1,1}, x_{1,4}, x_{1,6}, x_{1,7}\}, \boldsymbol{A_{1,8}} = \{x_{1,1}, x_{1,4}, x_{1,6}, x_{1,8}\}, \boldsymbol{A_{1,9}} = \{x_{1,2}, x_{1,3}, x_{1,5}, x_{1,7}\},$$
$$\boldsymbol{A_{1,10}} = \{x_{1,2}, x_{1,3}, x_{1,5}, x_{1,8}\}, \boldsymbol{A_{1,11}} = \{x_{1,2}, x_{1,3}, x_{1,6}, x_{1,7}\}, \boldsymbol{A_{1,12}} = \{x_{1,2}, x_{1,3}, x_{1,6}, x_{1,8}\},$$
$$\boldsymbol{A_{1,13}} = \{x_{1,2}, x_{1,4}, x_{1,5}, x_{1,7}\}, \boldsymbol{A_{1,14}} = \{x_{1,2}, x_{1,4}, x_{1,5}, x_{1,8}\}, \boldsymbol{A_{1,15}} = \{x_{1,2}, x_{1,4}, x_{1,6}, x_{1,7}\},$$

$$\boldsymbol{A_{1,16}} = \{x_{1,2}, x_{1,4}, x_{1,6}, x_{1,7}\}.$$

These are combined with four minimal cut sets, namely,

$$\boldsymbol{C_{1,1}} = \{x_{1,1}, x_{1,2}\}, \boldsymbol{C_{1,2}} = \{x_{1,3}, x_{1,4}\}, \boldsymbol{C_{1,3}} = \{x_{1,5}, x_{1,6}\}, \boldsymbol{C_{1,4}} = \{x_{1,7}, x_{1,8}\}.$$

Recall that the above representation of minimal path sets and minimal cut sets allows one to represent individual subsystems as a parallel-series structure and a series-parallel structure, respectively. As such, subsystem_1, as a parallel-series structure of its minimal path sets, is

$$\emptyset(\boldsymbol{x_1}) = max_{1,j} \propto_{1,j} (\boldsymbol{x_1}) = max_{1,j} \prod_{k \in \boldsymbol{A}_{1,j}} x_{1,k}$$

$$= max\{min(x_{1,1}, x_{1,3}, x_{1,5}, x_{1,7}), min(x_{1,1}, x_{1,3}, x_{1,5}, x_{1,8}), min(x_{1,1}, x_{1,3}, x_{1,6}, x_{1,7}),$$
$$min(x_{1,1}, x_{1,3}, x_{1,6}, x_{1,8}), min(x_{1,1}, x_{1,4}, x_{1,5}, x_{1,7}), min(x_{1,1}, x_{1,4}, x_{1,5}, x_{1,8}),$$
$$min(x_{1,1}, x_{1,4}, x_{1,6}, x_{1,7}), min(x_{1,1}, x_{1,4}, x_{1,6}, x_{1,8}), min(x_{1,2}, x_{1,3}, x_{1,5}, x_{1,7}),$$
$$min(x_{1,2}, x_{1,3}, x_{1,5}, x_{1,8}), min(x_{1,2}, x_{1,3}, x_{1,6}, x_{1,7}), min(x_{1,2}, x_{1,3}, x_{1,6}, x_{1,8}),$$
$$min(x_{1,2}, x_{1,4}, x_{1,5}, x_{1,7}), min(x_{1,2}, x_{1,4}, x_{1,5}, x_{1,8}), min(x_{1,2}, x_{1,4}, x_{1,6}, x_{1,7}),$$
$$min(x_{1,2}, x_{1,4}, x_{1,6}, x_{1,7}) \}$$

and when represented as a series-parallel structure of its minimal cut sets leads to

$$\emptyset(\boldsymbol{x_1}) = \prod_{j=1}^{q} \beta_{1,j}(\boldsymbol{x_1}) = \prod_{j=1}^{q} max_{k \in \boldsymbol{C}_{1,j}} x_{1,k}$$
$$= min\{max(x_{1,1}, x_{1,2}), max(x_{1,3}, x_{1,4}), max(x_{1,5}, x_{1,6}), max(x_{1,7}, x_{1,8})\}$$

This technique applies similarly to the remaining ISMM subsystems. Doing so, subsystem_2 has six minimal path sets. These are

$$\boldsymbol{A_{2,1}} = \{x_{2,1}, x_{2,3}, x_{2,4}, x_{2,6}\}, \boldsymbol{A_{2,2}} = \{x_{2,1}, x_{2,3}, x_{2,5}, x_{2,6}\}, \boldsymbol{A_{2,3}} = \{x_{2,1}, x_{2,4}, x_{2,5}, x_{2,6}\}$$
$$\boldsymbol{A_{2,4}} = \{x_{2,2}, x_{2,3}, x_{2,4}, x_{2,6}\}, \boldsymbol{A_{2,5}} = \{x_{2,2}, x_{2,3}, x_{2,5}, x_{2,6}\}, \boldsymbol{A_{2,6}} = \{x_{2,2}, x_{2,4}, x_{2,5}, x_{2,6}\}.$$

and three minimal cut sets,

$$\boldsymbol{C_{2,1}} = \{x_{2,1}, x_{2,2}\}, \boldsymbol{C_{2,2}} = \{x_{2,3}x_{2,4}, x_{2,3}x_{2,5}, x_{2,4}x_{2,5}\}, \boldsymbol{C_{2,3}} = \{x_{2,6}\}.$$

These results lead to defining $\emptyset(\boldsymbol{x_2})$ in terms of minimal sets,

$$\emptyset(\boldsymbol{x_2}) = max\{min(x_{2,1}, x_{2,3}, x_{2,4}, x_{2,6}), min(x_{2,1}, x_{2,3}, x_{2,5}, x_{2,6}), min(x_{2,1}, x_{2,4}, x_{2,5}, x_{2,6}),$$

$$min(x_{2,2}, x_{2,3}, x_{2,4}, x_{2,6}), min(x_{2,2}, x_{2,3}, x_{2,5}, x_{2,6}), min(x_{2,2}, x_{2,4}, x_{2,5}, x_{2,6})\}$$

and in terms of minimal cut sets,

$$\emptyset(\boldsymbol{x_2}) = min\{max(x_{2,1}, x_{2,2}), max(x_{2,3}x_{2,4}, x_{2,3}x_{2,5}, x_{2,4}x_{2,5}), max(x_{2,6})\}$$

Subsystem$_3$ has eight minimal path sets. These are

$$\boldsymbol{A_{3,1}} = \{x_{3,1}, x_{3,3}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\}, \boldsymbol{A_{3,2}} = \{x_{3,1}, x_{3,3}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\},$$
$$\boldsymbol{A_{3,3}} = \{x_{3,1}, x_{3,4}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\}, \boldsymbol{A_{3,4}} = \{x_{3,1}, x_{3,4}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\},$$
$$\boldsymbol{A_{3,5}} = \{x_{3,2}, x_{3,3}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\}, \boldsymbol{A_{3,6}} = \{x_{3,2}, x_{3,3}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\},$$
$$\boldsymbol{A_{3,7}} = \{x_{3,2}, x_{3,4}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\}, \boldsymbol{A_{3,8}} = \{x_{3,2}, x_{3,4}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}\}.$$

These are combined with seven minimal cut sets, namely,

$$\boldsymbol{C_{3,1}} = \{x_{3,1}, x_{3,2}\}, \boldsymbol{C_{3,2}} = \{x_{3,3}, x_{3,4}\}, \boldsymbol{C_{3,3}} = \{x_{3,5}, x_{3,6}\}, \boldsymbol{C_{3,4}} = \{x_{3,7}\}, \boldsymbol{C_{3,5}} = \{x_{3,8}\},$$
$$\boldsymbol{C_{3,6}} = \{x_{3,9}\}, \boldsymbol{C_{3,7}} = \{x_{3,10}\}.$$

leading to $\emptyset(\boldsymbol{x_3})$ in terms of minimal path sets,

$$\emptyset(\boldsymbol{x_3})$$
$$= max\{min(x_{3,1}, x_{3,3}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}), min(x_{3,1}, x_{3,3}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}),$$
$$min(x_{3,1}, x_{3,4}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}), min(x_{3,1}, x_{3,4}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}),$$
$$min(x_{3,2}, x_{3,3}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}), min(x_{3,2}, x_{3,3}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}),$$
$$min(x_{3,2}, x_{3,4}, x_{3,5}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}), min(x_{3,2}, x_{3,4}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10})\}$$

and in terms of minimal cut sets,

$$\emptyset(\boldsymbol{x_3}) = min\{max(x_{3,1}, x_{3,2}), max(x_{3,3}, x_{3,4}), max(x_{3,5}, x_{3,6}), max(x_{3,7}),$$
$$max(x_{3,8}), max(x_{3,9}), max(x_{3,10})\}$$

Subsystem$_4$ has only one minimal path set:

$$\boldsymbol{A_{4,1}} = \{x_{4,1}, x_{4,2}, x_{4,3}, x_{4,4}\}$$

This is combined with four single-element minimal cut sets, which are

$$\boldsymbol{C_{4,1}} = \{x_{4,1}\}, \boldsymbol{C_{4,2}} = \{x_{4,2}\}, \boldsymbol{C_{4,3}} = \{x_{4,3}\}, \boldsymbol{C_{4,4}} = \{x_{4,4}\}.$$

Since the vector $\boldsymbol{x_4}$ is a pure series arrangement, its structure function $\emptyset(\boldsymbol{x_4})$ in terms of minimal path sets is similar to the representation in terms of minimal cut sets, and written as

$$\emptyset(\boldsymbol{x_4}) = max\{min(x_{4,1}, x_{4,2}, x_{4,3}, x_{4,4})\}$$
$$= min\{max(x_{4,1}), max(x_{4,2}), max(x_{4,3}), max(x_{4,4})\}$$
$$= min(x_{4,1}, x_{4,2}, x_{4,3}, x_{4,4})$$

Similarly, in order for subsystem_5 to be in a functioning state, there is only one minimal path set, namely,

$$\boldsymbol{A_{5,1}} = \{x_{5,1}, x_{5,2}, x_{5,3}, x_{5,4}, x_{5,5}\}$$

and five single-element minimal cut sets, namely,

$$\boldsymbol{C_{5,1}} = \{x_{5,1}\}, \boldsymbol{C_{5,2}} = \{x_{5,2}\}, \boldsymbol{C_{5,3}} = \{x_{5,3}\}, \boldsymbol{C_{5,4}} = \{x_{5,4}\}, \boldsymbol{C_{5,5}} = \{x_{5,5}\}$$

Therefore, the structure function $\emptyset(\boldsymbol{x_5})$ in terms of both minimal path and cut sets is written as

$$\emptyset(\boldsymbol{x_5}) = min(x_{5,1}, x_{5,2}, x_{5,3}, x_{5,4}, x_{5,5})$$

At system-level, there is one minimal path set:

$$\boldsymbol{A_1} = \{\emptyset(\boldsymbol{x_1}), \emptyset(\boldsymbol{x_2}), \emptyset(\boldsymbol{x_3}), \emptyset(\boldsymbol{x_4}), \emptyset(\boldsymbol{x_5})\}$$

and five single-element minimal cut sets:

$$\boldsymbol{C_1} = \{\emptyset(\boldsymbol{x_1})\}, \boldsymbol{C_2} = \{\emptyset(\boldsymbol{x_2})\}, \boldsymbol{C_3} = \{\emptyset(\boldsymbol{x_3})\}, \boldsymbol{C_4} = \{\emptyset(\boldsymbol{x_4})\}, \boldsymbol{C_5} = \{\emptyset(\boldsymbol{x_5})\}$$

The consequent ISMM system-level structure function is found by

$$\emptyset(\boldsymbol{x}) = \prod_{i=1}^{5}\left(max_{i,j}\prod_{k\in \boldsymbol{A}_{i,j}} x_{i,k}\right) = \prod_{i=1}^{5}\left(\prod_{j=1}^{q} max_{k\in \boldsymbol{C}_{i,j}}\, x_{i,k}\right)$$
$$= min\{\emptyset(\boldsymbol{x_1}), \emptyset(\boldsymbol{x_2}), \emptyset(\boldsymbol{x_3}), \emptyset(\boldsymbol{x_4}), \emptyset(\boldsymbol{x_5})\}$$

4.3 Reliability Based on Random Events

4.3.1 Model Formulation

In contrast to structure functions where the goal is to determine the functioning (or non-functioning) state of each security control, subsystem, and overall security system, the goal here is to provide a measure, using random events, of the degree of continuity of correct security service for a given control, subsystem, and the overall security system.

To demonstrate reliability based on random events, we first introduce the mathematical modelling within the subsystem of every ISMM layer, noting that the representation can be similarly used to encode lower levels of structural abstraction. Then, we demonstrate the modeling of the overall system-level reliability.

Let S_i denote the event that $subsystem_i$ of n_i security controls is functioning, and $x_{i,j}, i = 1, \dots, 5$ and $j = 1, \dots, n_i$, denote the event that control i, j is working; thus $\overline{x}_{i,j}$ denotes the failure of control i, j (the event of control i, j *not* working).

Now, if subsystem_i controls are arranged logically in series, then, subsystem_iwill be working if and only if all n_i controls are working. Hence, event S_i will be the intersection of all $x_{i,j}'s$ events as follows.

$$S_i = x_{i,1} \cap x_{i,2} \cap \dots \cap x_{i,n_i} \tag{4-11}$$

and the probability that subsystem_i is working, i.e., the reliability of subsystem_i, is given by

$$R_i = P(S_i) = P\left(x_{i,1} \cap x_{i,2} \cap \dots \cap x_{i,n_i}\right) = P(x_{i,1} x_{i,2} \dots x_{i,n_i}) \tag{4-12}$$

If failure events of subsystem_i are not independent (the random events $x_{i,1}, x_{i,2}, \dots, x_{i,n_i}$ interact), then we write

$$R_i = P(S_i) = P(x_{i,1})\, P\left(x_{i,2}/x_{i,1}\right) P\left(x_{i,3}/x_{i,1} x_{i,2}\right) \dots P(x_{i,n_i}/x_{i,1} x_{i,2} \dots x_{i,n_i-1}) \tag{4-13}$$

and when failure events are independent (the random events $x_{i,1}, x_{i,2}, \dots, x_{i,n_i}$ are independent), we write

$$R_i = P(S_i) = P\left(x_{i,1}\right) P\left(x_{i,2}\right) P\left(x_{i,3}\right) \dots P\left(x_{i,n_i}\right) \tag{4-14}$$

So,

$$\begin{aligned} R_i &= \prod_{j=1}^{n_i} P(x_{i,j}) \\ &= \prod_{j=1}^{n_i} R_{i,j}\,, \qquad where\ R_{i,j} = P(x_{i,j}) \end{aligned} \tag{4-15}$$

On the other side, if subsystem_i controls are arranged logically in parallel, then subsystem_i will be working if at least one of its n_i security controls is working, making it a so-called redundant

configuration. Therefore, event S_i will be the union of all $x_{i,j}'s, i = 1, \dots, 5$ and $j = 1, \dots, n_i$, events as follows:

$$S_i = x_{i,1} \cup x_{i,2} \cup \dots \cup x_{i,n_i} \tag{4-16}$$

And the probability that subsystem_i is working, i.e., the reliability of subsystem_i, is written by

$$R_i = P(S_i) = P\left(x_{i,1} \cup x_{i,2} \cup \dots \cup x_{i,n_i}\right) = P\left(x_{i,1} + x_{i,2} + \cdots + x_{i,n_i}\right) \tag{4-17}$$

If failure events of subsystem_i are not independent (the random events $x_{i,1}, x_{i,2}, \dots, x_{i,n_i}$ are dependent), then we write

$$\begin{aligned} R_i = P(S_i) = &\left[P\left(x_{i,1}\right) + P\left(x_{i,2}\right) + P\left(x_{i,3}\right) + \cdots + P\left(x_{i,n_i}\right)\right] \\ &- \left[P\left(x_{i,1}x_{i,2}\right) + P\left(x_{i,1}x_{i,3}\right) + \cdots + P\left(x_{i,j}x_{i,k}\right)_{j \neq k}\right] \\ &+ \cdots + (-1)^{n_i - 1} P\left(x_{i,1}x_{i,2} \dots x_{n_i}\right) \end{aligned} \tag{4-18}$$

The formula can be written using the probability of failure for subsystem_i instead. Parallel subsystem_i failure occurs if all security controls at layer i fail, in which case

$$R_i = 1 - P(\overline{x}_{i,1}, \overline{x}_{i,2}, \dots \overline{x}_{i,n_i}) \tag{4-19}$$

and when failure events are independent (the random events $x_{i,1}, x_{i,2}, \dots, x_{i,n_i}$ are independent) we write

$$R_i = P(S_i) = 1 - P\left(\overline{x}_{i,1}\right) P\left(\overline{x}_{i,2}\right) P\left(\overline{x}_{i,3}\right) \dots P\left(\overline{x}_{i,n_i}\right) \tag{4-20}$$

So,

$$\begin{aligned} R_i &= 1 - \prod_{j=1}^{n_i} P(\overline{x}_{i,j}) \\ &= 1 - \prod_{j=1}^{n_i} \overline{R}_{i,j}\,, \qquad where\ \overline{R}_{i,j} = P(\overline{x}_{i,j}) \end{aligned} \tag{4-21}$$

Next, to define reliability based on random events at system level, let S_{ISMM} denote the event that the security system will be functioning; consequently, R_{ISMM} denotes the reliability of the security system. Because ISMM assumes independence and a series logical arrangement among individual

subsystems, the overall security system will be working if and only if all subsystems are working. Hence, event S_{ISMM} will be the intersection of all $S_i's, i = 1,2, \dots ,5$, events

$$S_{ISMM} = S_1 \cap S_2 \cap S_3 \cap S_4 \cap S_5 \tag{4-22}$$

and the probability of the overall system security working, i.e., the reliability of the ISMM-based security system, is then given by

$$\begin{aligned} R_{ISMM} &= P(S_{ISMM}) = P(S_1 \cap S_2 \cap S_3 \cap S_4 \cap S_5) \;\; due\ to\ series\ assumption \\ &= P(S_1 S_2 S_3 S_4 S_5) \\ &= P(S_1)\, P(S_2)\, P(S_3)\, P(S_4)\, P(S_5) \;\; due\ to\ independence\ assumption \end{aligned} \tag{4-23}$$

So,

$$\begin{aligned} R_{ISMM} &= \prod_{i=1}^{n=5} P(S_i) \\ &= \prod_{i=1}^{n=5} R_i\,, \qquad where\ R_i = P(S_i) \end{aligned} \tag{4-24}$$

Figure 4-1 shows how random events denoting security controls can be modelled on corresponding ISMM layers.

Event S_5	Definite Security $\{x_{5,1}, x_{5,2}, x_{5,3}, \dots, x_{5,n_5}\}$	$Subsystem_5$
Event S_4	Comprehensive Security Awareness $\{x_{4,1}, x_{4,2}, x_{4,3}, \dots, x_{4,n_4}\}$	$Subsystem_4$
Event S_3	Back-end System Security $\{x_{3,1}, x_{3,2}, x_{3,3}, \dots, x_{3,n_3}\}$	$Subsystem_3$
Event S_2	Front-end System Security $\{x_{2,1}, x_{2,2}, x_{2,3}, \dots, x_{2,n_2}\}$	$Subsystem_2$
Event S_1	Physical and Environmental Security $\{x_{1,1}, x_{1,2}, x_{1,3}, \dots, x_{1,n_1}\}$	$Subsystem_1$

Figure 4-1: ISMM-based random events modelling

4.3.2 Example

Recall the case study and associated structure functions in Section 3.11. The reliability functions for individual ISMM subsystems, assuming independence of failure events, are written by

$$R_1 = P(S_1)$$

$$= \left[1 - P\left(\overline{x}_{1,1}\right) P\left(\overline{x}_{1,2}\right)\right]\left[1 - P\left(\overline{x}_{1,3}\right) P\left(\overline{x}_{1,4}\right)\right]\left[1 - P\left(\overline{x}_{1,5}\right) P\left(\overline{x}_{1,6}\right)\right]\left[1 - P\left(\overline{x}_{1,7}\right) P\left(\overline{x}_{1,8}\right)\right]$$

$$= \left[1 - \overline{R}_{1,1}\overline{R}_{1,2}\right]\left[1 - \overline{R}_{1,3}\overline{R}_{1,4}\right]\left[1 - \overline{R}_{1,5}\overline{R}_{1,6}\right]\left[1 - \overline{R}_{1,7}\overline{R}_{1,8}\right].$$

$$R_2 = P(S_2) = \left[1 - P\left(\overline{x}_{2,1}\right) P\left(\overline{x}_{2,2}\right)\right]\left[P\left(x_{2,3}\right) P\left(x_{2,4}\right) P\left(x_{2,5}\right) + P\left(\overline{x}_{2,3}\right) P\left(x_{2,4}\right) P\left(x_{2,5}\right) + P\left(x_{2,3}\right) P\left(\overline{x}_{2,4}\right) P\left(x_{2,5}\right) + P\left(x_{2,3}\right) P\left(x_{2,4}\right) P\left(\overline{x}_{2,5}\right)\right] P\left(x_{2,6}\right)$$

$$= \left[1 - \overline{R}_{2,1}\overline{R}_{2,2}\right]\left[R_{2,3}R_{2,4} + R_{2,3}R_{2,5} + R_{2,4}R_{2,5} - 2R_{2,3}R_{2,4}R_{2,5}\right]R_{2,6}.$$

$$R_3 = P(S_3) = \left[1 - \overline{R}_{3,1}\overline{R}_{3,2}\right]\left[1 - \overline{R}_{3,3}\overline{R}_{3,4}\right]\left[1 - \overline{R}_{3,5}\overline{R}_{3,6}\right]R_{3,7}R_{3,8}R_{3,9}R_{3,10}.$$

$$R_4 = P(S_4) = R_{4,1}R_{4,2}R_{4,3}R_{4,4}.$$

$$R_5 = P(S_5) = R_{5,1}R_{5,2}R_{5,3}R_{5,4}R_{5,5}.$$

As the overall security system will be working if and only if all of its subsystems are working, event S_{ISMM} will be the intersection of all $S_i, i = 1,2, \ldots ,5,$ events, as follows

$$S_{ISMM} = S_1 \cap S_2 \cap S_3 \cap S_4 \cap S_5$$

So, the reliability is derived as

$$R_{ISMM} = \prod_{i=1}^{n=5} P(S_i) = \prod_{i=1}^{n=5} R_i \,, where\ R_i = \ P(S_i)$$

This leads to writing ISMM-based reliability in the form

$$R_{ISMM}$$
$$= \left[1 - \overline{R}_{1,1}\overline{R}_{1,2}\right]\left[1 - \overline{R}_{1,3}\overline{R}_{1,4}\right]\left[1 - \overline{R}_{1,5}\overline{R}_{1,6}\right]\left[1 - \overline{R}_{1,7}\overline{R}_{1,8}\right]\left[1 - \overline{R}_{2,1}\overline{R}_{2,2}\right]\left[R_{2,3}R_{2,4} + R_{2,3}R_{2,5} + R_{2,4}R_{2,5} - 2R_{2,3}R_{2,4}R_{2,5}\right]R_{2,6}\left[1 - \overline{R}_{3,1}\overline{R}_{3,2}\right]\left[1 - \overline{R}_{3,3}\overline{R}_{3,4}\right]\left[1 - \overline{R}_{3,5}\overline{R}_{3,6}\right]R_{3,7}R_{3,8}R_{3,9}R_{3,10}R_{4,1}R_{4,2}R_{4,3}R_{4,4}R_{5,1}R_{5,2}R_{5,3}R_{5,4}R_{5,5}$$

4.4 Reliability Based on Random Variables

4.4.1 Model Formulation

Reliability is obtained from the probability of violating a limit (threshold state). The limit in the security context herein is defined to be the functional state of security controls and system, demonstrated by the three essential information security functions or goals: confidentiality, integrity and availability. In this section, we present first reliability analysis based on static probabilities, then we present the case when system life is modelled as a function of controls' lives, using time-dependent random variables.

Time-independent reliabilities: In a way similar to the modeling approach using random events, the reliability measure, i.e., the degree of continuity of correct security service, is defined and evaluated based on random variables.

Let $\mathrm{X}_{i,j}$ be a random variable representing the state of the *j*th security control at the *i*th layer or subsystem, $i = 1, \dots, 5$ and $j = 1, \dots, n_i$. The value $P\{X_{i,j} = 1\} = p_{i,j}$, the probability that control $X_{i,j}$ is functioning, is called the reliability of control $X_{i,j}$, and can be defined as

$$R_{i,j} = P\{\mathrm{X}_{i,j} = 1\} = 1 - P\{\mathrm{X}_{i,j} = 0\} = p_{i,j} \tag{4-25}$$

and similarly,

$$R_i = P\{\emptyset(\boldsymbol{x_i}) = 1\} = 1 - P\{\emptyset(\boldsymbol{x_i}) = 0\} \quad \text{where } \boldsymbol{x_i} = (X_{i,1}, X_{i,2}, \dots, X_{i,n_i}) \tag{4-26}$$

is called the reliability of $subsystem_i, for\ all\ i = 1, \dots, 5$. Since $\emptyset(\boldsymbol{x_i})$ is a Bernoulli random variable, the reliability of $subsystem_i, R_i$, can be computed by taking the expectation, that is

$$R_i = P\{\emptyset(\boldsymbol{x_i}) = 1\} = E[\emptyset(\boldsymbol{x_i})] \tag{4-27}$$

For subsystems that consist of security controls connected in series, when the random variables $X_{i,1}, X_{i,2}, \dots, X_{i,n_i}$ are independent (i.e., controls do not interact with respect to a functioning state) we can write

$$R_i = P\{X_{i,1} = 1\}P\{X_{i,2} = 1\} \dots P\{X_{i,n_i} = 1\} = \prod_{j=1}^{n_i} p_{i,j} \tag{4-28}$$

And when the random variables $X_{i,1}, X_{i,2}, \dots, X_{i,n_i}$ are dependent we write

$$\begin{aligned} R_i = P\{X_{i,1} = 1\} P\{X_{i,2} = 1 / X_{i,1} = 1\} \dots \\ P\{X_{i,n_i} = 1 / X_{i,1} = 1, X_{i,2} = 1, \dots X_{i,n_i-1} = 1\} \end{aligned} \tag{4-29}$$

Also, for subsystems that consist of security controls connected in parallel, when the random variables $X_{i,1}, X_{i,2}, \dots, X_{i,n_i}$ are independent, with $q_{i,j} = 1 - p_{i,j}$, we can write

$$\begin{aligned} R_i &= 1 - P\{X_{i,1} = 0\} P\{X_{i,2} = 0\} \dots P\{X_{i,n_i} = 0\} \\ &= 1 - \prod_{j=1}^{n_i} (1 - p_{i,j}) = 1 - \prod_{j=1}^{n_i} q_{i,j} \end{aligned} \tag{4-30}$$

And when the random variables $X_{i,1}, X_{i,2}, \dots, X_{i,n_i}$ are dependent we write

$$\begin{aligned} R_i = 1 - P\{X_{i,1} = 0\} P\{X_{i,2} = 0 / X_{i,1} = 0\} \dots \\ P\{X_{i,n_i} = 0 / X_{i,1} = 0, X_{i,2} = 0, \dots, X_{i,n_i-1} = 0\} \end{aligned} \tag{4-31}$$

The same logic applies to mixed structure subsystems (subsystems that contain both parallel and series security controls at the same time). Accordingly, the probability of the overall system security working, i.e., the reliability of the ISMM-based security system, is then given by

$$\begin{aligned} R_{ISMM} &= P\{\emptyset(\boldsymbol{x}) = 1\} \\ &= 1 - P\{\emptyset(\boldsymbol{x}) = 0\} \\ &wehre\ \boldsymbol{x} = (\emptyset(\boldsymbol{x_1}), \emptyset(\boldsymbol{x_2}), \emptyset(\boldsymbol{x_3}), \emptyset(\boldsymbol{x_4}), \emptyset(\boldsymbol{x_5})) \end{aligned} \tag{4-32}$$

As a result of series logical arrangement and independence among ISMM layers,

$$\begin{aligned} R_{ISMM} &= P\{\emptyset(\boldsymbol{x_1}) = 1\} P\{\emptyset(\boldsymbol{x_2}) = 1\} P\{\emptyset(\boldsymbol{x_3}) = 1\} P\{\emptyset(\boldsymbol{x_4}) = 1\} P\{\emptyset(\boldsymbol{x_5}) = 1\} \\ &= \prod_{i=1}^{n=5} P\{\emptyset(\boldsymbol{x_i}) = 1\} \end{aligned} \tag{4-33}$$

So,

$$R_{ISMM} = R_1 \times R_2 \times R_3 \times R_4 \times R_5 = \prod_{i=1}^{n=5} R_i \tag{4-34}$$

Time-dependent reliabilities: Variable probabilities are more commonly used in reliability analysis, as mentioned in Section 2.2. In this modeling, reliability functions are defined over time.

To do so, let us assume that $T_{i,j}$ is a random variable that represents time taken for a failure event of control $x_{i,j}$ to occur, whether the failure is due to malicious or nonmalicious causes. We can informally think of the input space to this variable as the set of all possible events that cause a particular security control to fail, including both those involved in normal operational use and intentional attacks. The reliability of control $x_{i,j}$ is then defined by

$$\begin{aligned} R_{i,j}(t) = P_{i,j}(t) &= P\{lifetime\ of\ control\ j\ at\ layer\ i > t\} \\ &= 1 - F_{i,j}(t) \end{aligned} \tag{4-35}$$

The reliability for $subsystem_i, i = 1, \dots, 5$, is also defined in terms of the probability of security failure as a function of time and written as

$$R_i(t) = P(T_i > t) = 1 - F_i(t) \tag{4-36}$$

This function means $\text{subsystem}_\text{i}$ will function for time t or greater if and only if it is still functioning at time t. Analogical to the definition in conventional reliability, $R_i(t)$ is a monotonic non-increasing function of t with unity at the start of life: $R_i(0) = 1$ and $R_i(\infty) = 0$; $F_i(t)$, failure distribution, however, represents the probability that $\text{subsystem}_\text{i}$ will fail before time t, and equals

$$F_i(t) = P(T_i \leq t) = 1 - R_i(t), i = 1, \dots, 5 \tag{4-37}$$

In order to compute reliability at every ISMM layer subsystem, we follow the same technique used in reliability analysis of decomposing the subsystem at every layer into a set of controls, or components, where their reliabilities are known or to be computed with respect to time, so

$$R_i(t) = R\left(P_{i,1}(t), P_{i,2}(t), \dots, P_{i,n_i}(t)\right), for\ i = 1, \dots, 5 \text{ and } j = 1, \dots, n_i \tag{4-38}$$

For series $\text{subsystem}_\text{i}$ with independent n_i controls we can write

$$R_i(t) = R_{i,1}(t) \times R_{i,2}(t) \times \dots \times R_{i,n_i}(t) = \prod_{j=1}^{n_i} R_{i,j}(t) \tag{4-39}$$

Also, for parallel $\text{subsystem}_\text{i}$ with independent n_i controls we can write

$$\begin{aligned} R_i(t) &= 1 - \left[(1 - R_{i,1}(t)) \times (1 - R_{i,2}(t)) \times \dots \times (1 - R_{i,n_i}(t))\right] \\ &= 1 - \prod_{j=1}^{n_i} \left(1 - R_{i,j}(\text{t})\right) \end{aligned} \tag{4-40}$$

$$= 1 - \prod_{j=1}^{n_i} F_{i,j}(t)$$

Accordingly, the probability that the overall system security is working at time t, i.e., the reliability of the ISMM-based security system, is given by

$$R_{ISMM}(t) = R\big(P_1(t), P_2(t), P_3(t), P_4(t), P_5(t)\big), for\ i = 1, \dots, 5 \tag{4-41}$$

where

$$P_i(t) = P\{lifetime\ of\ subsystem_i > t\} = 1 - F_i(t), for\ i = 1, \dots, 5 \tag{4-42}$$

Similarly, the fact that ISMM layers are assumed to be independent and follow the series logical arrangement results in

$$R_{ISMM}(t) = R_1(t) \times R_2(t) \times R_3(t) \times R_4(t) \times R_5(t) = \prod_{i=1}^{n=5} R_i(t) \tag{4-43}$$

also,

$$R_{ISMM}(t) = P_1(t) \times P_2(t) \times P_3(t) \times P_4(t) \times P_5(t) = \prod_{i=1}^{n=5} P_i(t) \tag{4-44}$$

However, one must note that both R_{ISMM} and R_i can be expressed as functions of subsystems and controls' reliabilities, respectively; that is, $R_{ISMM} = R(\boldsymbol{P_{ISMM}})$ and $R_i = R(\boldsymbol{P_i})$. These functions are also monotonic increasing functions of the vectors $\boldsymbol{P_i}$ and $\boldsymbol{P_{ISMM}}$, respectively, where $\boldsymbol{P_i} = \big(P_{i,1}(t), P_{i,2}(t), \dots, P_{i,n_i}(t)\big)$ and $\boldsymbol{P_{ISMM}} = (P_1(t), P_2(t), P_3(t), P_4(t), P_5(t))$.

As mentioned earlier in Section 2.2.4, a *k*-out-of-*n* structure with identical and independent controls is represented using the binomial distribution, allowing its reliability function to be written for $\text{subsystem}_\text{i}$, as an example, in the form

$$R(\boldsymbol{p_i}) = \sum_{j=k}^{n_i} \binom{n_i}{j} {p_{i,j}}^j (1 - p_{i,j})^{n_i - j} \tag{4-45}$$

where $p_{i,j} = p_{i,1} = p_{i,2} = \cdots = p_{i,n_i}$

Note that the same complexity found in traditional reliability associated with calculating the reliability of a *k*-out-of-*n* system when its components are not identical is inherited here too, as calculating the reliability requires the state enumeration approach to sum up all the probabilities of possible system permutations with the number of the working controls is not less than k.

Moreover, the same key observations about pure series and parallel arrangements are applicable at any ISMM structure. We demonstrate those observations first at subsystem level and then show the extension to system level.

Firstly, for a series arrangement subsystem, the larger the number of controls connected in series, the lower the reliability of the subsystem. Therefore, the reliability of $subsystem_i$ having $(n_i + 1)$ controls in a series arrangement is upper-bounded by the reliability of the same subsystem having (n_i) controls. Considering $R_{n_i+1}(t) < 1$, we write

$$\begin{aligned} &R_{i,1}(t) \times R_{i,2}(t) \times \ldots \times R_{i,n_i}(t) \\ &\quad > R_{i,1}(t) \times R_{i,2}(t) \times \ldots \times R_{i,n_i}(t) \times R_{n_i+1}(t) \end{aligned} \tag{4-46}$$

and in its compact form

$$\prod_{j=1}^{n_i} R_{i,j}(t) > \prod_{j=1}^{n_i+1} R_{i,j}(t) \tag{4-47}$$

Also, the reliability of the subsystem in series arrangement is smaller than the reliability of its least reliable control, and the subsystem reliability decreases if any of its controls reliability decreases. Therefore, the reliability of the series subsystem is upper-bounded by its weakest link. Let $R_{i,s}(t)$ denote the least reliable control in the $subsystem_i$ series arrangement of n_i controls, where $s \in \{1, \ldots, n_i\}$ and $R_{\mathrm{i},s}(t) < 1$; we write

$$R_{i,1}(t) \times R_{i,2}(t) \times \ldots \times R_{i,n_i}(t) = \prod_{j=1}^{n_i} R_{i,j}(t) < R_{i,s}(t) \tag{4-48}$$

This observation also applies to the system-level structure due to the assumption of independence and logical series arrangement among individual ISMM layers. Thus, the reliability of the ISMM-based security system is smaller than the reliability of its least reliable subsystem. Additionally, overall system security reliability decreases if any subsystem's reliability decreases. In a nutshell, the reliability of the overall security system is upper-bounded by its weakest link, or subsystem. Let $R_s(t)$ denote the least reliable subsystem, where $s \in \{1, \dots, 5\}$ and $R_s(t) < 1$; we can write

$$R_1(t) \times R_2(t) \times R_3(t) \times R_4(t) \times R_5(t) = \prod_{i=1}^{n=5} R_i(t) < R_s(t) \tag{4-49}$$

Secondly, for a parallel arrangement subsystem, the larger the number of controls in parallel, the larger the reliability of the subsystem. Therefore, the reliability of the $(n_i + 1)$ controls parallel subsystem is lower-bounded by the reliability of the same subsystem having (n_i) controls. Adding an extra control $(n_i + 1)$ for a parallel arrangement of n_i controls, $R_{n_i+1}(t) < 1$, we write

$$\begin{aligned} &1 - \left[(1 - R_{i,1}(t)) \times (1 - R_{i,2}(t)) \times \dots \times (1 - R_{i,n_i}(\mathrm{t}))\right] \\ &< 1 - \left[(1 - R_{i,1}(t)) \times (1 - R_{i,2}(t)) \times \dots \times (1 - R_{i,n_i}(\mathrm{t})) \times (1 - R_{n_i+1}(\mathrm{t}))\right] \end{aligned} \tag{4-50}$$

and in its compact form

$$1 - \prod_{j=1}^{n_i} (1 - R_{i,j}(\mathrm{t})) < 1 - \prod_{j=1}^{n_i+1} \left(1 - R_{i,j}(\mathrm{t})\right) \tag{4-51}$$

Also, the reliability of the parallel subsystem is larger than the reliability of its most reliable control, and hence, lower-bounded by this amount. Let $R_{i,l}(t), l \in \{1, \dots, n_i\}$ denote the most reliable control in a parallel arrangement of n_i controls; we write

$$\begin{aligned} &1 - \left[(1 - R_{i,1}(t)) \times (1 - R_{i,2}(t)) \times \dots \times (1 - R_{i,n_i}(\mathrm{t}))\right] \\ &= 1 - \prod_{j=1}^{n_i} (1 - R_{i,j}(\mathrm{t})) > R_{i,l}(t) \end{aligned} \tag{4-52}$$

As explained in [47], in the case of a system or subsystem with n components where individual control reliabilities are known but no or little information on their logical structure is available, one might use their series arrangement as the lower reliability bound and parallel arrangement at their upper reliability bound of the system. Another useful relationship in the case of identical controls can

be established if the number of controls, say $n_{i,j}$, to reach a known reliability goal, say $R_{i,j}^{g}$, is to be found. For a series structure, $n_{i,j}$ can be found according to the relation

$$R_{i,j}^{g}(t) = (R_{i,j}(t))^{n_{i,j}} \tag{4-53}$$

and for a parallel structure, $n_{i,j}$ can be found by

$$R_{i,j}^{g}(t) = 1 - (1 - R_{i,j}(t))^{n_{i,j}} \tag{4-54}$$

One must note that replicating components leads to a higher reliability than replicating systems [47], [48]. To conclude this part, there are four important functions related to reliability evaluation, regardless of the level of ISMM structure: 1) the failure density function, which describes how the failure probability is spread over the chosen measurement unit and denoted, at control level for instance, by $f_{i,j}(t)$. $f_{i,j}(t)$ is always non-negative and the total area beneath it is always equal to one as it is basically a probability distribution function, so

$$\int_0^\infty f_{i,j}(t)dt = 1\,, for\ i = 1, \dots, 5 \text{ and } j = 1, \dots, n_i \tag{4-55}$$

2) Failure distribution function, $F_{i,j}(t)$, which is the cumulative distribution function of the failure density function $f_{i,j}(t)$ and given by the relation

$$f_{i,j}(t) = \frac{d}{dt}F_{i,j}(t) = -\frac{d}{dt}R_{i,j}(t) \tag{4-56}$$

As mentioned earlier, $F_{i,j}(t)$ represents the probability that control j at layer or subsystem i will fail before time t, and gives the area beneath the failure density function until time t, and equals

$$F_{i,j}(t) = P_{i,j}(T \le t) = \int_0^t f_{i,j}(v)dv \tag{4-57}$$

3) Reliability function $R_{i,j}(t)$, which gives the area beneath the failure density function after time t, and equals

$$\begin{aligned} R_{i,j}(t) &= P_{i,j}(T > t) = 1 - F_{i,j}(t) \\ &= 1 - \int_0^t f_{i,j}(v)dv = exp\left[-\int_0^t \lambda_{i,j}(v)dv\right] \end{aligned} \tag{4-58}$$

4) Hazard function $\lambda_{i,j}(t)$, sometimes called instantaneous failure rate, which is defined as the limit of the failure rate as the interval length approaches zero. It equals

$$\lambda_{i,j}(t) = \frac{f_{i,j}(t)}{R_{i,j}(t)} = \frac{f_{i,j}(t)}{1 - F_{i,j}(t)} = -\frac{d}{dt}\left[\ln R_{i,j}(t)\right] \tag{4-59}$$

Another important parameter to these equations is the *mean time to failure* $MTTF$ (or, *mean effort to failure* when using the variable *effort* instead). It is the expected value of the continuous random variable T and gives the area beneath the reliability function. $MTTF$ is given by

$$MTTF_{i,j} = \int_0^\infty t f_{i,j}(t)dt = \int_0^\infty R_{i,j}(t)dt \tag{4-60}$$

4.4.2 Analytical Example

Time-independent reliabilities: Recall the case study and associated structure functions in Section 3.11. The reliability functions for ISMM subsystems, assuming the independence of random variables, can be formulated as follows.

$$R_1 = \left[1 - P\{X_{1,1} = 0\}P\{X_{1,2} = 0\}\right]\left[1 - P\{X_{1,3} = 0\}P\{X_{1,4} = 0\}\right]$$
$$\left[1 - P\{X_{1,5} = 0\}P\{X_{1,6} = 0\}\right]\left[1 - P\{X_{1,7} = 0\}P\{X_{1,8} = 0\}\right]$$
$$= \left[1 - (1 - p_{1,1})(1 - p_{1,2})\right]\left[1 - (1 - p_{1,3})(1 - p_{1,4})\right]$$
$$\left[1 - (1 - p_{1,5})(1 - p_{1,6})\right]\left[1 - (1 - p_{1,7})(1 - p_{1,8})\right]$$
$$= \left[1 - q_{1,1}q_{1,2}\right]\left[1 - q_{1,3}q_{1,4}\right]\left[1 - q_{1,5}q_{1,6}\right]\left[1 - q_{1,7}q_{1,8}\right].$$

$$R_2 = \left[1 - P\{X_{2,1} = 0\}P\{X_{2,2} = 0\}\right]\left[P\{X_{2,3} = 1\}P\{X_{2,4} = 1\}P\{X_{2,5} = 1\} + P\{X_{2,3} = 0\}P\{X_{2,4} = 1\}P\{X_{2,5} = 1\} + P\{X_{2,3} = 1\}P\{X_{2,4} = 0\}P\{X_{2,5} = 1\} + P\{X_{2,3} = 1\}P\{X_{2,4} = 1\}P\{X_{2,5} = 0\}\right]P\{X_{2,6} = 1\}$$
$$= \left[1 - q_{2,1}q_{2,2}\right]\left[p_{2,3}p_{2,4} + p_{2,3}p_{2,5} + p_{2,4}p_{2,5} - 2p_{2,3}p_{2,4}p_{2,5}\right]p_{2,6}.$$

$$R_3 = \left[1 - P\{X_{3,1} = 0\}P\{X_{3,2} = 0\}\right]\left[1 - P\{X_{3,3} = 0\}P\{X_{3,4} = 0\}\right]$$
$$\left[1 - P\{X_{3,5} = 0\}P\{X_{3,6} = 0\}\right]P\{X_{3,7} = 1\}P\{X_{3,8} = 1\}P\{X_{3,9} = 1\}P\{X_{3,10} = 1\}$$
$$= \left[1 - q_{3,1}q_{3,2}\right]\left[1 - q_{3,3}q_{3,4}\right]\left[1 - q_{3,5}q_{3,6}\right]p_{3,7}p_{3,8}p_{3,9}p_{3,10}.$$

$$R_4 = P\{X_{4,1} = 1\}P\{X_{4,2} = 1\}P\{X_{4,3} = 1\}P\{X_{4,4} = 1\}$$

$$= p_{4,1}p_{4,2}p_{4,3}p_{4,4}.$$

$$R_5 = P\{X_{5,1} = 1\}P\{X_{5,2} = 1\}P\{X_{5,3} = 1\}P\{X_{5,4} = 1\}P\{X_{5,5} = 1\}$$

$$= p_{5,1}p_{5,2}p_{5,3}p_{5,4}p_{5,5}.$$

Substituting these reliability functions into $R_{ISMM} = \prod_{i=1}^{n=5} Pr\{\emptyset(\mathbf{x_i}) = 1\} = \prod_{i=1}^{n=5} R_i$ to find the probability of the event that the overall system security is working, i.e., the reliability of the ISMM-based security system,

$$R_{ISMM} = R_1 \times R_2 \times R_3 \times R_4 \times R_5$$
$$= [1 - q_{1,1}q_{1,2}][1 - q_{1,3}q_{1,4}][1 - q_{1,5}q_{1,6}][1 - q_{1,7}q_{1,8}]$$
$$[1 - q_{2,1}q_{2,2}][p_{2,3}p_{2,4} + p_{2,3}p_{2,5} + p_{2,4}p_{2,5} - 2p_{2,3}p_{2,4}p_{2,5}]p_{2,6}$$
$$[1 - q_{3,1}q_{3,2}][1 - q_{3,3}q_{3,4}][1 - q_{3,5}q_{3,6}]p_{3,7}p_{3,8}p_{3,9}p_{3,10}$$
$$p_{4,1}p_{4,2}p_{4,3}p_{4,4}p_{5,1}p_{5,2}p_{5,3}p_{5,4}p_{5,5}$$

Time-dependent reliabilities: We demonstrate the reliability based on time-dependent random variables where reliability values can be calculated across different mission times, assuming the unit *time* as the variable upon which the failure event is defined; i.e., $T_{i,j}$ is a random variable that represents the time taken for a failure event for the *j*th control at the *i*th layer to occur. We build individual subsystem-level reliability functions as follows

$$R_1(t) = [1 - (1 - R_{1,1}(t)) \times (1 - R_{1,2}(t))][1 - (1 - R_{1,3}(t)) \times (1 - R_{1,4}(t))]$$
$$[1 - (1 - R_{1,5}(t)) \times (1 - R_{1,6}(t))][1 - (1 - R_{1,7}(t)) \times (1 - R_{1,8}(t))]$$

$$R_2(t) = [1 - (1 - R_{2,1}(t)) \times (1 - R_{2,2}(t))][R_{2,3}(t)R_{2,4}(t) + R_{2,3}(t)R_{2,5}(t) +$$
$$R_{2,4}(t)R_{2,5}(t) - 2R_{2,3}(t)R_{2,4}(t)R_{2,5}(t)]R_{2,6}(t)$$

$$R_3(t) = [1 - (1 - R_{3,1}(t)) \times (1 - R_{3,2}(t))][1 - (1 - R_{3,3}(t)) \times (1 - R_{3,4}(t))][1 -$$
$$(1 - R_{3,5}(t)) \times (1 - R_{3,6}(t))]R_{3,7}(t)R_{3,8}(t)R_{3,9}(t)R_{3,10}(t)$$

$$R_4(t) = R_{4,1}(t)R_{4,2}(t)R_{4,3}(t)R_{4,4}(t)$$

$$R_5(t) = R_{5,1}(t)R_{5,2}(t)R_{5,3}(t)R_{5,4}(t)R_{5,5}(t)$$

Accordingly, the probability that the overall system security is functioning in terms of the failure variable, time t, i.e., $R_{ISMM}(t)$, is calculated as

$$R_{ISMM}(t) = R_1(t) \times R_2(t) \times R_3(t) \times R_4(t) \times R_5(t)$$
$$= \big[1 - (1 - R_{1,1}(t)) \times (1 - R_{1,2}(t))\big]\big[1 - (1 - R_{1,3}(t)) \times (1 - R_{1,4}(t))\big]\big[1 -$$
$$(1 - R_{1,5}(t)) \times (1 - R_{1,6}(t))\big]\big[1 - (1 - R_{1,7}(t)) \times (1 - R_{1,8}(t))\big]\big[1 - (1 - R_{2,1}(t)) \times$$
$$(1 - R_{2,2}(t))\big]\big[R_{2,3}(t)R_{2,4}(t) + R_{2,3}(t)R_{2,5}(t) + R_{2,4}(t)R_{2,5}(t) -$$
$$2R_{2,3}(t)R_{2,4}(t)R_{2,5}(t)\big]R_{2,6}(t)\big[1 - (1 - R_{3,1}(t)) \times (1 - R_{3,2}(t))\big]\big[1 - (1 - R_{3,3}(t)) \times$$
$$(1 - R_{3,4}(t))\big]\big[1 - (1 - R_{3,5}(t)) \times (1 - R_{3,6}(t))\big]R_{3,7}(t)R_{3,8}(t)R_{3,9}(t)R_{3,10}(t)$$
$$R_{4,1}(t)R_{4,2}(t)R_{4,3}(t)R_{4,4}(t)\ R_{5,1}(t)R_{5,2}(t)R_{5,3}(t)R_{5,4}(t)R_{5,5}(t)$$

4.4.3 Numerical Example

Time-independent reliabilities: To demonstrate numerically ISMM-based reliability and consequent maturity analysis, we assume the fixed probabilities of failure of controls $P\{\mathrm{X}_{i,j} = 0\} = q_{i,j}$ demonstrated on each RBD below.

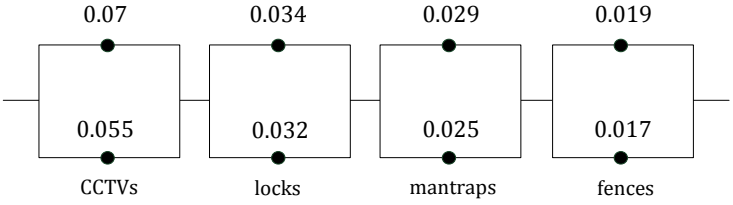

Figure 4-2: Example of $subsystem_1$ static reliabilities

$Subsystem_1$ can be treated as a series-parallel structure, as demonstrated in section 3.8. Recall that

$$R_1 = \big[1 - q_{1,1}q_{1,2}\big]\big[1 - q_{1,3}q_{1,4}\big]\big[1 - q_{1,5}q_{1,6}\big]\big[1 - q_{1,7}q_{1,8}\big]$$

Substituting the $q_{1,j}$ values in Figure 4-2 leads to

$$R_1 = [1 - 0.07 \times 0.055][1 - 0.034 \times 0.032][1 - 0.029 \times 0.025][1 - 0.019 \times 0.017]$$
$$= 0.99402$$

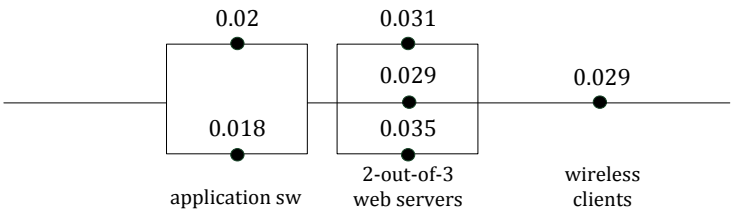

Figure 4-3: Example of $subsystem_2$ static reliabilities

Consequently, the reliability of $subsystem_2$ is written by

$$R_2 = [1 - q_{2,1}q_{2,2}][p_{2,3}p_{2,4} + p_{2,3}p_{2,5} + p_{2,4}p_{2,5} - 2p_{2,3}p_{2,4}p_{2,5}]p_{2,6}$$

Substituting the $q_{2,j}$ values demonstrated in Figure 4-3 leads to

$$R_2 = [1 - 0.02 \times 0.018][0.969 \times 0.971 + 0.969 \times 0.965 + 0.971 \times 0.965$$
$$- 2 \times 0.969 \times 0.971 \times 0.965]0.971$$
$$= 0.96780$$

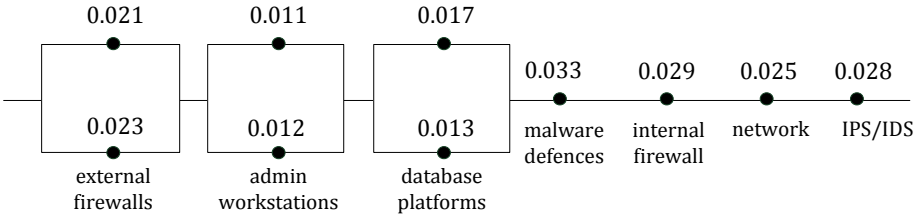

Figure 4-4: Example of $subsystem_3$ static reliabilities

The reliability of $subsystem_3$ is written by

$$R_3 = [1 - q_{3,1}q_{3,2}][1 - q_{3,3}q_{3,4}][1 - q_{3,5}q_{3,6}]p_{3,7}p_{3,8}p_{3,9}p_{3,10}$$

Substituting the $q_{3,j}$ values in Figure 4-4 leads to

$$R_3 = [1 - 0.021 \times 0.023][1 - 0.011 \times 0.012][1 - 0.017 \times 0.013]0.967$$
$$\times 0.971 \times 0.975 \times 0.972$$
$$= 0.88911$$

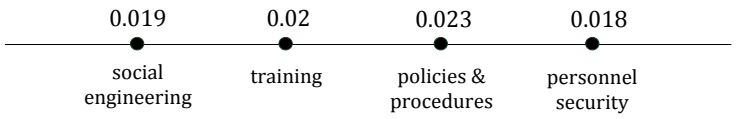

Figure 4-5: Example of $subsystem_4$ static reliabilities

Similarly, $subsystem_4$ is written by

$$R_4 = p_{4,1}p_{4,2}p_{4,3}p_{4,4}$$

Substituting the $q_{4,j}$ values in Figure 4-5 leads to

$$R_4 = 0.981 \times 0.98 \times 0.977 \times 0.982$$
$$= 0.92236$$

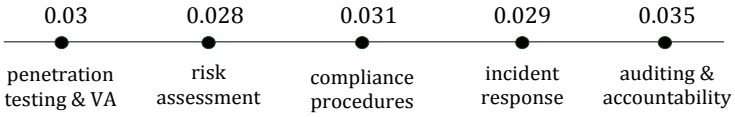

Figure 4-6: Example of $subsystem_5$ static reliabilities

and $subsystem_5$ is written by

$$R_5 = p_{5,1}p_{5,2}p_{5,3}p_{5,4}p_{5,5}$$

Substituting the $q_{5,j}$ values in Figure 4-6 leads to

$$R_5 = 0.97 \times 0.972 \times 0.969 \times 0.971 \times 0.965$$
$$= 0.85607$$

For the overall system, i.e., $system_{ISMM}$, the corresponding reliability is written by

$$R_{ISMM} = \prod_{i=1}^{n=5} R_i$$
$$= [1 - q_{1,1}q_{1,2}][1 - q_{1,3}q_{1,4}][1 - q_{1,5}q_{1,6}][1 - q_{1,7}q_{1,8}]$$
$$[1 - q_{2,1}q_{2,2}][p_{2,3}p_{2,4} + p_{2,3}p_{2,5} + p_{2,4}p_{2,5} - 2p_{2,3}p_{2,4}p_{2,5}]p_{2,6}$$

$$[1-q_{3,1}q_{3,2}][1-q_{3,3}q_{3,4}][1-q_{3,5}q_{3,6}]p_{3,7}p_{3,8}p_{3,9}p_{3,10}$$
$$p_{4,1}p_{4,2}p_{4,3}p_{4,4}p_{5,1}p_{5,2}p_{5,3}p_{5,4}p_{5,5}$$

Substituting the $q_{i,j}$ values for all subsystems leads to

$$R_{ISMM} = 0.99402 \times 0.96780 \times 0.88911 \times 0.92236 \times 0.85607$$
$$= 0.67538$$

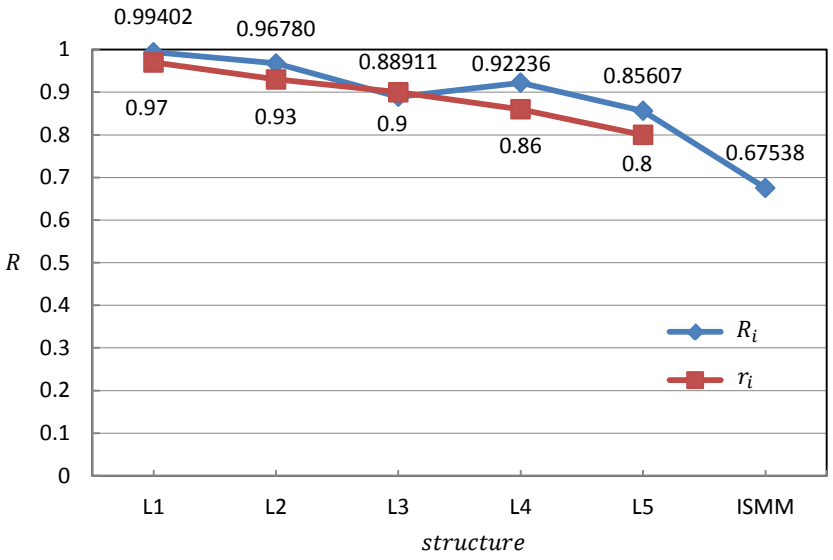

Figure 4-7: ISMM-based reliability analysis for time-independent reliabilities

Maturity analysis: Recall that we use reliability as the performance measure defining the maturity measure, thus maturity is a function of the reliability measure of individual subsystems, i.e., $\mathcal{M}_{ISMM} = \mathcal{M}_R$. This measure can be found graphically, as presented earlier in Figure 4-7, or analytically, as shown here. Recall the maturity function equations in (3-14), (3-15), and (3-16). The maturity test can be established using the acceptability function as

$$I\big(R_i \geq threshold(R_i)\big) = I(R_i \geq r_i), for\ all\ i = 1, \ldots, 5$$

Applying this leads to constructing the following tests,

$$I(R_1 \geq r_1) = I(R_2 \geq r_2) = 1$$
$$I(R_3 \geq r_3) = 0$$
$$I(R_4 \geq r_4) = I(R_5 \geq r_5) = 1$$

As a result,

$$\mathcal{M}_{ISMM} = \mathcal{M}_R = \left\{ max(m)\, s.t. \left\{ \prod_{i=1}^{m} I\big(R_i \geq threshold(R_i)\big) \right\} = 1 \right\} = 2$$

Observe that the maturity remains equal to 2 due to the test $I(R_3(1) \geq r_3)$, regardless of $I(R_4(1) \geq r_4)$ and $I(R_5(1) \geq r_5)$, thus preserving the precedence of controls with respect to their intended strengths and failure boundaries of the system. The maturity deficiency, however, in R_3 can simply be found by $r_3 - R_3$.

Another important observation is the distinction between reliability analysis and maturity analysis. When calculating the reliability measure, the permutations of the values of individual subsystems' reliabilities $R_i's$, a total of 5! permutations, always lead to the same computed system reliability R_{ISMM}. In contrast, such permutations lead to different groups or classes of computed maturity, in the range $[0 \ldots 5]$, distributed according to their minimum bounds $r_i's$ and corresponding reliabilities $R_i's$. The sample permutations in Table 4-2, using the minimum bounds in Table 4-1, demonstrate this feature.

Table 4-1: ISMM-based reliability minimum bounds

R_i	R_1	R_2	R_3	R_4	R_5
r_i	0.97	0.93	0.90	0.86	0.80

Table 4-2: Comparison between reliability and maturity measures[6]

Permutation no.	R_1	R_2	R_3	R_4	R_5	R_{ISMM}	$\mathcal{M}_{ISMM}$
1	0.99402	0.96780	0.88911	0.92236	0.85607	0.67538	2
2	0.96780	0.99402	0.88911	0.92236	0.85607	0.67538	0
3	0.99402	0.88911	0.96780	0.92236	0.85607	0.67538	1
4	0.99402	0.96780	0.92236	0.88911	0.85607	0.67538	5
5	0.99402	0.96780	0.88911	0.85607	0.92236	0.67538	2
…						…	…
5!	0.99402	0.96780	0.92236	0.85607	0.88911	0.67538	3

Time-dependent reliabilities: To simplify calculations in this part we assume that control failures exhibit the exponential model, i.e., a constant failure rate, nonrepairable controls until subsystem-level (and therefore, system-level too) failure, and independent controls, where redundant controls are active[7] [47], [115]. For each subsystem, we show the reliability formulation of both cases: identical redundant controls and dissimilar redundant controls. We continue the analysis, however, considering the identical case to reduce mathematical computations. Recall that the exponential model implies the following functions.

$$R_{i,j}(t) = e^{-\lambda_{i,j}t}, for\ all\ i = 1, \dots, 5\ and\ j = 1, \dots, n_i$$

$$\lambda_{i,j}(t) = \lambda_{i,j}$$

$$MTTF_{i,j} = \frac{1}{\lambda_{i,j}}$$

***Subsystem*$_1$ analysis.** Substituting the exponential model into the analytical formulations in Section 4.4.2, we obtain in the case of dissimilar controls (redundant controls are not identical),

[6] Reliabilities in red indicate the changes from the original subsystems' reliabilities.
[7] Active redundancy (or parallel, hot structure) means redundant control(s) and operating control(s) are subjected to the same load with the same failure rate.

$$R_1(t) = \left[R_{1,1}(t) + R_{1,2}(t) - R_{1,1}(t)R_{1,2}(t)\right]\left[R_{1,3}(t) + R_{1,4}(t) - R_{1,3}(t)R_{1,4}(t)\right]\left[R_{1,5}(t) + R_{1,6}(t) - R_{1,5}(t)R_{1,6}(t)\right]\left[R_{1,7}(t) + R_{1,8}(t) - R_{1,7}(t)R_{1,8}(t)\right]$$

$$= \left[e^{-\lambda_{1,1}t} + e^{-\lambda_{1,2}t} - e^{-(\lambda_{1,1}+\lambda_{1,2})t}\right]\left[e^{-\lambda_{1,3}t} + e^{-\lambda_{1,4}t} - e^{-(\lambda_{1,3}+\lambda_{1,4})t}\right]\left[e^{-\lambda_{1,5}t} + e^{-\lambda_{1,6}t} - e^{-(\lambda_{1,5}+\lambda_{1,6})t}\right]\left[e^{-\lambda_{1,7}t} + e^{-\lambda_{1,8}t} - e^{-(\lambda_{1,7}+\lambda_{1,8})t}\right]$$

Observe that the above equation yields $3 \times 3 \times 3 \times 3$ terms. For independent identical redundancy, yielding $2 \times 2 \times 2 \times 2$ terms, we obtain

$$R_1(t) = \left[2R_{1,1}(t) - \left(R_{1,1}(t)\right)^2\right]\left[2R_{1,3}(t) - \left(R_{1,3}(t)\right)^2\right]\left[2R_{1,5}(t) - \left(R_{1,5}(t)\right)^2\right]\left[2R_{1,7}(t) - \left(R_{1,7}(t)\right)^2\right]$$

$$= \left[2e^{-\lambda_{1,1}t} - e^{-2\lambda_{1,1}t}\right]\left[2e^{-\lambda_{1,3}t} - e^{-2\lambda_{1,3}t}\right]\left[2e^{-\lambda_{1,5}t} - e^{-2\lambda_{1,5}t}\right]\left[2e^{-\lambda_{1,7}t} - e^{-2\lambda_{1,7}t}\right]$$

$$= 16e^{-(\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7})t} - 8e^{-(\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7})t} - 8e^{-(\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7})t} + 4e^{-(\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7})t} - 8e^{-(\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7})t} + 4e^{-(\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7})t} + 4e^{-(\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7})t} - 2e^{-(\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7})t} - 8e^{-(2\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7})t} + 4e^{-(2\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7})t} + 4e^{-(2\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7})t} - 2e^{-(2\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7})t} + 4e^{-(2\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7})t} - 2e^{-(2\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7})t} - 2e^{-(2\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7})t} + e^{-(2\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7})t}$$

Integrating the complete expression of $R_1(t)$ from zero to infinity, we obtain the corresponding mean time to failure as follows.

$$MTTF_1$$

$$= \frac{16}{\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7}} - \frac{8}{\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7}} - \frac{8}{\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7}}$$
$$+ \frac{4}{\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7}} - \frac{8}{\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7}} + \frac{4}{\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7}}$$
$$+ \frac{4}{\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7}} - \frac{2}{\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7}}$$
$$- \frac{8}{2\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7}} + \frac{4}{2\lambda_{1,1}+\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7}} + \frac{4}{2\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7}}$$
$$- \frac{2}{2\lambda_{1,1}+\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7}} + \frac{4}{2\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+\lambda_{1,7}}$$
$$- \frac{2}{2\lambda_{1,1}+2\lambda_{1,3}+\lambda_{1,5}+2\lambda_{1,7}} - \frac{2}{2\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+\lambda_{1,7}}$$
$$+ \frac{1}{2\lambda_{1,1}+2\lambda_{1,3}+2\lambda_{1,5}+2\lambda_{1,7}}$$

Table 4-3: Exponential model failure rates for $subsystem_1$

Control $C_{1,j}$	Failure rate $\lambda_{1,j}$ $(week^{-1})$
$C_{1,1}$	0.09
$C_{1,2}$	0.09
$C_{1,3}$	0.08
$C_{1,4}$	0.08
$C_{1,5}$	0.10
$C_{1,6}$	0.10
$C_{1,7}$	0.075
$C_{1,8}$	0.075

Consider that $subsystem_1$ exhibits the failure rates of individual controls in Table 4-3. Figure 4-8 demonstrates the corresponding performance of the four series 1-out-of-2 structures individually,

named $S_{1,1}, S_{1,2}, S_{1,3}$, and $S_{1,4}$, respectively, and the corresponding overall $subsystem_1$ reliability $R_1(t)$. Observe that all performance curves, including the mean time to failure $MTTF_1$ point, fall after the predetermined mission time ($t = 1$) and above the minimum reliability bound r_1 at this layer. As a result, the reliability performance of security controls at this layer meets the requirement set for the maturity measure. Note that subsystem reliability, i.e., R_1 in this case, will always represent the largest exponential decay among the rest of its series structures, as is consistent with the reliability observations described in section 4.4.1.

The numerical value of the reliability can be easily calculated once the mission time t is known. For instance, for a mission time of one week, i.e., $t = 1$,

$$R_1(1) = 0.9727$$

and the mean time to failure,

$$MTTF_1 = 6.7308\ weeks$$

In the case of the mission time set for one day, i.e., $t = \frac{1}{7}$, we obtain

$$R_1\left(\frac{1}{7}\right) = 0.9994$$

For a longer mission time, perhaps 30 days, i.e., $t = \frac{30}{7}$,

$$R_1\left(\frac{30}{7}\right) = 0.6675$$

However, if we want to limit the reliability to a minimum of some value, say three nines, meaning $R_1(t) = 0.999$, the corresponding mission time, as shown in

Figure 4-8, must be limited to $t = 0.18\ weeks \approx 1.26\ days = 30.2\ hours = 1814\ minutes$. This result can also be verified analytically by computing $R_1(0.18)$.

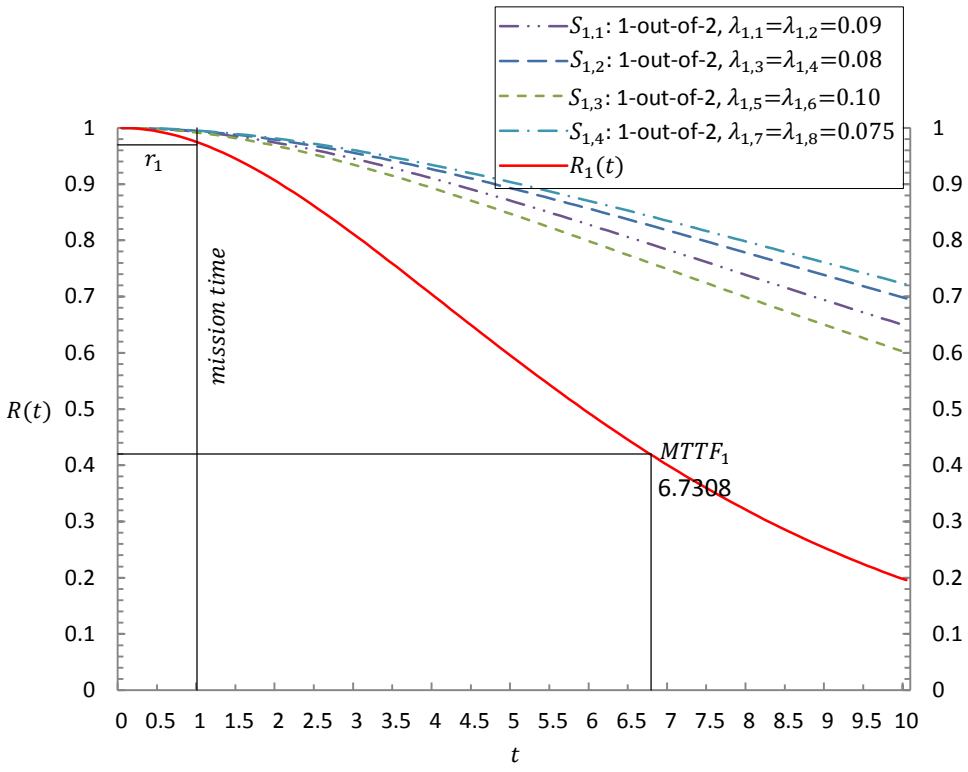

Figure 4-8: ISMM-based time-dependent reliability analysis for $subsystem_1$

***Subsystem*$_2$ analysis.** For the case of dissimilar redundancy in $subsystem_2$, we obtain

$$R_2(t) = \left[R_{2,1}(t) + R_{2,2}(t) - R_{2,1}(t)R_{2,2}(t)\right]\left[R_{2,3}(t)R_{2,4}(t) + R_{2,3}(t)R_{2,5}(t) + R_{2,4}(t)R_{2,5}(t) - 2R_{2,3}(t)R_{2,4}(t)R_{2,5}(t)\right]R_{2,6}(t)$$

$$= \left[e^{-\lambda_{2,1}t} + e^{-\lambda_{2,2}t} - e^{-(\lambda_{2,1}+\lambda_{2,2})t}\right]\left[e^{-(\lambda_{2,3}+\lambda_{2,4})t} + e^{-(\lambda_{2,3}+\lambda_{2,5})t} + e^{-(\lambda_{2,4}+\lambda_{2,5})t} - 2e^{-(\lambda_{2,3}+\lambda_{2,4}+\lambda_{2,5})t}\right]e^{-\lambda_{2,6}t}$$

and when the identical redundancy is considered, we obtain

$$R_2(t) = \left[2e^{-\lambda_{2,1}t} - e^{-2\lambda_{2,1}t}\right]\left[3e^{-2\lambda_{2,3}t} - 2e^{-3\lambda_{2,3}t}\right]e^{-\lambda_{2,6}t}$$

$$= 6e^{-(\lambda_{2,1}+2\lambda_{2,3}+\lambda_{2,6})t} - 4e^{-(\lambda_{2,1}+3\lambda_{2,3}+\lambda_{2,6})t} - 3e^{-(2\lambda_{2,1}+2\lambda_{2,3}+\lambda_{2,6})t} + 2e^{-(2\lambda_{2,1}+3\lambda_{2,3}+\lambda_{2,6})t}$$

Integrating the complete expression of $R_2(t)$ to obtain the corresponding $MTTF_2$ yields

$$MTTF_2 = \frac{6}{\lambda_{2,1} + 2\lambda_{2,3} + \lambda_{2,6}} - \frac{4}{\lambda_{2,1} + 3\lambda_{2,3} + \lambda_{2,6}} - \frac{3}{2\lambda_{2,1} + 2\lambda_{2,3} + \lambda_{2,6}} + \frac{2}{2\lambda_{2,1} + 3\lambda_{2,3} + \lambda_{2,6}}$$

Table 4-4: Exponential model failure rates for $subsystem_2$

Control $C_{2,j}$	Failure rate $\lambda_{2,j}$ $(week^{-1})$
$C_{2,1}$	0.075
$C_{2,2}$	0.075
$C_{2,3}$	0.08
$C_{2,4}$	0.08
$C_{2,5}$	0.08
$C_{2,6}$	0.05

Assuming that $subsystem_2$ exhibits the failure rates of the individual controls in Table 4-4, the corresponding performance is demonstrated in Figure 4-9. The chart depicts the different structures making up $subsystem_2$ in a sequence of a series arrangement, namely, the 1-out-of-2 structure, called $S_{2,1}$; the 2-out-of-3 structure, called $S_{2,2}$; and the one-item structure, called $S_{2,3}$, and the corresponding overall $subsystem_2$ reliability $R_2(t)$. Similar to the case in $subsystem_1$, all performance curves, including the mean time to failure $MTTF_2$ point, fall after the predetermined mission time and above the minimum reliability bound at this layer r_2. As a result, the reliability performance of security controls at this layer meets the requirement set for the maturity measure.

The numerical value of the reliability for the mission time of one week, $t = 1$,

$$R_2(1) = 0.93034$$

The mean time to failure,

$$MTTF_2 = 6.3058\, weeks$$

To evaluate the reliability when the mission time is set for one day, $\mathrm{t} = \frac{1}{7}$,

$$R_2\left(\frac{1}{7}\right) = 0.99239$$

And when the longer mission time is considered, i.e., 30 days, $t = \frac{30}{7}$

$$R_2\left(\frac{30}{7}\right) = 0.59404$$

Moreover, if we want to limit the reliability to a minimum of three nines, i.e., $R_2(t) = 0.999$, the corresponding mission time, as shown in in Figure 4-9, must be limited to $t = 0.018\,week \approx 0.13\,day = 3.12\,hours = 187\,minutes$.

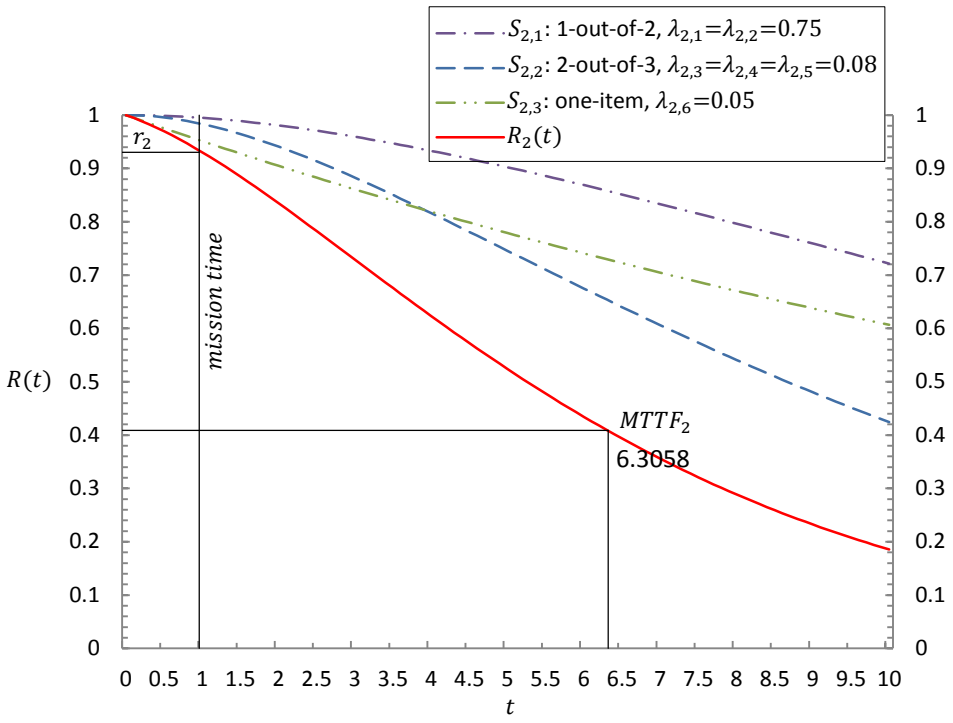

Figure 4-9: ISMM-based time-dependent reliability analysis for $subsystem_2$

***Subsystem*$_3$ analysis.** For the case of dissimilar redundancy in $subsystem_3$, we obtain

$$R_3(t) = \left[R_{3,1}(t) + R_{3,2}(t) - R_{3,1}(t)R_{3,2}(t)\right]\left[R_{3,3}(t) + R_{3,4}(t) - R_{3,3}(t)R_{3,4}(t)\right]\left[R_{3,5}(t) + R_{3,6}(t) - R_{3,5}(t)R_{3,6}(t)\right]R_{3,7}(t)R_{3,8}(t)R_{3,9}(t)R_{3,10}(t)$$

$$= \left[e^{-\lambda_{3,1}t} + e^{-\lambda_{3,2}t} - e^{-(\lambda_{3,1}+\lambda_{3,2})t}\right]\left[e^{-\lambda_{3,3}t} + e^{-\lambda_{3,4}t} - e^{-(\lambda_{3,3}+\lambda_{3,4})t}\right]\left[e^{-\lambda_{3,5}t} + e^{-\lambda_{3,6}t} - e^{-(\lambda_{3,5}+\lambda_{3,6})t}\right]e^{-\lambda_{3,7}t}e^{-\lambda_{3,8}t}e^{-\lambda_{3,9}t}e^{-\lambda_{3,10}t}$$

and when the identical redundancy is considered, we obtain

$$\begin{aligned} R_3(t) &= \left[2e^{-\lambda_{3,1}t} - e^{-2\lambda_{3,1}t}\right]\left[2e^{-\lambda_{3,3}t} - e^{-2\lambda_{3,3}t}\right]\left[2e^{-\lambda_{3,5}t} - e^{-2\lambda_{3,5}t}\right]e^{-\lambda_{3,7}t}e^{-\lambda_{3,8}t}e^{-\lambda_{3,9}t}e^{-\lambda_{3,10}t} \\ &= 8e^{-(\lambda_{3,1}+\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} - 4e^{-(\lambda_{3,1}+\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} \\ &- 4e^{-(\lambda_{3,1}+2\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} + 2e^{-(\lambda_{3,1}+2\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} \\ &- 4e^{-(2\lambda_{3,1}+\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} + 2e^{-(2\lambda_{3,1}+\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} \\ &+ 2e^{-(2\lambda_{3,1}+2\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} - e^{-(2\lambda_{3,1}+2\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10})t} \end{aligned}$$

Integrating the complete expression of $R_3(t)$ to obtain the corresponding $MTTF_3$ yields

$$\begin{aligned} MTTF_3 &= \frac{8}{\lambda_{3,1}+\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &- \frac{4}{\lambda_{3,1}+\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &- \frac{4}{\lambda_{3,1}+2\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &+ \frac{2}{\lambda_{3,1}+2\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &- \frac{4}{2\lambda_{3,1}+\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &+ \frac{2}{2\lambda_{3,1}+\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &+ \frac{2}{2\lambda_{3,1}+2\lambda_{3,3}+\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \\ &- \frac{1}{2\lambda_{3,1}+2\lambda_{3,3}+2\lambda_{3,5}+\lambda_{3,7}+\lambda_{3,8}+\lambda_{3,9}+\lambda_{3,10}} \end{aligned}$$

Table 4-5: Exponential model failure rates for $subsystem_3$

Control $C_{3,j}$	Failure rate $\lambda_{3,j}$ $(week^{-1})$
$C_{3,1}$	0.08
$C_{3,2}$	0.08
$C_{3,3}$	0.09
$C_{3,4}$	0.09
$C_{3,5}$	0.07
$C_{3,6}$	0.07
$C_{3,7}$	0.07
$C_{3,8}$	0.045
$C_{3,9}$	0.05
$C_{3,10}$	0.04

Assuming that $subsystem_3$ exhibits the failure rates of the individual controls in Table 4-5, the corresponding reliability performance of controls is depicted in Figure 4-10. Note that the first redundant configuration is called $S_{3,1}$, the second is $S_{3,2}$, and so on for the remaining structures making up the series arrangement individually. Although the reliabilities of individual structures fall after the predetermined mission time and above the minimum reliability bound, overall subsystem reliability crosses such boundaries, causing this subsystem to fail its maturity test.

The numerical value of the reliability for a mission time of one week, $t = 1$,

$$R_3(1) = 0.80016$$

The mean time to failure,

$$MTTF_3 = 3.5742$$

When the mission time is set for one day, $t = \frac{1}{7}$, we obtain

$$R_3\left(\frac{1}{7}\right) = 0.97076$$

And in the case of the longer mission time of 30 days, $t = \frac{30}{7}$,

$$R_3\left(\frac{30}{7}\right) = 0.31849$$

Also, if we want to limit the reliability to a minimum of $R_3(t) = 0.999$, the corresponding mission time, as shown in Figure 4-10, must be limited to $t = 0.005\ weeks \approx 0.035\ days = 0.84\ hours = 50\ minutes$.

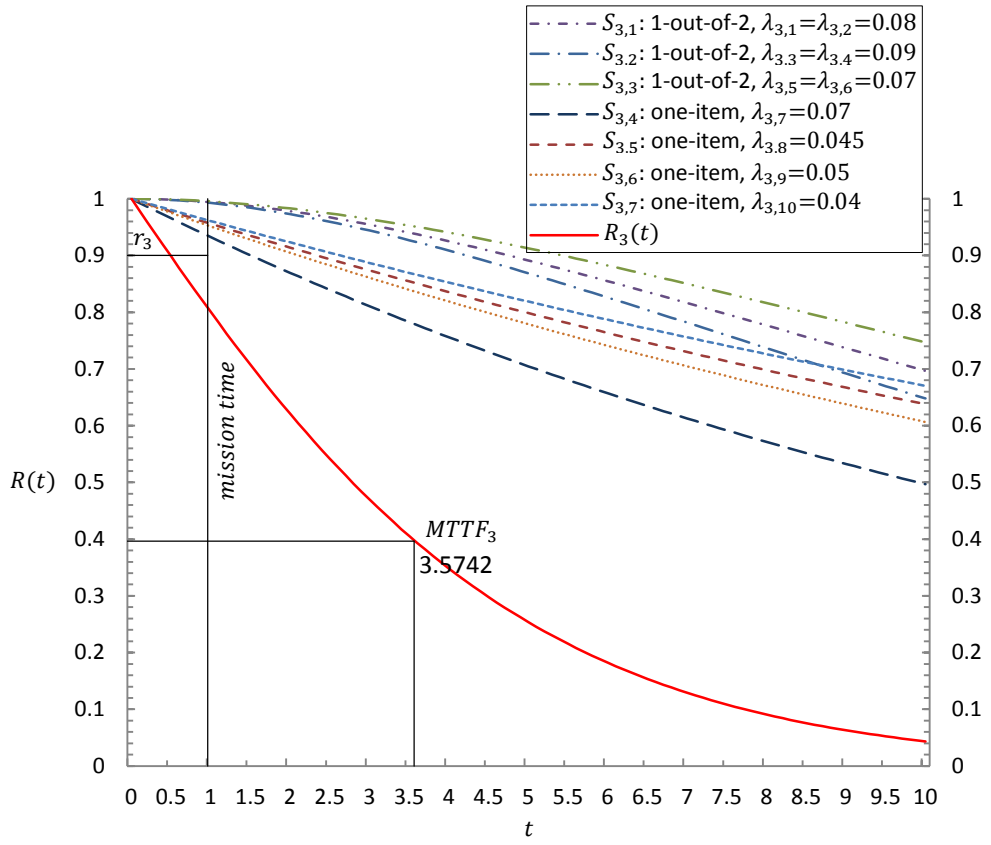

Figure 4-10: ISMM-based time-dependent reliability analysis for $subsystem_3$

***Subsystem*$_4$ analysis**. Recall that $subsystem_4$ is a pure series subsystem, i.e., no redundancy, leading to

$$R_4(t) = R_{4,1}(t)R_{4,2}(t)R_{4,3}(t)R_{4,4}(t) = e^{-\lambda_{4,1}t}e^{-\lambda_{4,2}t}e^{-\lambda_{4,3}t}e^{-\lambda_{4,4}t} = e^{-(\lambda_{4,1}+\lambda_{4,2}+\lambda_{4,3}+\lambda_{4,4})t}$$

Integrating the complete expression of $R_4(t)$ to obtain the corresponding $MTTF_4$ yields

$$MTTF_4 = \frac{1}{\lambda_{4,1} + \lambda_{4,2} + \lambda_{4,3} + \lambda_{4,4}}$$

Table 4-6: Exponential model failure rates for $subsystem_4$

Control $C_{4,j}$	Failure rate $\lambda_{4,j}$ $(week^{-1})$
$C_{4,1}$	0.11
$C_{4,2}$	0.08
$C_{4,3}$	0.05
$C_{4,4}$	0.07

Assuming that $subsystem_4$ exhibits the failure rates of the individual controls presented in Table 4-6, the corresponding performance is depicted in Figure 4-11. Observe that the overall reliability of the corresponding subsystem fails to meet the maturity requirement. The numerical value, however, of the reliability for a mission time of one year, $t = 1$,

$$R_4(1) = 0.73345$$

The mean time to failure,

$$MTTF_4 = 3.2258$$

If the mission time is set for one day, $t = \frac{1}{7}$,

$$R_4\left(\frac{1}{7}\right) = 0.95669$$

For the longer mission time of 30 days, $= \frac{30}{7}$,

$$R_4\left(\frac{30}{7}\right) = 0.26486$$

Also, if we want to limit the reliability to the minimum of three nines, i.e., $R_4(t) = 0.999$, the corresponding mission time must be limited to

$$t = \frac{-(\ln(R_4(t)))}{\lambda_4} = \frac{-(\ln 0.999)}{\lambda_4}, where\ \lambda_4 = \sum_{j=1}^{n_4=4} \lambda_{4,j} = 0.31\ week^{-1}$$

$$= 0.0032\ weeks \approx 0.0224\ days = 0.54\ hours = 32\ minutes$$

This equation is applicable here because of the property that the series arrangement of an exponential model leads to an exponential model as well. The result can be verified both analytically by calculating $R_4(0.0032)$ and graphically as in Figure 4-11.

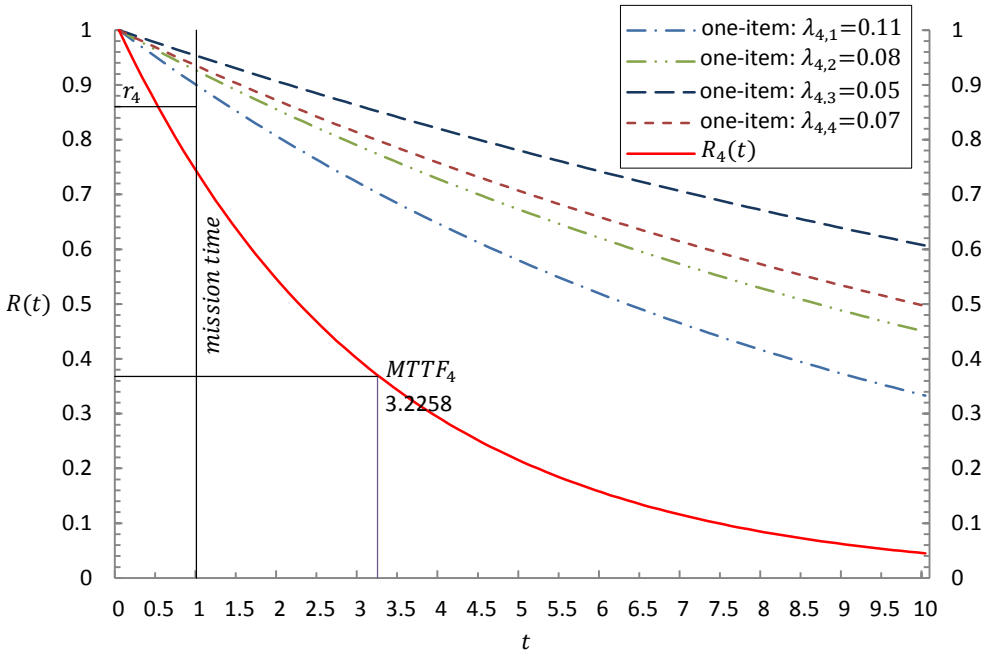

Figure 4-11: ISMM-based time-dependent reliability analysis for $subsystem_4$

***Subsystem*$_5$ analysis**. Similarly, when $subsystem_5$ is considered, we obtain

$$R_5(t) = R_{5,1}(t)R_{5,2}(t)R_{5,3}(t)R_{5,4}(t)R_{5,5}(t) = e^{-\lambda_{5,1}t}e^{-\lambda_{5,2}t}e^{-\lambda_{5,3}t}e^{-\lambda_{5,4}t}e^{-\lambda_{5,5}t}$$

$$= e^{-(\lambda_{5,1}+\lambda_{5,2}+\lambda_{5,3}+\lambda_{5,4}+\lambda_{5,5})t}$$

Integrating the complete expression of $R_5(t)$ to obtain the corresponding $MTTF_5$ yields

$$MTTF_5 = \frac{1}{\lambda_{5,1} + \lambda_{5,2} + \lambda_{5,3} + \lambda_{5,4} + \lambda_{5,5}}$$

Table 4-7: Exponential model failure rates for $subsystem_5$

Control $C_{5,j}$	Failure rate $\lambda_{5,j}$ $(week^{-1})$
$C_{5,1}$	0.07
$C_{5,2}$	0.1
$C_{5,3}$	0.08
$C_{5,4}$	0.085
$C_{5,5}$	0.09

Assuming that $subsystem_5$ exhibits the failure rates of the individual controls in Table 4-7, the corresponding performance of the pure series arrangement structure is shown in Figure 4-12, which clearly fails to meet the maturity test. The numerical value of the reliability, however, for the set mission time of one week, $t = 1$,

$$R_5(1) = 0.65377$$

The mean time to failure,

$$MTTF_5 = 2.3529$$

If the mission time is set for one day, $t = \frac{1}{7}$,

$$R_5\left(\frac{1}{7}\right) = 0.94109$$

For the longer mission time of 30 days, $t = \frac{30}{7}$,

$$R_5\left(\frac{30}{7}\right) = 0.16179$$

Also, if we want to limit the reliability to the minimum of three nines, i.e., $R_5(t) = 0.999$, the corresponding mission time must be limited to

$$t = \frac{-(\ln(R_5(t)))}{\lambda_5} = \frac{-(\ln 0.999)}{\lambda_5}, where\ \lambda_5 = \sum_{j=1}^{n_5=5} \lambda_{5,j} = 0.425\ week^{-1}$$

$$= 0.0024\ weeks \approx 0.0168\ days = 0.40\ hours = 24\ minutes$$

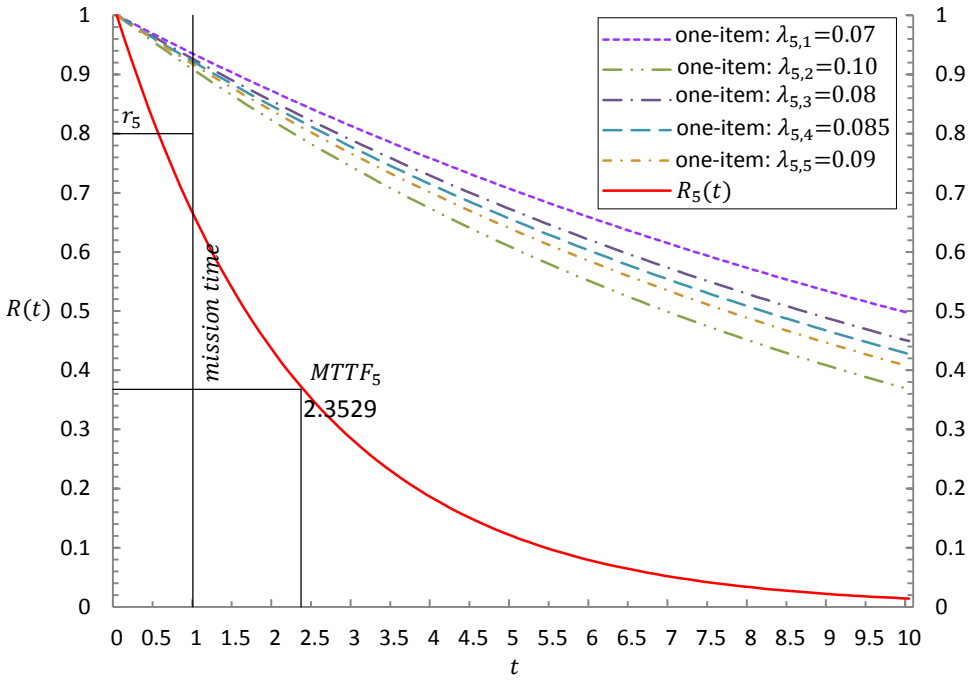

Figure 4-12: ISMM-based time-dependent reliability analysis for $subsystem_5$

***System*$_{ISMM}$ analysis**. The reliability of the overall system security $R_{ISMM}(t)$ is calculated as

$$R_{ISMM}(t) = \left[R_{1,1}(t) + R_{1,2}(t) - R_{1,1}(t)R_{1,2}(t)\right]\left[R_{1,3}(t) + R_{1,4}(t) - R_{1,3}(t)R_{1,4}(t)\right]$$
$$\left[R_{1,5}(t) + R_{1,6}(t) - R_{1,5}(t)R_{1,6}(t)\right]\left[R_{1,7}(t) + R_{1,8}(t) - R_{1,7}(t)R_{1,8}(t)\right]$$
$$\left[R_{2,1}(t) + R_{2,2}(t) - R_{2,1}(t)R_{2,2}(t)\right]\left[R_{2,3}(t)R_{2,4}(t) + R_{2,3}(t)R_{2,5}(t)\right.$$
$$\left. + R_{2,4}(t)R_{2,5}(t) - 2R_{2,3}(t)R_{2,4}(t)R_{2,5}(t)\right]R_{2,6}(t)\left[R_{3,1}(t) + R_{3,2}(t)\right.$$
$$\left. - R_{3,1}(t)R_{3,2}(t)\right]\left[R_{3,3}(t) + R_{3,4}(t) - R_{3,3}(t)R_{3,4}(t)\right]\left[R_{3,5}(t) + R_{3,6}(t)\right.$$
$$\left. - R_{3,5}(t)R_{3,6}(t)\right]R_{3,7}(t)R_{3,8}(t)R_{3,9}(t)R_{3,10}(t)$$
$$R_{4,1}(t)R_{4,2}(t)R_{4,3}(t)R_{4,4}(t)R_{5,1}(t)R_{5,2}(t)R_{5,3}(t)R_{5,4}(t)R_{5,5}(t)$$

Substituting the exponential model,

$$R_{ISMM}(t)$$
$$= \left[e^{-\lambda_{1,1}t} + e^{-\lambda_{1,2}t} - e^{-(\lambda_{1,1}+\lambda_{1,2})t}\right]\left[e^{-\lambda_{1,3}t} + e^{-\lambda_{1,4}t} - e^{-(\lambda_{1,3}+\lambda_{1,4})t}\right]\left[e^{-\lambda_{1,5}t} + e^{-\lambda_{1,6}t}\right.$$
$$\left. - e^{-(\lambda_{1,5}+\lambda_{1,6})t}\right]\left[e^{-\lambda_{1,7}t} + e^{-\lambda_{1,8}t} - e^{-(\lambda_{1,7}+\lambda_{1,8})t}\right]\left[e^{-\lambda_{2,1}t} + e^{-\lambda_{2,2}t} - e^{-(\lambda_{2,1}+\lambda_{2,2})t}\right]\left[e^{-(\lambda_{2,3}+\lambda_{2,4})t}\right.$$
$$\left. + e^{-(\lambda_{2,3}+\lambda_{2,5})t} + e^{-(\lambda_{2,4}+\lambda_{2,5})t} - 2e^{-(\lambda_{2,3}+\lambda_{2,4}+\lambda_{2,5})t}\right]e^{-\lambda_{2,6}t}\left[e^{-\lambda_{3,1}t} + e^{-\lambda_{3,2}t}\right.$$
$$\left. - e^{-(\lambda_{3,1}+\lambda_{3,2})t}\right]\left[e^{-\lambda_{3,3}t} + e^{-\lambda_{3,4}t} - e^{-(\lambda_{3,3}+\lambda_{3,4})t}\right]\left[e^{-\lambda_{3,5}t} + e^{-\lambda_{3,6}t}\right.$$
$$\left. - e^{-(\lambda_{3,5}+\lambda_{3,6})t}\right]e^{-\lambda_{3,7}t}e^{-\lambda_{3,8}t}e^{-\lambda_{3,9}t}e^{-\lambda_{3,10}t}e^{-\lambda_{4,1}t}e^{-\lambda_{4,2}t}e^{-\lambda_{4,3}t}e^{-\lambda_{4,4}t}e^{-\lambda_{5,1}t}e^{-\lambda_{5,2}t}e^{-\lambda_{5,3}t}e^{-\lambda_{5,4}t}e^{-\lambda_{5,5}t}$$

The numerical value of the reliability for the mission time of one week, $t = 1$,

$$R_{\text{ISMM}}(1) = 0.34720$$

The mean time to failure,

$$MTTF_{ISMM} = 0.9134$$

If the mission time is set for one day, $t = \frac{1}{7}$,

$$R_{\text{ISMM}}\left(\frac{1}{7}\right) = 0.86682$$

And for the longer mission time, 30 days, $t = \frac{30}{7}$

$$R_{\text{ISMM}}\left(\frac{30}{7}\right) = 0.00541$$

Also, if we want to limit the reliability to a minimum of some value, say $R_{\text{ISMM}}(t) = 0.999$, the corresponding mission time, as shown in Figure 4-13, must be limited to $t = 0.0012\ weeks \approx 0.0084\ days = 0.202\ hours = 12\ minutes$.

Observe that this shortened mission time figure (i.e., $t = 12\ minutes$ in this example) represents an important design parameter necessary to reach the system reliability goal of three nines, without any changes in system structures (logical arrangements) or failure model. The alternative approach to reaching this target reliability is to increase the reliability of individual controls (by means of more economical investment), thus decreasing their failure probabilities while sustaining current mission time (i.e., $t = 1\ week$).

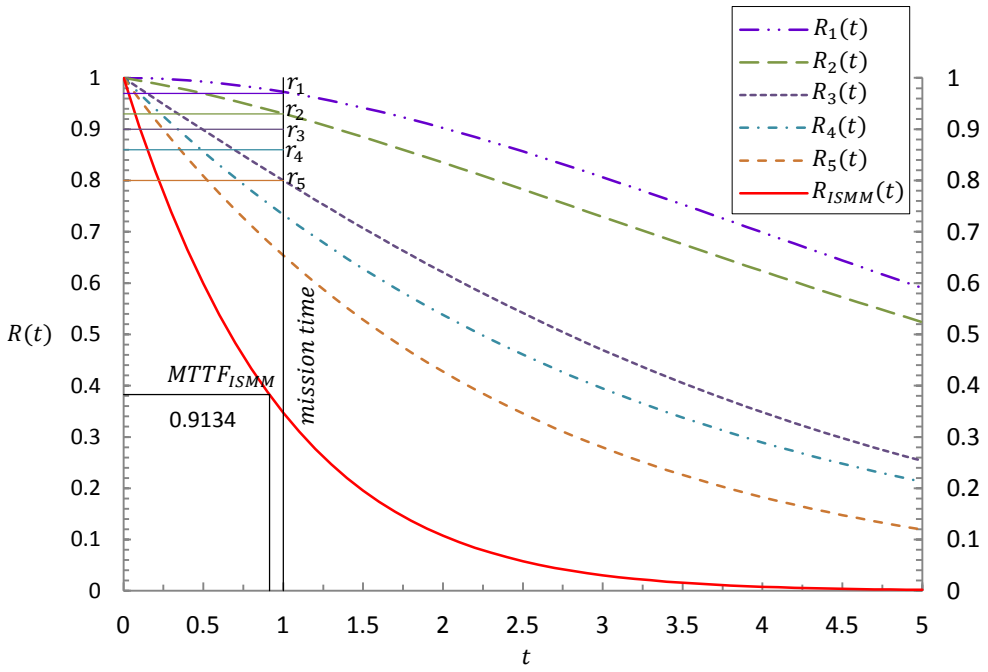

Figure 4-13: ISMM-based time-dependent reliability analysis for $system_{ISMM}$

Another important remark is the realization that the increase in reliability function caused by redundancy is very important for short missions, that is, when $t \ll MTTF$, as stated in [115]. For instance, the redundancy in R_1 and R_2 led to meeting their minimum maturity bounds r_1 and r_2, respectively, before mission time $t = 1$. Further, the fast exponential decay in R_4 and R_5 is due to their arrangement being a pure series. The redundancy, however, in R_3 was surpassed by its remaining number of series one-item controls, which caused its exponential decay to go faster than R_1 and R_2 but still slower than R_4 and R_5.

Maturity analysis: The maturity measure can be found graphically, as presented in Figure 4-13, or analytically, as shown here. Recall the maturity function equations in (3-14), (3-15), and (3-16). The maturity test can be established using the acceptability function as

$$I\big(R_i(t) \geq threshold(R_i(t))\big) = I(R_i(t) \geq r_i), for\ all\ i = 1, \dots, 5; mission\ time\ t = 1,$$

leading to the following tests,

$$I(R_1(1) \geq r_1) = I(R_2(1) \geq r_2) = 1$$
$$I(R_3(1) \geq r_3) = I(R_4(1) \geq r_4) = I(R_5(1) \geq r_5) = 0$$

As a result,

$$\mathcal{M}_{ISMM} = \mathcal{M}_R = \left\{\max(m)\, s.t. \left\{\prod_{i=1}^{m} I\big(R_i(1) \geq threshold(R_i(1))\big)\right\} = 1\right\} = 2$$

Moreover, we can use the graphical representation in Figure 4-13 to find the maximum mission time that satisfies current maturity conditions, logical arrangements, and the failure model. This value can be found graphically by making the minimum decrease of the mission time point (sliding the vertical mission time line to the left) such that $R_i(t) \geq r_i \; for \; all \; i = 1, \ldots, 5$. This procedure leads to a point that intersects the mission time line with the reliability $R_i(t)$ curve and its minimum bound r_i line of at least one subsystem. Applying this procedure leads to the intersection point for $subsystem_4$ at $t = 0.48$. Thus, we conclude that the new mission time $\hat{t} = 0.48\, weeks \approx 3.36\, days = 80\, hours$, as shown in Figure 4-14, satisfies the condition

$$\mathcal{M}_{ISMM} = \mathcal{M}_R = \left\{\max(m)\, s.t. \left\{\prod_{i=1}^{m} I\big(R_i(\hat{t}) \geq threshold(R_i(\hat{t}))\big)\right\} = 1\right\} = 5$$

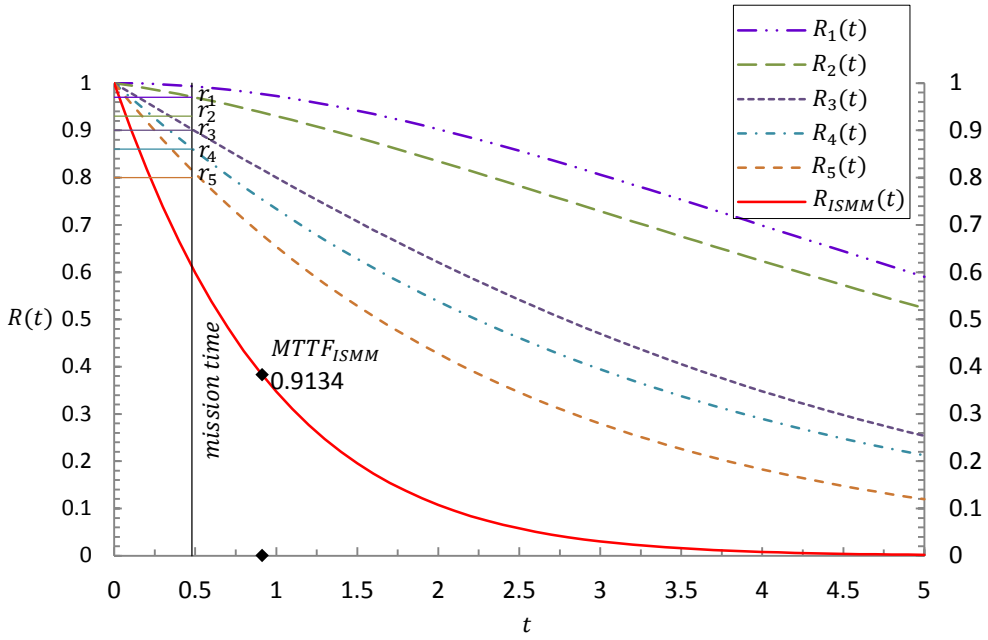

Figure 4-14: Redesigning mission time t to advance maturity

4.5 Summary

This chapter has shown how reliability analysis methods can be useful and extended to analyse security systems. Using the case study in Section 3.11, we have particularly demonstrated the extensions of the minimal path method, the minimal cut method, and reliability analysis based on both random events and random variables. We have also shown how to establish the maturity adequacy function and maturity analysis using the reliability measure. The analysis have shown the importance and use of logical arrangements of security controls, failure statistics, and designated mission times as design parameters to address the dependability of security systems.

Chapter 5
ISMM-based Multi-state System Evaluation Using the Universal Generating Function

5.1 Introduction

The issue of approaching the evaluation problem of a security system using methods extended from conventional reliability has been addressed earlier in Chapter 4. Such an extension allows one to particularly study performance of the security system using reliability and availability measures, and build maturity analysis accordingly. While these measures are useful to security studies as they reflect operational capabilities about the system of interest, a natural extension is addressing the evaluation problem of multistate security systems using multiple performance measures. The concept of MSS systems and its mathematical formulation, described in [116], [117], [124], [125], [126], and UGF method, described in [113], [118], [119], [128], [129], provide a suitable combination that can be extended into an ISMM-based modelling approach to address this research extension.

In this chapter, we show how to model and analyze a multistate security system using multiple performance measures. Observe that we consider only the case of MSS systems with the assumption of independent elements, or controls in this case. While the definitions presented here provide the necessary grounds for modelling dependent systems, the detailed application of the UGF method on MSS systems with dependent elements can be found in [116], [164]. Complicated solution methods, such as heuristics and approximation techniques explained in [102], [113], [116], [117], [125], are readily applicable to this work when exact solutions become difficult or intractable analytically.

The rest of this chapter is organized as follows. Section 5.2 introduces the definition and formulation of a multi-layer MSS system model based on the ISMM model. Section 5.3 demonstrates how the UGF method can be used to study various performance measures of the introduced multi-layer MSS system, not necessarily restricted to reliability and availability measures. Then, this extension is demonstrated analytically and numerically in Section 5.4 and Section 5.5, respectively. For mutually validating both the reliability-theoretic and MSS UGF-based methods, in addition to simplifying and unifying the analysis provided, however, the demonstration is based on reliability and maturity analysis, using the same case study presented in Section 3.11.

5.2 Multi-layer MSS Model

Extending MSS system representation into an ISMM-based modelling approach is motivated by the analysis tools and benefits that can be facilitated by the combined use of the MSS method and ISMM model and their properties; in particular, the established bounds of a security system using ISMM model, its precedence of layers, vectors and structures, control interrelationships, and associated quantifier functions using MSS-based performance measures. This combination leads to building what we call *Multi-layer MSS model*, or *MLMSS* for short. The resulting combination allows one to use MSS representation to model a security system when its controls and consequent subsystems can take an arbitrary finite number of different states of performance rates, ranging from perfect functioning on one side to complete failure on the other side.

Recall that ISMM-based system $\boldsymbol{C}$ *or* $\boldsymbol{C_{ISMM}}$ consists basically of five ordered layers, denoted by $\boldsymbol{C_i}, i = 1, \dots, 5$, with a finite set of controls $C_{i,j}$ at each layer, each of which may, in turn, consist of another smaller-level abstraction of set of components. In an analogy to MSS definitions introduced in [124], [125], [126], assume that control $C_{i,j}$ can have $K_{i,j}$ different, mutually exclusive states that encode the discrete scale of a particular performance measure, from complete failure to perfect functioning. At control level, this can be represented by the set

$$\boldsymbol{g_{i,j}} = \left\{g_{i,j,1}, g_{i,j,2}, \dots, g_{i,j,K_{i,j}}\right\}, \tag{5-1}$$

with related probability of performance for each state,

$$\boldsymbol{p_{i,j}} = \left\{p_{i,j,1},\ p_{i,j,2}, \dots, p_{i,j,K_{i,j}}\right\}, \tag{5-2}$$

where $g_{i,j,l}$ is the performance rate with associated probability $p_{i,j,l}$ for control i at layer j in the state l, $l \in \{1,2, \dots, K_{i,j}\}$, and $K_{i,j}$ is the number of different performance rates for control $C_{i,j}$. Consequently, in the stochastic domain, the performance rate at any instance can be represented by $G_{i,j}(t);\ 0 \le t \le T$, i.e., a random variable taking values from $\boldsymbol{g_{i,j}}$, so $G_{i,j}(t) \in \boldsymbol{g_{i,j}}$ with a probability $P_{i,j}(t) \in \boldsymbol{p_{i,j}}$, where $\sum_{l=1}^{K_{i,j}} p_{i,j,l}(t) = 1$, which represents a stochastic process for the time interval $[0, T]$ of the MSS operation time.

At subsystem level, layer$_i$ or $\boldsymbol{C_i}$ performance space can be represented by the set

$$\boldsymbol{g_i} = \left\{g_{i,1}, g_{i,2}, \dots, g_{i,K_i}\right\}, \tag{5-3}$$

with related probability of performance for each state,

$$\boldsymbol{p_i} = \{p_{i,1},\ p_{i,2}, \dots, p_{i,K_i}\}, \tag{5-4}$$

where $g_{i,l}$ is the performance rate with associated probability $p_{i,l}$ for $subsystem_i$ (or $\boldsymbol{C_i}$) in the state $l, l \in \{1,2, \dots, K_i\}$. Thus $\boldsymbol{C_i}$ takes K_i different states defined by the states of its individual controls $C_{i,j}, where\ K_i = \prod_{j=1}^{n_i} K_{i,j}\ , i \in \{1, \dots ,5\}, j \in \{1, \dots, n_i\}$, and n_i is the total number of controls at $\boldsymbol{C_i}$. Also, for the stochastic representation let the performance rate at any instance for $\boldsymbol{C_i}$ be represented by $G_i(t);\ 0 \le t \le T$, i.e., a random variable taking values from $\boldsymbol{g_i}$, so $G_i(t) \in \boldsymbol{g_i}$ with a probability $P_i(t) \in \boldsymbol{p_i}$ where $\sum_{l=1}^{K_i} p_{i,l}(t) = 1$.

As a result, the performance space at system level $\boldsymbol{C}$ is defined by the set of its individual subsystems' performances, denoted by the set

$$\boldsymbol{g} = \{g_1, g_2, \dots, g_K\}, \tag{5-5}$$

with related probability of performance for each state,

$$\boldsymbol{p} = \{p_1,\ p_2, \dots, p_K\}, \tag{5-6}$$

where g_l is the performance rate with associated probability p_l for system $\boldsymbol{C}$ in the state $l, l \in \{1,2, \dots, K\}$. Note that K in this case represents the different states determined by the states of its individual subsystems $\boldsymbol{C_i}$, so, $K = \prod_{i=1}^{5} K_i$. Consequently, the MLMSS performance rate $G(t)$ of an ISMM-based system at any instance t is a random variable too, taking values from $\boldsymbol{g}$, so $G(t) \in \boldsymbol{g}$ with a probability $P(t) \in \boldsymbol{p}$ where $\sum_{l=1}^{K} p_l(t) = 1$.

Let $\boldsymbol{L_i^{n_i}} = \{g_{i,1,1}, g_{i,1,2}, \dots, g_{i,1,K_{i,1}}\} \times \{g_{i,2,1}, g_{i,2,2}, \dots, g_{i,2,K_{i,2}}\} \times \dots \times \{g_{i,n_i,1}, g_{i,n_i,2}, \dots, g_{i,n_i,K_{i,n_i}}\}$, which is the space for all possible combinations of performance rates for all controls at layer $\boldsymbol{C_i}$. Also, let $\boldsymbol{M_i} = \{g_{i,1}, g_{i,2}, \dots, g_{i,K_i}\}$, which is the space of all possible values of performance rates for layer $\boldsymbol{C_i}$. Then we can define the structure function

$$\emptyset\left(G_{i,1}(t), G_{i,2}(t), \dots, G_{i,n_i}(t)\right) : \boldsymbol{L_i^{n_i}} \to \boldsymbol{M_i} \tag{5-7}$$

which performs a mapping function to a higher order space, that is, mapping the space of the controls' performance rates into the space of the subsystem's performance rates. As a result, the subsystem output performance distribution (OPD_i) can be defined by the two finite vectors $\boldsymbol{g_i}$ and $\boldsymbol{p_i}$ and the

system structure function $\emptyset\left(G_{i,1}(t), G_{i,2}(t), \dots, G_{i,n_i}(t)\right)$, where $\boldsymbol{p_i} = \{p_{i,l}(t)\} = \mathrm{P}\{G_i(t) = g_{i,l}\}; l \in \{1, \dots, K_i\}$.

Similarly for system level representation, let $\boldsymbol{L^n} = \{g_{1,1}, g_{1,2}, \dots, g_{1,K_1}\} \times \{g_{2,1}, g_{2,2}, \dots, g_{2,K_2}\} \times \dots \times \{g_{n,1}, g_{n,2}, \dots, g_{n,K_n}\}$, where $n = 5$, which basically represents the space of all possible combinations of performance rates for all system $\boldsymbol{C}$ layers. Also, let $\boldsymbol{M} = \{g_1, g_2, \dots, g_K\}$, which is the space of all possible values of the performance rates of system $\boldsymbol{C}$. Then we can write the structure function

$$\emptyset\big(G_1(t), G_2(t), \dots, G_n(t)\big): \boldsymbol{L^n = L^5 \rightarrow M} \tag{5-8}$$

which performs a mapping function of the space of the subsystems' performance rates into the space of the entire system's performance rates. As a result, the MLMSS output performance distribution (ODP) can be defined by the two finite vectors $\boldsymbol{g}$ and $\boldsymbol{p}$ and the system structure function $\emptyset\big(G_1(t), G_2(t), \dots, G_n(t)\big)$, where $\boldsymbol{p} = \{p_l(t)\} = \mathrm{P}\{G(t) = g_l\}; l \in \{1, \dots, K\}$.

5.3 Universal Generating Function in Analysis of Multi-layer MSS System

Following the introduction of the MLMSS method using the ISMM model, we introduce the mathematical representation necessary for extending the UGF method so that both representation and analysis of MLMSS security systems can be achieved. We first demonstrate this extension at control $C_{i,j}$ level, then subsystem $\boldsymbol{C_i}$ level, and finally the consequent system $\boldsymbol{C}$ level representation.

Recall that MLMSS representation of control $C_{i,j}$ of $K_{i,j}$ different and mutually exclusive performance states is defined by the set $\boldsymbol{g_{i,j}} = \left\{g_{i,j,1}, g_{i,j,2}, \dots, g_{i,j,K_{i,j}}\right\}$ with related probability of performance for each state $\boldsymbol{p_{i,j}} = \left\{p_{i,j,1}, p_{i,j,2}, \dots, p_{i,j,K_{i,j}}\right\}$, where layer index $i = 1, \dots, 5$, control index $j = 1, \dots, n_i$, and performance state index $k_{i,j} = 1, \dots, K_{i,j}$. The UGF method employs the use of both z-transform of discrete random variables and composition operators, where the resulting function is called u-function. Let $X_{i,j}$ be a discrete random variable that represents the state of the performance of interest for control $C_{i,j}$; the corresponding z-transform function as defined in [116], [117], [129] can be extended as follows.

$$\psi_{X_{i,j}}(z) = E[z^{X_{i,j}}] = \sum_{k_{i,j}=1}^{K_{i,j}} p_{x_{i,j,k_{i,j}}} z^{g_{i,j,k_{i,j}}} \tag{5-9}$$

The coefficients of the terms represent the probabilistic value of some object or state encoded by the exponent of the terms for control $C_{i,j}$, which in turn, can take arbitrary real values in this case for all values of z. It is notable that the extended definition of *z*-transform here inherits the essential properties of generating functions explained earlier in Section 2.3.2. Particularly, to find the expectation of a random variable, say $X_{i,j}$, that represents the performance state for control $C_{i,j}$, we write

$$\psi'_{X_{i,j}}(z)\Big|_{z=1} = \left[\sum_{k_{i,j}=1}^{K_{i,j}} g_{i,j,k_{i,j}} p_{x_{i,j,k_{i,j}}} z^{g_{i,j,k_{i,j}}-1}\right]_{z=1} = E\left[X_{i,j}\right] \tag{5-10}$$

And the *z*-transform of the summation of independent discrete random variables, say the set of $X'_{i,j}s, i = 1, \dots, 5$, and $j = 1, \dots, n_i$, representing the performance states of layer$_i$, or $\boldsymbol{C_i}$ subsystem, is the product of individual *z*-transforms of the random variables as follows.

$$\psi_{\sum_{j=1}^{n_i} X_{i,j}}(z) = \prod_{j=1}^{n_i} \psi_{X_{i,j}}(z) \tag{5-11}$$

Furthermore, *u*-function is extended to represent ISMM-based complex MSS systems where more composition operators, other than the addition of exponents in polynomials, can be defined to accommodate a wider range of performance measures and interactions among controls. This function can similarly represent a control, subsystem, or system as a polynomial $U(z)$ of a group of smaller components using simple algebraic operations over their individual *u*-functions $u(z)'s$. At the control $C_{i,j}$ level, individual $u_{i,j}(z)$ takes the following form

$$u_{i,j}(z) = \sum_{k_{i,j}=1}^{K_{i,j}} p_{i,j,k_{i,j}} z^{g_{i,j,k_{i,j}}} \tag{5-12}$$

The exponent of the terms encodes a state or object of interest for control $C_{i,j}$ with associated probabilities encoded by the coefficient of the terms. For two controls, say $C_{1,1}, C_{1,2} \in \boldsymbol{C_1}$, the corresponding $U(z)$ function takes the following form

$$
\begin{aligned}
U(z) &= \Omega_{\omega}\left(u_{1,1}(z), u_{1,2}(z)\right) \\
&= \Omega_{\omega}\left[\sum_{k_{1,1}=1}^{K_{1,1}} p_{1,1,k_{1,1}}\, z^{g_{1,1,k_{1,1}}}, \sum_{k_{1,2}=1}^{K_{1,2}} p_{1,2,k_{1,2}}\, z^{g_{1,2,k_{1,2}}}\right] \\
&= \sum_{k_{1,1}=1}^{K_{1,1}} \sum_{k_{1,2}=1}^{K_{1,2}} \left(p_{1,1,k_{1,1}} p_{1,2,k_{1,2}}\, z^{\omega\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right)}\right)
\end{aligned}
\tag{5-13}
$$

Note that $\omega(.)$ function represents the composition operator that reflects the MLMSS performance measure of interest and is strictly defined by the corresponding relationship between these two controls. For instance, the $\omega(.)$ function when used for binary reliability of ISMM-based system can be defined to equal the maximum of the control states when connected in parallel and the minimum when connected in series, denoted by $\omega_p(.)$ and $\omega_s(.)$, respectively. Then, the reliability of the two controls is found by computing the expected value of the variable, say arbitrarily $X_{1,3}$, which has the p.m.f. represented by $U(z)$, i.e., $U'(z)|_{z=1} = E\left[X_{1,3}\right]$. Thus $\omega(.)$ in this scenario can be defined as follows.

$$
\omega_p\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) = max\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) \tag{5-14}
$$

and

$$
\omega_s\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) = min\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) \tag{5-15}
$$

where $k_{i,j} \in \{1,2\}$, $g_{i,j,k_{i,j}} \in \{0,1\}$, and the *u*-function of individual controls takes the following form

$$
u_{i,j}(z) = p_{i,j,1}\, z^0 + (1 - p_{i,j,1})\, z^1 \tag{5-16}
$$

To address the *k*-out-of-*n* binary structure, the reliability can be directly computed in the case of identical and independent controls using the binomial distribution equation in (2-33). However, when controls are not identical, the probabilities of the possible realizations of the structure where the number of functioning components is at least *k* must be summed up using the *k*-out-of-*n* algorithm presented earlier in Section 2.3.2.

At subsystem C_i level, recall that MLMSS representation is defined by $g_i = \{g_{i,1}, g_{i,2}, \dots, g_{i,K_i}\}$ with related probability of performance for each state $p_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,K_i}\}$. The corresponding $U_i(z)$ function for subsystem C_i is written in the following general form:

$$
\begin{aligned}
U_i(z) &= \Omega_\omega\left(u_{i,1}(z), u_{i,2}(z), \dots, u_{i,n_i}(z)\right) \\
&= \Omega_\omega\left[\sum_{k_{i,1}=1}^{K_{i,1}} p_{i,1,k_{i,1}}\, z^{g_{i,1,k_{i,1}}}, \sum_{k_{i,2}=1}^{K_{i,2}} p_{i,2,k_{i,2}}\, z^{g_{i,2,k_{i,2}}}, \dots, \sum_{k_{i,n_i}=1}^{K_{i,n_i}} p_{i,n_i,k_{i,n_i}}\, z^{g_{i,n_i,k_{i,n_i}}}\right] \\
&= \sum_{k_{i,1}=1}^{K_{i,1}} \sum_{k_{i,2}=1}^{K_{i,2}} \dots \sum_{k_{i,n_i}=1}^{K_{i,n_i}} \left(p_{i,1,k_{i,1}}\, p_{i,2,k_{i,2}} \dots p_{i,n_i,k_{i,n_i}}\, z^{\omega\left(g_{i,1,k_{i,1}}, g_{i,2,k_{i,2}}, \dots, g_{i,n_i,k_{i,n_i}}\right)}\right)
\end{aligned}
\tag{5-17}
$$

When $U_i(z)$ represents the p.m.f. of the *r.v.* X_i for subsystem $\boldsymbol{C_i}$ or layer$_i$, we obtain

$$
U'_{x_i}(z)\big|_{z=1} = \frac{d}{dz} U_{x_i}(1) = E[X_i] \tag{5-18}
$$

At system $\boldsymbol{C}$ level, recall that MLMSS representation is defined by $\boldsymbol{g} = \{g_1, g_2, \dots, g_k\}$ with related probability of performance for each state $\boldsymbol{p} = \{p_1, p_2, \dots, p_k\}$. The corresponding $U(z)$ function takes the following general form:

$$
\begin{aligned}
U(z) &= \Omega_\omega\left(u_1(z), u_2(z), \dots, u_n(z)\right) \\
&= \Omega_\omega\left[\sum_{k_1=1}^{K_1} p_{1,k_1}\, z^{g_{1,k_1}}, \sum_{k_2=1}^{K_2} p_{2,k_2}\, z^{g_{2,k_2}}, \dots, \sum_{k_n=1}^{K_n} p_{n,k_n}\, z^{g_{n,k_n}}\right] \\
&= \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \dots \sum_{k_n=1}^{K_n} \left(p_{1,k_1}\, p_{2,k_2} \dots p_{n,k_n}\, z^{\omega\left(g_{1,k_1}, g_{2,k_2}, \dots, g_{n,k_n}\right)}\right)
\end{aligned}
\tag{5-19}
$$

where $n = 5$. Similarly, when $U(z)$ represents the p.m.f. of the *r.v.* X for system level $\boldsymbol{C}$, we obtain

$$
U'_x(z)|_{z=1} = \frac{d}{dz} U_x(1) = E[X] \tag{5-20}
$$

It is notable that the reduction techniques to minimise computational cost used in the original definitions and representations of UGF are applicable in this extension. Particularly, 1) collecting like terms, and 2) recursive procedures when enumerating controls states. So, if a *u*-function, perhaps for a particular control $C_{i,j}$ representing the distribution of $r.v.X_{i,j}$, contains the terms $p_{i,j,a}\, z^{g_{i,j,a}}$ and $p_{i,j,b}\, z^{g_{i,j,b}}$ where $g_{i,j,a} = g_{i,j,b}$, the two terms, using the essential property of regular polynomials, can be combined into one term as follows.

$$
\left(p_{i,j,a} + p_{i,j,b}\right) z^{g_{i,j,a}} \tag{5-21}
$$

Moreover, the *u*-function of higher-level abstraction, e.g., system, or subsystem, can be obtained recursively, with no sense to the order, using the *u*-functions of its constituting subordinate objects. Fortunately, such recursive determination of the *u*-functions is already enabled by the associative and commutative properties of the composition operator based on the definition in Section 3.9. Recall that ISMM quantifier functions by definition requires these properties to hold in the first place. This leads to the following relationship to hold, for subsystem $\boldsymbol{C_i}$ as an example,

$$\begin{aligned}&\Omega_\omega\left(u_{i,1}(z),\dots,u_{i,l}(z),u_{i,l+1}(z),\dots,u_{i,n_i}(z)\right)\\&\quad=\Omega_\omega\left\{\Omega_\omega\left(u_{i,1}(z),\dots,u_{i,l}(z)\right),\ \Omega_\omega\left(u_{i,l+1}(z),\dots,u_{i,n_i}(z)\right)\right\}\end{aligned}\tag{5-22}$$

and

$$\begin{aligned}&\Omega_\omega\left(u_{i,1}(z),\dots,u_{i,l}(z),u_{i,l+1}(z),\dots,u_{i,n_i}(z)\right)\\&\quad=\Omega_\omega\left(u_{i,1}(z),\dots,u_{i,l+1}(z),u_{i,l}(z),\dots,u_{i,n_i}(z)\right)\end{aligned}\tag{5-23}$$

5.4 Analytical Example

Similar to the reliability-theoretic evaluation in Chapter 4, we perform reliability and maturity analysis, presenting both analytical and numerical solutions to the case study in Section 3.11. The analysis is based on the same assumption of binary states of security controls, although these methods address multi-state systems. Observe that while UGF might not be the most effective method for analyzing binary systems in particular, due to the advances in their theory, solution procedures are the same when considering reliability for fixed mission times, time-dependent reliability, and steady-state availability; and the overall approach remains universal, covering the wide range of system types [116].

At control level, control $C_{i,j}$ with $K_{i,j}$ different and mutually exclusive performance states is defined by the set $\boldsymbol{g_{i,j}}=\{g_{i,j,1},g_{i,j,2}\}$ and related probability of performance for each state $\boldsymbol{p_{i,j}}=\{p_{i,j,1},\ p_{i,j,2}\}$. For binary systems, we represent the *u*-function of individual control $C_{i,j}$ by

$$u_{i,j}(z)=\sum_{k_{i,j}=1}^{K_{i,j}}p_{i,j,k_{i,j}}z^{g_{i,j,k_{i,j}}}=p_{i,j,1}\,z^0+\left(1-p_{i,j,1}\right)z^1, for\ all\ i\in\{1,\dots,5\},j$$
$$\in\{1,\dots,n_i\},k_{i,j}\in\{1,2\},g_{i,j,1}=0,g_{i,j,2}=1$$

with the $\omega(.)$ function in the case of a parallel arrangement,

$$\omega_p\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) = \max(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}),$$

and in the case of a series arrangement,

$$\omega_s\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right) = min\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right).$$

Mapping the case study in in Section 3.11 into the ISMM context using the UGF method leads to building the system structure depicted in Figure 5-1.

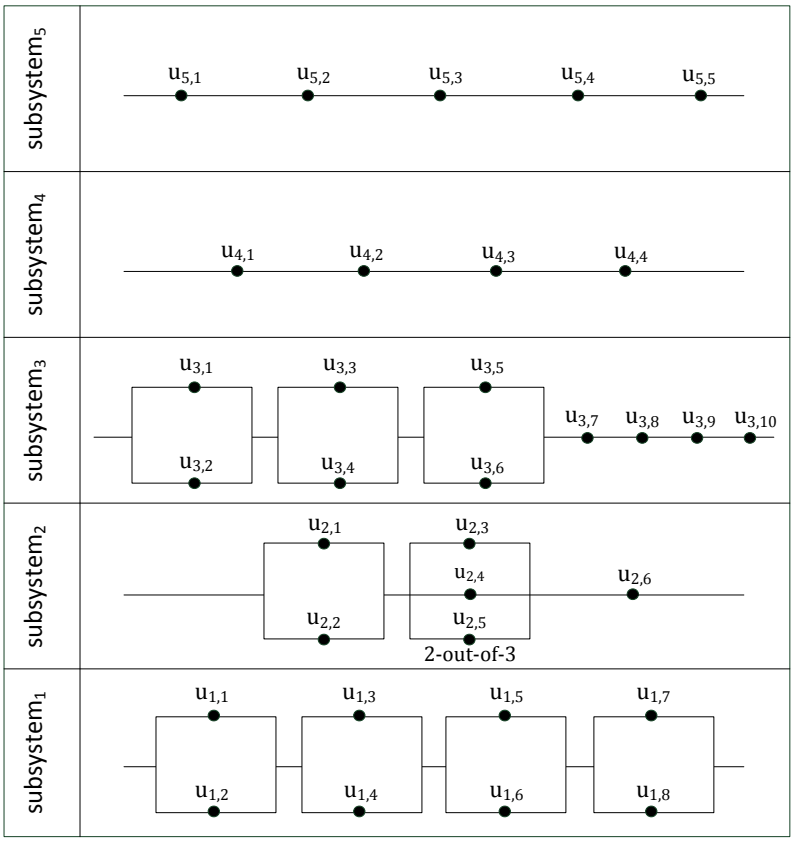

Figure 5-1: Mapping security controls on ISMM model using UGF method

***Subsystem*$_1$ formulation**. The corresponding *u*-function of $Subsystem_1$ is formulated as follows.

$$U_1(z) = \Omega_\omega\left(u_{1,1}(z), u_{1,2}(z), u_{1,3}(z), u_{1,4}(z), u_{1,5}(z), u_{1,6}(z), u_{1,7}(z), u_{1,8}(z)\right)$$

$$= \Omega_\omega[\sum_{k_{1,1}=1}^{2} p_{1,1,k_{1,1}}\, z^{g_{1,1,k_{1,1}}}, \sum_{k_{1,2}=1}^{2} p_{1,2,k_{1,2}}\, z^{g_{1,2,k_{1,2}}}, \sum_{k_{1,3}=1}^{2} p_{1,3,k_{1,3}}\, z^{g_{1,3,k_{1,3}}},$$

$$\sum_{k_{1,4}=1}^{2} p_{1,4,k_{1,4}}\, z^{g_{1,4,k_{1,4}}}, \sum_{k_{1,5}=1}^{2} p_{1,5,k_{1,5}}\ , \sum_{k_{1,6}=1}^{2} p_{1,6,k_{1,6}}\, z^{g_{1,6,k_{1,6}}}$$

$$\sum_{k_{1,7}=1}^{2} p_{1,7,k_{1,7}}\, z^{g_{1,7,k_{1,7}}}, \sum_{k_{1,8}=1}^{2} p_{1,8,k_{1,8}}\, z^{g_{1,8,k_{i,8}}}]$$

The next step is to solve this equation. The direct evaluation approach of the function $U_1(z)$ requires $2^{n_1} = 2^8 = 256$ evaluations to solve the model. However, employing the recursive feature of *u*-function reduces such cost. To do so, we evaluate and collect like terms of intermediate results recursively, as depicted in Figure 5-2, according to the following steps.

1. $U_{1,9}(z) = \Omega_{\omega_p}\left(u_{1,1}(z), u_{1,2}(z)\right)$
2. $U_{1,10}(z) = \Omega_{\omega_p}\left(u_{1,3}(z), u_{1,4}(z)\right)$
3. $U_{1,11}(z) = \Omega_{\omega_p}\left(u_{1,5}(z), u_{1,6}(z)\right)$
4. $U_{1,12}(z) = \Omega_{\omega_p}\left(u_{1,7}(z), u_{1,8}(z)\right)$
5. $U_{1,13}(z) = \Omega_{\omega_s}\left(u_{1,9}(z), u_{1,10}(z)\right)$
6. $U_{1,14}(z) = \Omega_{\omega_s}\left(u_{1,13}(z), u_{1,11}(z)\right)$
7. $U_1(z) = \Omega_{\omega_s}\left(u_{1,14}(z), u_{1,12}(z)\right)$

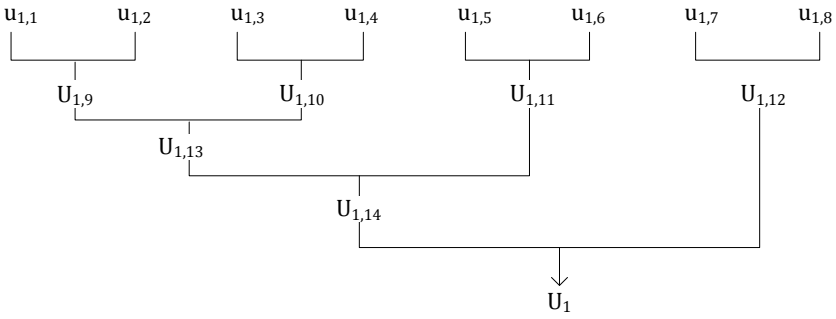

Figure 5-2: Order of evaluation for the function $U_1(z)$

These steps are further expanded as follows.

$$U_{1,9}(z) = \Omega_{\omega_p}\left(u_{1,1}(z), u_{1,2}(z)\right) = \Omega_{\omega_p}\left[\sum_{k_{1,1}=1}^{2} p_{1,1,k_{1,1}}\, z^{g_{1,1,k_{1,1}}}, \sum_{k_{1,2}=1}^{2} p_{1,2,k_{1,2}}\, z^{g_{1,2,k_{1,2}}}\right]$$

$$= \sum_{k_{1,1}=1}^{2}\sum_{k_{1,2}=1}^{2} p_{1,1,k_{1,1}} p_{1,2,k_{1,2}}\; z^{\max\left(g_{1,1,k_{1,1}}, g_{1,2,k_{1,2}}\right)}$$

$$= p_{1,1,1}p_{1,2,1}z^{\max(g_{1,1,1},g_{1,2,1})} + p_{1,1,1}p_{1,2,2}z^{\max(g_{1,1,1},g_{1,2,2})} + p_{1,1,2}p_{1,2,1}z^{\max(g_{1,1,2},g_{1,2,1})}$$
$$+ p_{1,1,2}p_{1,2,2}z^{\max(g_{1,1,2},g_{1,2,2})}$$

$$= p_{1,1,1}p_{1,2,1}z^{\max(0,0)} + p_{1,1,1}p_{1,2,2}z^{\max(0,1)} + p_{1,1,2}p_{1,2,1}z^{\max(1,0)} + p_{1,1,2}p_{1,2,2}z^{\max(1,1)}$$

$$= p_{1,1,1}p_{1,2,1}z^{0} + \left(p_{1,1,1}p_{1,2,2} + p_{1,1,2}p_{1,2,1} + p_{1,1,2}p_{1,2,2}\right)z^{1}$$

$$= p_{1,1,1}p_{1,2,1}z^{0} + (p_{1,1,2} + p_{1,2,2} - p_{1,1,2}p_{1,2,2})z^{1}$$

$$= p_{1,1,1}p_{1,2,1}z^{0} + \left(1 - p_{1,1,1}p_{1,2,1}\right)z^{1}.$$

Similarly,

$$\mathrm{U}_{1,10}(\mathrm{z}) = \Omega_{\omega_\mathrm{p}}\left(\mathrm{u}_{1,3}(\mathrm{z}), \mathrm{u}_{1,4}(\mathrm{z})\right) = \mathrm{p}_{1,3,1}\mathrm{p}_{1,4,1}\mathrm{z}^{0} + (1 - \mathrm{p}_{1,3,1}\mathrm{p}_{1,4,1})\mathrm{z}^{1}.$$

$$\mathrm{U}_{1,11}(\mathrm{z}) = \Omega_{\omega_\mathrm{p}}\left(\mathrm{u}_{1,5}(\mathrm{z}), \mathrm{u}_{1,6}(\mathrm{z})\right) = \mathrm{p}_{1,5,1}\mathrm{p}_{1,6,1}\mathrm{z}^{0} + (1 - \mathrm{p}_{1,5,1}\mathrm{p}_{1,6,1})\mathrm{z}^{1}.$$

$$\mathrm{U}_{1,12}(\mathrm{z}) = \Omega_{\omega_\mathrm{p}}\left(\mathrm{u}_{1,7}(\mathrm{z}), \mathrm{u}_{1,8}(\mathrm{z})\right) = \mathrm{p}_{1,7,1}\mathrm{p}_{1,8,1}\mathrm{z}^{0} + (1 - \mathrm{p}_{1,7,1}\mathrm{p}_{1,8,1})\mathrm{z}^{1}.$$

After that we evaluate the first two functions $U_{1,9}(z)$ and $U_{1,10}(z)$, collect like terms, and then evaluate the result with the next one, and so on until we have combined all the controls in $subsystem_1$, as follows.

$$
\begin{aligned}
U_{1,13}(z) &= \Omega_{\omega_s}\left(u_{1,9}(z), u_{1,10}(z)\right) \\
&= \Omega_{\omega_s}\left[p_{1,1,1}p_{1,2,1}z^0 + (1-p_{1,1,1}p_{1,2,1})z^1, p_{1,3,1}p_{1,4,1}z^0 + (1-p_{1,3,1}p_{1,4,1})z^1\right] \\
&= p_{1,1,1}p_{1,2,1}p_{1,3,1}p_{1,4,1}z^{\min(0,0)} + p_{1,1,1}p_{1,2,1}(1-p_{1,3,1}p_{1,4,1})z^{\min(0,1)} \\
&+ (1-p_{1,1,1}p_{1,2,1})p_{1,3,1}p_{1,4,1}z^{\min(1,0)} + (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})z^{\min(1,1)} \\
&= \left[p_{1,1,1}p_{1,2,1}p_{1,3,1}p_{1,4,1} + p_{1,1,1}p_{1,2,1}(1-p_{1,3,1}p_{1,4,1}) + (1-p_{1,1,1}p_{1,2,1})p_{1,3,1}p_{1,4,1}\right]z^0 \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})z^1 \\
&= \left[p_{1,1,1}p_{1,2,1} + (1-p_{1,1,1}p_{1,2,1})p_{1,3,1}p_{1,4,1}\right]z^0 + (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})z^1 \\
&= \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right]z^0 + (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})z^1.
\end{aligned}
$$

$$
\begin{aligned}
U_{1,14}(z) &= \Omega_{\omega_s}\left(u_{1,13}(z), u_{1,11}(z)\right) \\
&= \Omega_{\omega_s}\Big[\left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right]z^0 \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})z^1, p_{1,5,1}p_{1,6,1}z^0 + (1-p_{1,5,1}p_{1,6,1})z^1\Big] \\
&= \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right]p_{1,5,1}p_{1,6,1}z^{\min(0,0)} \\
&+ \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right](1-p_{1,5,1}p_{1,6,1})z^{\min(0,1)} \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})p_{1,5,1}p_{1,6,1}z^{\min(1,0)} \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})(1-p_{1,5,1}p_{1,6,1})z^{\min(1,1)} \\
&= \Big[\left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right]p_{1,5,1}p_{1,6,1} \\
&+ \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right](1-p_{1,5,1}p_{1,6,1}) \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})p_{1,5,1}p_{1,6,1}\Big]z^0 \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})(1-p_{1,5,1}p_{1,6,1})z^1 \\
&= \left[p_{1,5,1}p_{1,6,1} + \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})\right](1-p_{1,5,1}p_{1,6,1})\right]z^0 \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})(1-p_{1,5,1}p_{1,6,1})z^1 \\
&= \left[1-(1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})(1-p_{1,5,1}p_{1,6,1})\right]z^0 \\
&+ (1-p_{1,1,1}p_{1,2,1})(1-p_{1,3,1}p_{1,4,1})(1-p_{1,5,1}p_{1,6,1})z^1.
\end{aligned}
$$

Finally, the *u*-function for $subsystem_1$ can be found by

$$
\begin{aligned}
U_1(z) &= \Omega_{\omega_s}\left(u_{1,14}(z), u_{1,12}(z)\right) \\
&= \Omega_{\omega_s}\big(\big[1-\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\big]z^0 \\
&+ \left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)z^1, p_{1,7,1}p_{1,8,1}z^0 \\
&+ \left(1-p_{1,7,1}p_{1,8,1}\right)z^1\big) \\
&= \Big[\big[1-\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\big]p_{1,7,1}p_{1,8,1} \\
&+ \big[1-\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\big]\left(1-p_{1,7,1}p_{1,8,1}\right) \\
&+ \left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)p_{1,7,1}p_{1,8,1}\Big]z^0 \\
&+ \big[\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\left(1-p_{1,7,1}p_{1,8,1}\right)\big]z^1 \\
&= \big[p_{1,7,1}p_{1,8,1} \\
&+ \big[1-\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\big]\left(1-p_{1,7,1}p_{1,8,1}\right)\big]z^0 \\
&+ \big[\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\left(1-p_{1,7,1}p_{1,8,1}\right)\big]z^1 \\
&= \big[1-\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\left(1-p_{1,7,1}p_{1,8,1}\right)\big]z^0 \\
&+ \big[\left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\left(1-p_{1,7,1}p_{1,8,1}\right)\big]z^1.
\end{aligned}
$$

which can be rewritten in its compact form,

$$U_1(z) = p_{1,1}z^0 + p_{1,2}z^1$$

It is notable that this technique reduces the number of evaluations of the final term $U_1(z)$ to seven equations with four evaluations each, i.e., $7 \times 4 = 28$ evaluations. Recall that the mean performance of the p.m.f. $U_1(z)$ in this case represents the reliability of $subsystem_1$, i.e.,

$$R_1 = U'_{g_1}(z)\big|_{z=1} = \frac{d}{dz}U_{g_1}(1) = E[G_1]$$

This leads to

$$R_1 = p_{1,2} = \left(1-p_{1,1,1}p_{1,2,1}\right)\left(1-p_{1,3,1}p_{1,4,1}\right)\left(1-p_{1,5,1}p_{1,6,1}\right)\left(1-p_{1,7,1}p_{1,8,1}\right)$$

Observe that this is the same results found earlier in Section 4.4.2.

***Subsystem*$_2$ formulation**. The $U_2(z)$ function takes the following form

$$U_2(z) = \Omega_{\omega}\left(u_{2,1}(z), u_{2,2}(z), u_{2,3}(z), u_{2,4}(z), u_{2,5}(z), u_{2,6}(z)\right)$$

$$= \Omega_{\omega}[\sum_{k_{2,1}=1}^{2} p_{2,1,k_{2,1}} z^{g_{2,1,k_{2,1}}}, \sum_{k_{2,2}=1}^{2} p_{2,2,k_{2,2}} z^{g_{2,2,k_{2,2}}}, \sum_{k_{2,3}=1}^{2} p_{2,3,k_{2,3}} z^{g_{2,3,k_{2,3}}},$$

$$\sum_{k_{2,4}=1}^{2} p_{2,4,k_{2,4}} z^{g_{2,4,k_{2,4}}}, \sum_{k_{2,5}=1}^{2} p_{2,5,k_{2,5}} z^{g_{2,5,k_{2,5}}}, \sum_{k_{2,6}=1}^{2} p_{2,6,k_{2,6}} z^{g_{2,6,k_{2,6}}}].$$

Next, we solve $U_2(z)$. To avoid confusion of computation, we first build the recursive procedures with like terms collected for the 2-out-of-3 web servers structure using the *k*-out-of-*n* algorithm outlined in Section 2.3.2, followed by the procedures for the remaining controls, according to the order presented in Figure 5-3.

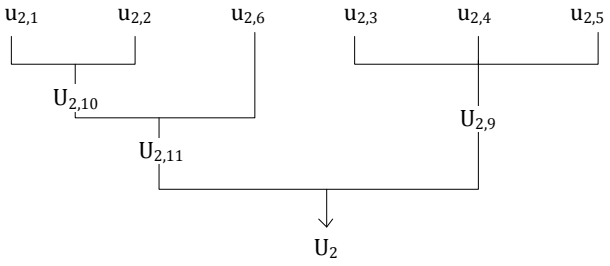

Figure 5-3: Order of evaluation for the function $U_2(z)$

Applying the *k*-out-of-*n* algorithm leads to the following steps:

1. Assign $R_{2\,|3} = 0$, denoting the reliability of the 2-out-of-3 structure
2. Then, find

$$U_{2,7}(z) = \Omega_{\omega_+}\left(u_{2,3}(z), u_{2,4}(z)\right) = \Omega_{\omega_+}\left(p_{2,3,1}z^0 + p_{2,3,2}z^1, p_{2,4,1}z^0 + p_{2,4,2}z^1\right)$$
$$= p_{2,3,1}p_{2,4,1}z^0 + (p_{2,3,1}p_{2,4,2} + p_{2,3,2}p_{2,4,1})z^1 + p_{2,3,2}p_{2,4,2}z^2$$

3. Remove $p_{2,3,2}p_{2,4,2}z^2$ and assign $R_{2|3} = p_{2,3,2}p_{2,4,2}$, leading to

$$U_{2,7}(z) = p_{2,3,1}p_{2,4,1}z^0 + (p_{2,3,1}p_{2,4,2} + p_{2,3,2}p_{2,4,1})z^1$$

4. Further, find

$$U_{2,8}(z) = \Omega_{\omega_+}\left(u_{2,7}(z), u_{2,5}(z)\right)$$

$$= \Omega_{\omega_+}\left(p_{2,3,1}p_{2,4,1}z^0 + \left(p_{2,3,1}p_{2,4,2} + p_{2,3,2}p_{2,4,1}\right)z^1, p_{2,5,1}z^0 + p_{2,5,2}z^1\right)$$
$$= p_{2,3,1}p_{2,4,1}p_{2,5,1}z^0$$
$$+ \left(p_{2,3,1}p_{2,4,1}p_{2,5,2} + p_{2,3,1}p_{2,4,2}p_{2,5,1} + p_{2,3,2}p_{2,4,1}p_{2,5,1}\right)z^1$$
$$+ \left(p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)z^2$$

5. Remove $\left(p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)$, and assign $R_{2|3} = p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}$
6. Write the resulted reliability of the 2-out-of-3 structure, i.e., $R_{2|3}$, into the *u*-function form, i.e., the binary state $U_{2,9}(z)$, as follows

$$U_{2,9}(z)$$
$$= \left(1 - \left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)\right)z^0$$
$$+ \left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)z^1$$

For the remaining controls of $subsystem_2$ structure,

$$U_{2,10}(z) = \Omega_{\omega_p}\left(u_{2,1}(z), u_{2,2}(z)\right) = p_{2,1,1}p_{2,2,1}z^0 + (1 - p_{2,1,1}p_{2,2,1})z^1$$

$$U_{2,11}(z) = \Omega_{\omega_s}\left(u_{2,10}(z), u_{2,6}(z)\right)$$
$$= \left[1 - \left(1 - p_{2,1,1}p_{2,2,1}\right)p_{2,6,2}\right]z^0 + \left(1 - p_{2,1,1}p_{2,2,1}\right)p_{2,6,2}z^1$$

and then the overall $subsystem_2$ is written by

$$U_2(z) = \Omega_{\omega_s}\left(u_{2,11}(z), u_{2,9}(z)\right)$$
$$= \left[1 - \left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\right]z^0$$
$$+ \left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}z^1$$

which can be written in its compact form,

$$U_2(z) = p_{2,1}z^0 + p_{2,2}z^1$$

Observe that $p_{2,3,2}p_{2,4,2} = p_{2,3,2}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,2}p_{2,5,1}$. This technique, however, reduces the number of evaluations of the final term $U_2(z)$ to five equations with four evaluations each, i.e., $5 \times 4 = 20$ evaluations as opposed to $2^6 = 64$ evaluations. To find the reliability of $subsystem_2$, we obtain

$$R_2 = U'_{g_2}(z)\big|_{z=1} = \frac{d}{dz}U_{g_2}(1) = E[G_2] = p_{2,2}$$
$$= \left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}$$

***subsystem*$_3$ formulation.** We first obtain the corresponding $U_3(z)$,

$$U_3(z)$$
$$= \Omega_\omega\left(u_{3,1}(z), u_{3,2}(z), u_{3,3}(z), u_{3,4}(z), u_{3,5}(z), u_{3,6}(z), u_{3,7}(z), u_{3,8}(z), u_{3,9}(z), u_{3,10}(z)\right)$$
$$= \Omega_\omega\left[\sum_{k_{3,1}=1}^{2} p_{3,1,k_{3,1}}\, z^{g_{3,1,k_{3,1}}}, \sum_{k_{3,2}=1}^{2} p_{3,2,k_{3,2}}\, z^{g_{3,2,k_{3,2}}}, \ldots, \sum_{k_{3,10}=1}^{2} p_{3,10,k_{3,10}}\, z^{g_{3,10,k_{3,10}}}\right]$$

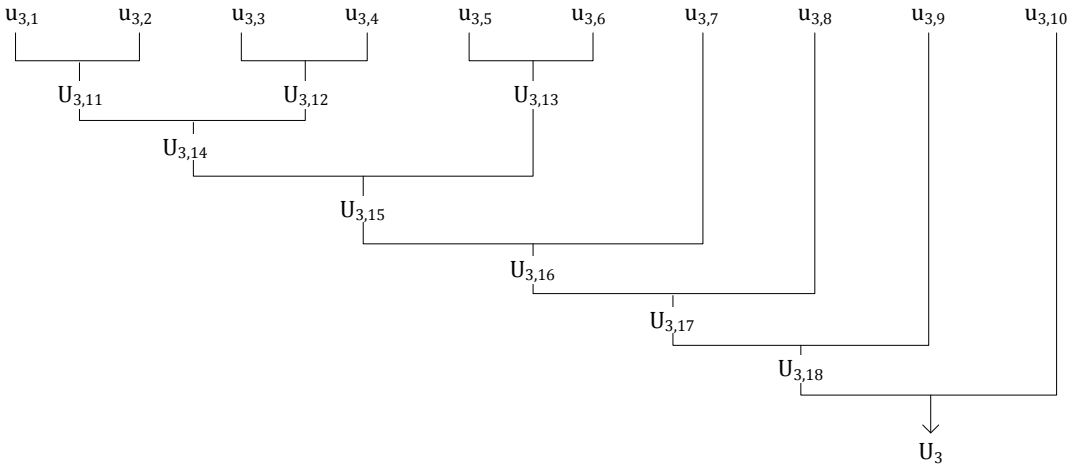

Figure 5-4: Order of evaluation for the function $U_3(z)$

We then solve $U_3(z)$ following the evaluation order depicted in Figure 5-4. Similarly, we build the recursive procedures, collecting like terms, according to the following order, as follows

$$U_{3,11}(z) = \Omega_{\omega_p}\left(u_{3,1}(z), u_{3,2}(z)\right) = p_{3,1,1}p_{3,2,1}z^0 + \left(1 - p_{3,1,1}p_{3,2,1}\right)z^1.$$

$$U_{3,12}(z) = \Omega_{\omega_p}\left(u_{3,3}(z), u_{4,4}(z)\right) = p_{3,3,1}p_{3,4,1}z^0 + \left(1 - p_{3,3,1}p_{3,4,1}\right)z^1.$$

$$U_{3,13}(z) = \Omega_{\omega_p}\left(u_{3,5}(z), u_{3,6}(z)\right) = p_{3,5,1}p_{3,6,1}z^0 + \left(1 - p_{3,5,1}p_{3,6,1}\right)z^1.$$

$$U_{3,14}(z) = \Omega_{\omega_s}\left(u_{3,11}(z), u_{3,12}(z)\right) = \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)z^1.$$

$$U_{3,15}(z) = \Omega_{\omega_s}\left(u_{3,14}(z), u_{3,13}(z)\right)$$
$$= \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)z^1.$$

$$U_{3,16}(z) = \Omega_{\omega_s}\left(u_{3,15}(z), u_{3,7}(z)\right)$$
$$= \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}z^1.$$

$$U_{3,17}(z) = \Omega_{\omega_s}\left(u_{3,16}(z), u_{3,8}(z)\right)$$
$$= \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}z^1.$$

$$U_{3,18}(z) = \Omega_{\omega_s}\left(u_{3,17}(z), u_{3,9}(z)\right)$$
$$= \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}z^1.$$

$$U_3(z) = \Omega_{\omega_s}\left(u_{3,18}(z), u_{3,10}(z)\right)$$
$$= \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}\right]z^0$$
$$+\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}z^1.$$

When $U_3(z)$ represented using the short form,

$$U_3(z) = p_{3,1}z^0 + p_{3,2}z^1$$

Note that this technique reduces the number of evaluations of the final term $U_3(z)$ to nine equations with four evaluations each, i.e., $9 \times 4 = 36$ evaluations as opposed to $2^{10} = 1024$ evaluations. To find the reliability of $subsystem_3$, we obtain

$$R_3 = U'_{g_3}(z)\big|_{z=1} = p_{3,2}$$
$$= \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}$$

***Subsystem*$_4$ formulation**. We obtain the corresponding $U_4(z)$ as follows.

$$U_4(z) = \Omega_\omega\left(u_{4,1}(z), u_{4,2}(z), u_{4,3}(z), u_{4,4}(z)\right)$$

$$= \Omega_\omega\left[\sum_{k_{4,1}=1}^{2} p_{4,1,k_{4,1}}\, z^{g_{4,1,k_{4,1}}}, \ldots, \sum_{k_{4,4}=1}^{2} p_{4,4,k_{4,4}}\, z^{g_{4,4,k_{4,4}}}\right]$$

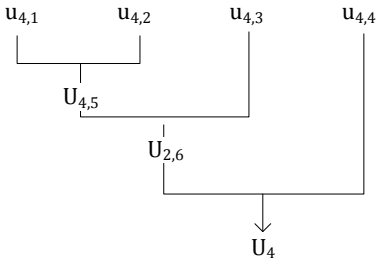

Figure 5-5: Order of evaluation for the function $U_4(z)$

We then compute $U_4(z)$ following the order of evaluation depicted in Figure 5-5 as follows. Building recursive procedures with like terms collected leads to,

$$U_{4,5}(z) = \Omega_{\omega_s}\left(u_{4,1}(z), u_{4,2}(z)\right) = (1 - p_{4,1,2}p_{4,2,2})z^0 + (p_{4,1,2}p_{4,2,2})z^1$$

$$U_{4,6}(z) = \Omega_{\omega_s}\left(u_{4,5}(z), u_{4,3}(z)\right) = (1 - p_{4,1,2}p_{4,2,2}p_{4,3,2})z^0 + (p_{4,1,2}p_{4,2,2}p_{4,3,2})z^1$$

$$U_4(z) = \Omega_{\omega_s}\left(u_{4,6}(z), u_{4,4}(z)\right) = (1 - p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2})z^0 + (p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2})z^1$$

which can be represented using the short form,

$$U_4(z) = p_{4,1}z^0 + p_{4,2}z^1$$

This technique reduces the number of evaluations of the final term $U_4(z)$ to three equations with four evaluations each, i.e., $3 \times 4 = 12$ evaluations as opposed to $2^4 = 16$ evaluations. To find the reliability of $subsystem_4$, we obtain

$$R_4 = U'_{g_4}(z)\big|_{z=1} = p_{4,2}$$

$$= p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}$$

***Subsystem*$_5$ formulation**. The corresponding $U_5(z)$ is obtained as follows.

$$U_5(z) = \Omega_\omega\left(u_{5,1}(z), u_{5,2}(z), u_{5,3}(z), u_{5,4}(z), u_{5,5}(z)\right)$$
$$= \Omega_\omega\left[\sum_{k_{5,1}=1}^{2} p_{5,1,k_{5,1}}\, z^{g_{5,1,k_{5,1}}}, \ldots, \sum_{k_{5,5}=1}^{2} p_{5,5,k_{5,5}}\, z^{g_{5,5,k_{5,5}}}\right]$$

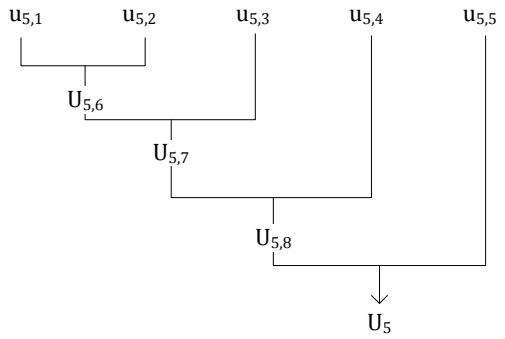

Figure 5-6: Order of evaluation for the function $U_5(z)$

Similarly, we build the recursive procedures, collecting like terms, according to the evaluation order depicted in Figure 5-6 as follows.

$$U_{5,6}(z) = \Omega_{\omega_s}\left(u_{5,1}(z), u_{5,2}(z)\right) = \left(1 - p_{5,1,2}p_{5,2,2}\right)z^0 + \left(p_{5,1,2}p_{5,2,2}\right)z^1.$$

$$U_{5,7}(z) = \Omega_{\omega_s}\left(u_{5,6}(z), u_{5,3}(z)\right) = \left(1 - p_{5,1,2}p_{5,2,2}p_{5,3,2}\right)z^0 + \left(p_{5,1,2}p_{5,2,2}p_{5,3,2}\right)z^1.$$

$$U_{5,8}(z) = \Omega_{\omega_s}\left(u_{5,7}(z), u_{5,4}(z)\right) = \left(1 - p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}\right)z^0 + \left(p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}\right)z^1$$

$$U_5(z) = \Omega_{\omega_s}\left(u_{5,8}(z), u_{5,5}(z)\right)$$
$$= \left(1 - p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\right)z^0 + \left(p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\right)z^1.$$

which can be represented using the short form,

$$U_5(z) = p_{5,1}z^0 + p_{5,2}z^1$$

This technique reduces the number of evaluations of the final term $U_5(z)$ to four equations with four evaluations each, i.e., $4 \times 4 = 16$ evaluations as opposed to $2^5 = 32$ evaluations. To find the reliability of $subsystem_5$, we obtain

$$R_5 = U'_{g_5}(z)\big|_{z=1} = p_{5,2} = p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}$$

***System*$_{ISMM}$ formulation.** Finally the *u*-function of the overall security system $U_{\text{ISMM}}(z)$ is obtained as follows.

$$\begin{aligned} U_{ISMM}(z) &= \Omega_{\omega}\big(u_1(z), u_2(z), u_3(z), u_4(z), u_5(z)\big) \\ &= \Omega_{\omega}\left[\sum_{k_1=1}^{2} p_{1,k_1}\, z^{g_{1,k_1}}, \ldots, \sum_{k_5=1}^{2} p_{5,k_5}\, z^{g_{5,k_5}}\right] \end{aligned}$$

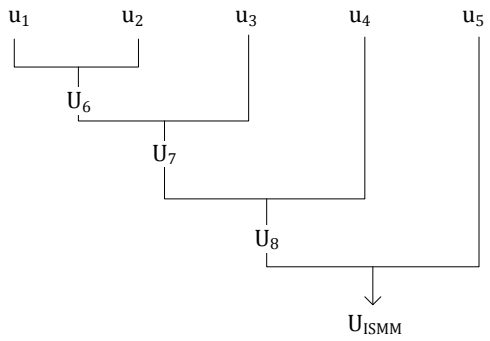

Figure 5-7: Order of evaluation for the function $U_{ISMM}(z)$

Similarly, we build the recursive procedures combined with like terms collected according to the order depicted in Figure 5-7. In the short form, $U_6(z)$ using subsystem-level representation,

$$\begin{aligned} U_6(z) &= \Omega_{\omega_s}\big(u_1(z), u_2(z)\big) \\ &= (1 - p_{1,2}p_{2,2})z^0 + p_{1,2}p_{2,2}z^1 \end{aligned}$$

And in its long form using control-level representation,

$$U_6(z) = \Omega_{\omega_s}\big(u_1(z), u_2(z)\big)$$

$$= \left[1 - \left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\left(1 - p_{1,5,1}p_{1,6,1}\right)\left(1 - p_{1,7,1}p_{1,8,1}\right)\left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\right]z^0 + \left[\left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\left(1 - p_{1,5,1}p_{1,6,1}\right)\left(1 - p_{1,7,1}p_{1,8,1}\right)\left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\right]z^1$$

To find $U_7(z)$ in its short form, we write

$$U_7(z) = \Omega_{\omega_s}\left(u_6(z), u_3(z)\right) = (1 - p_{1,2}p_{2,2}p_{3,2})z^0 + p_{1,2}p_{2,2}p_{3,2}z^1$$

and using the long form,

$$U_7(z) = \Omega_{\omega_s}\left(u_6(z), u_3(z)\right)$$
$$= \left[1 - \left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\left(1 - p_{1,5,1}p_{1,6,1}\right)\left(1 - p_{1,7,1}p_{1,8,1}\right)\left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}\right]z^0 + \left[\left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\left(1 - p_{1,5,1}p_{1,6,1}\right)\left(1 - p_{1,7,1}p_{1,8,1}\right)\left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}\right]z^1$$

To find $U_8(z)$ in its short form, we write

$$U_8(z) = \Omega_{\omega_s}\left(u_7(z), u_4(z)\right) = (1 - p_{1,2}p_{2,2}p_{3,2}p_{4,2})z^0 + p_{1,2}p_{2,2}p_{3,2}p_{4,2}z^1$$

and using the long form,

$$U_8(z) = \Omega_{\omega_s}\left(u_7(z), u_4(z)\right)$$

$$= \big[1 - \big(1 - p_{1,1,1}p_{1,2,1}\big)\big(1 - p_{1,3,1}p_{1,4,1}\big)\big(1 - p_{1,5,1}p_{1,6,1}\big)\big(1 - p_{1,7,1}p_{1,8,1}\big)\big(1$$
$$- p_{2,1,1}p_{2,2,1}\big)\big(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\big)p_{2,6,2}\big(1$$
$$- p_{3,1,1}p_{3,2,1}\big)\big(1 - p_{3,3,1}p_{3,4,1}\big)\big(1$$
$$- p_{3,5,1}p_{3,6,1}\big)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}\big]z^0$$
$$+ \big[\big(1 - p_{1,1,1}p_{1,2,1}\big)\big(1 - p_{1,3,1}p_{1,4,1}\big)\big(1 - p_{1,5,1}p_{1,6,1}\big)\big(1 - p_{1,7,1}p_{1,8,1}\big)\big(1$$
$$- p_{2,1,1}p_{2,2,1}\big)\big(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\big)p_{2,6,2}\big(1$$
$$- p_{3,1,1}p_{3,2,1}\big)\big(1 - p_{3,3,1}p_{3,4,1}\big)\big(1$$
$$- p_{3,5,1}p_{3,6,1}\big)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}\big]z^1$$

Finally, to find $U_{ISMM}(z)$ in its compact form, we write

$$U_{ISMM}(z) = \Omega_{\omega_s}\big(u_8(z), u_5(z)\big)$$
$$= \big(1 - p_{1,2}p_{2,2}p_{3,2}p_{4,2}p_{5,2}\big)z^0 + p_{1,2}p_{2,2}p_{3,2}p_{4,2}p_{5,2}z^1$$
$$= p_{ISMM,1}z^0 + (1 - p_{ISMM,1})z^1$$

which can be written using the extended form as

$$U_{ISMM}(z) = \Omega_{\omega_s}\big(u_8(z), u_5(z)\big)$$
$$= \big[1 - \big(1 - p_{1,1,1}p_{1,2,1}\big)\big(1 - p_{1,3,1}p_{1,4,1}\big)\big(1 - p_{1,5,1}p_{1,6,1}\big)\big(1 - p_{1,7,1}p_{1,8,1}\big)\big(1$$
$$- p_{2,1,1}p_{2,2,1}\big)\big(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\big)p_{2,6,2}\big(1$$
$$- p_{3,1,1}p_{3,2,1}\big)\big(1 - p_{3,3,1}p_{3,4,1}\big)\big(1$$
$$- p_{3,5,1}p_{3,6,1}\big)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\big]z^0$$
$$+ \big[\big(1 - p_{1,1,1}p_{1,2,1}\big)\big(1 - p_{1,3,1}p_{1,4,1}\big)\big(1 - p_{1,5,1}p_{1,6,1}\big)\big(1 - p_{1,7,1}p_{1,8,1}\big)\big(1$$
$$- p_{2,1,1}p_{2,2,1}\big)\big(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\big)p_{2,6,2}\big(1$$
$$- p_{3,1,1}p_{3,2,1}\big)\big(1 - p_{3,3,1}p_{3,4,1}\big)\big(1$$
$$- p_{3,5,1}p_{3,6,1}\big)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\big]z^1$$

Observe that this technique reduces the number of evaluations of the final term $U_{\mathrm{ISMM}}(z)$ to four equations with four evaluations each, i.e., $4 \times 4 = 16$ evaluations as opposed to $2^5 = 32$ evaluations. To find the reliability of the overall $system_{ISMM}$, we obtain

$$R_{ISMM} = U'_{g_{ISMM}}(z)\big|_{z=1}$$
$$= \left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\left(1 - p_{1,5,1}p_{1,6,1}\right)\left(1 - p_{1,7,1}p_{1,8,1}\right)\left(1 - p_{2,1,1}p_{2,2,1}\right)\left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)p_{2,6,2}\left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)$$
$$p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}$$

Above we have shown the derivations based on fixed probabilities. Observe that these equations can be used for evaluating reliability changes over time using the same relationships derived, by simply substituting $p_{i,j,k_{i,j}}$ with $p_{i,j,k_{i,j}}(t)$.

5.5 Numerical Example

Time-independent reliabilities: To demonstrate analysis of this part, the same fixed probabilities of failure of controls, represented by $P\{\mathrm{X}_{i,j} = 0\} = q_{i,j}$ for all $i, j's$ in Section 4.4.3, are used to evaluate the corresponding *u*-functions of controls. We demonstrate how analysis of intermediate and final *u*-functions can be performed systematically.

***Subsystem*$_1$ analysis**. Recall that the *u*-function representing $subsystem_1$ is written by

$$U_1(z) = \Omega_\omega\left(u_{1,1}(z), u_{1,2}(z), \dots, u_{1,8}(z)\right)$$

Substituting the failure probabilities of controls leads to,

$$u_{1,1}(z) = p_{1,1,1}z^0 + \left(1 - p_{1,1,1}\right)z^1 = 0.07z^0 + 0.93z^1$$

$$u_{1,2}(z) = p_{1,2,1}z^0 + \left(1 - p_{1,2,1}\right)z^1 = 0.055z^0 + 0.945z^1$$

And so on for the remaining controls until the function,

$$u_{1,8}(z) = p_{1,8,1}z^0 + \left(1 - p_{1,8,1}\right)z^1 = 0.017z^0 + 0.983z^1$$

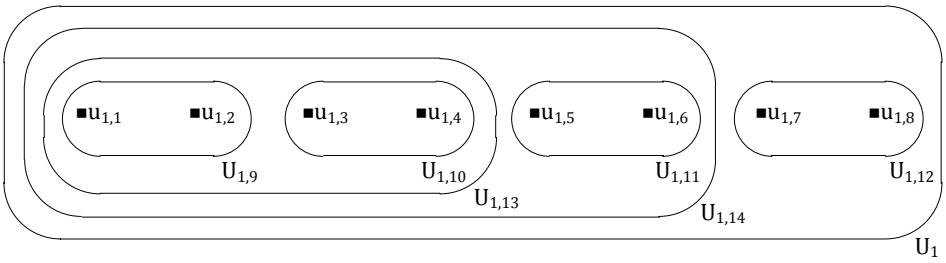

Figure 5-8: Intermediate u-functions for $subsystem_1$

Observe that this method facilitates a convenient way to study the performance of different intermediate permutations for a given structure of multiple items. This analysis is performed according to the arrangement and failure probabilities defined by lower-level u-functions. Various intermediate u-functions for $subsystem_1$ are depicted in Figure 5-8. For instance, to evaluate the u-function of the combined CCTV controls, $C_{1,1}$ and $C_{1,2}$, we find

$$\begin{aligned} U_{1,9}(z) &= \Omega_{\omega_p}\left(u_{1,1}(z), u_{1,2}(z)\right) \\ &= p_{1,1,1}p_{1,2,1}z^0 + \left(1 - p_{1,1,1}p_{1,2,1}\right)z^1 \\ &= (0.07 \times 0.055)z^0 + [1 - (0.07 \times 0.055)]z^1 \\ &= 0.00385z^0 + 0.99615z^1 \end{aligned}$$

To evaluate the reliability of this structure we calculate

$$R_{1,9} = U'_{g_{1,9}}(z)\big|_{z=1} = \frac{d}{dz}U_{g_{1,9}}(1) = 0.99615$$

Further, to evaluate the performance of the combined controls of CCTVs and locks, i.e., $C_{1,1}, C_{1,2}, C_{1,3}$, and $C_{1,4}$, modelled by $U_{1,13}(z)$, we find

$$\begin{aligned} U_{1,13}(z) &= \Omega_{\omega_s}\left(u_{1,9}(z), u_{1,10}(z)\right) \\ &= \left[1 - \left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)\right]z^0 + \left(1 - p_{1,1,1}p_{1,2,1}\right)\left(1 - p_{1,3,1}p_{1,4,1}\right)z^1 \\ &= [1 - (1 - 0.07 \times 0.055)(1 - 0.034 \times 0.032)]z^0 \\ &+ (1 - 0.07 \times 0.055)(1 - 0.034 \times 0.032)z^1 \\ &= 0.00493z^0 + 0.99507z^1 \end{aligned}$$

leading to the reliability of this structure,

$$R_{1,13} = \frac{d}{dz} U_{g_{1,13}}(1) = 0.99507$$

And so on for the remaining different combinations of controls. The overall performance of $subsystem_1$ can then be calculated by

$$\begin{aligned} U_1(z) &= \Omega_{\omega_s}\left(u_{1,14}(z), u_{1,12}(z)\right) \\ &= [1 - (1 - 0.07 \times 0.055)(1 - 0.034 \times 0.032)(1 - 0.029 \times 0.025)(1 - 0.019 \times 0.017)]z^0 \\ &\quad + [(1 - 0.07 \times 0.055)(1 - 0.034 \times 0.032)(1 - 0.029 \times 0.025)(1 - 0.019 \\ &\quad \times 0.017)]z^1 \\ &= 0.00598z^0 + 0.99402z^1 \end{aligned}$$

leading to finding the reliability by

$$R_1 = \frac{d}{dz} U_{g_1}(1) = E[G_1] = 0.99402$$

***Subsystem*$_2$ analysis.** Recall that the corresponding u-function is written by $U_2(z) = \Omega_{\omega}\left(u_{2,1}(z), u_{2,2}(z), u_{2,3}(z), u_{2,4}(z), u_{2,5}(z), u_{2,6}(z)\right)$. Different intermediate aggregations of controls can be evaluated in a manner similar to the earlier analysis of $subsystem_1$. For instance, the performance distribution of the intermediate 2-out-of-3 structure representing the web servers controls, i.e., $C_{2,3}, C_{2,4}$ and $C_{2,5}$, is found by

$$\begin{aligned} &U_{2,9}(z) \\ &\quad = \left(1 - \left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)\right)z^0 \\ &\quad + \left(p_{2,3,2}p_{2,4,2} + p_{2,3,1}p_{2,4,2}p_{2,5,2} + p_{2,3,2}p_{2,4,1}p_{2,5,2}\right)z^1 \\ &\quad = \left(1 - (0.969 \times 0.971 + 0.031 \times 0.971 \times 0.965 + 0.969 \times 0.029 \times 0.965)\right)z^0 \\ &\quad + (0.969 \times 0.971 + 0.031 \times 0.971 \times 0.965 + 0.969 \times 0.029 \times 0.965)z^1 \\ &= 0.00294z^0 + 0.99706z^1 \end{aligned}$$

with reliability,

$$R_{2,9} = \frac{d}{dz} U_{g_{2,8}}(1) = 0.99706$$

The overall performance of the subsystem,

$$U_2(z) = [1 - (1 - 0.02 \times 0.018)(0.969 \times 0.971 + 0.031 \times 0.971 \times 0.965 + 0.969 \times 0.029 \times 0.965)0.971]z^0 + [(1 - 0.02 \times 0.018)(0.969 \times 0.971 + 0.031 \times 0.971 \times 0.965 + 0.969 \times 0.029 \times 0.965)0.971]z^1 = 0.0322z^0 + 0.96780z^1$$

leading to,

$$R_2 = \frac{d}{dz}U_{g_2}(1) = 0.96780$$

Subsystem$_3$ **analysis.** Recall that the corresponding $U_3(z)$ is formulated by $U_3(z) = \Omega_\omega\left(u_{3,1}(z), u_{3,2}(z), u_{3,3}(z), u_{3,4}(z), u_{3,5}(z), u_{3,6}(z), u_{3,7}(z), u_{3,8}(z), u_{3,9}(z), u_{3,10}(z)\right)$. As an example of an intermediate performance of the subsets of $U_3(z)$, the performance of $U_{3,11}(z)$, representing the external firewall controls $C_{3,1}$ and $C_{3,2}$, is obtained by

$$U_{3,11}(z) = p_{3,1,1}p_{3,2,1}z^0 + \left(1 - p_{3,1,1}p_{3,2,1}\right)z^1 = (0.021 \times 0.023)z^0 + (1 - 0.021 \times 0.023)z^1 = 0.00048z^0 + 0.99952z^1$$

leading to,

$$R_{3,11} = \frac{d}{dz}U_{g_{3,11}}(1) = 0.99952$$

The resulting $subsystem_3$ performance distribution, however, can be found by

$$U_3(z) = \left[1 - \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}\right]z^0 + \left(1 - p_{3,1,1}p_{3,2,1}\right)\left(1 - p_{3,3,1}p_{3,4,1}\right)\left(1 - p_{3,5,1}p_{3,6,1}\right)p_{3,7,2}p_{3,8,2}p_{3,9,2}p_{3,10,2}z^1 = [1 - (1 - 0.021 \times 0.023)(1 - 0.011 \times 0.012)(1 - 0.017 \times 0.013)0.967 \times 0.971 \times 0.975 \times 0.972]z^0 + [(1 - 0.021 \times 0.023)(1 - 0.011 \times 0.012)(1 - 0.017 \times 0.013)0.967 \times 0.971 \times 0.975 \times 0.972]z^1 = 0.11089z^0 + 0.88911z^1$$

leading to,

$$R_3 = \frac{d}{dz} U_{g_3}(1) = 0.88911$$

***Subsystem*$_4$ analysis.** Recall that the corresponding $U_4(z)$ is written by $U_4(z) = \Omega_\omega \left(u_{4,1}(z), u_{4,2}(z), u_{4,3}(z), u_{4,4}(z)\right)$. To evaluate the intermediate $U_{4,5}(z)$, representing the social engineering and training controls $C_{4,1}$ and $C_{4,2}$, respectively, we find

$$\begin{aligned} U_{4,5}(z) &= \left(1 - p_{4,1,2}p_{4,2,2}\right)z^0 + \left(p_{4,1,2}p_{4,2,2}\right)z^1 \\ &= (1 - 0.981 \times 0.98)z^0 + (0.981 \times 0.98)z^1 \\ &= 0.03862z^0 + 0.96138z^1 \end{aligned}$$

leading to,

$$R_{4,5} = \frac{d}{dz} U_{g_{4,5}}(1) = 0.96138$$

The resulting $subsystem_4$ performance distribution, however, is found by

$$\begin{aligned} U_4(z) &= \left(1 - p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}\right)z^0 + \left(p_{4,1,2}p_{4,2,2}p_{4,3,2}p_{4,4,2}\right)z^1 \\ &= (1 - 0.981 \times 0.98 \times 0.977 \times 0.982)z^0 + (0.981 \times 0.98 \times 0.977 \times 0.982)z^1 \\ &= 0.07764z^0 + 0.92236z^1 \end{aligned}$$

leading to,

$$R_4 = \frac{d}{dz} U_{g_4}(1) = 0.92236$$

***Subsystem*$_5$ analysis.** Recall that the corresponding $U_5(z)$ is written by $U_5(z) = \Omega_\omega \left(u_{5,1}(z), u_{5,2}(z), u_{5,3}(z), u_{5,4}(z), u_{5,5}(z)\right)$. To evaluate the intermediate $U_{5,6}(z)$, representing the penetrating testing and risk assessment controls together, $C_{5,1}$ and $C_{5,2}$, respectively, we find

$$\begin{aligned} U_{5,6}(z) &= (1 - p_{5,1,2}p_{5,2,2})z^0 + (p_{5,1,2}p_{5,2,2})z^1 \\ &= (1 - 0.97 \times 0.972)z^0 + (0.97 \times 0.972)z^1 \\ &= 0.05716z^0 + 0.94284z^1 \end{aligned}$$

leading to,

$$R_{5,6} = \frac{d}{dz} U_{g_{5,6}}(1) = 0.94284$$

The resulting $subsystem_5$ performance distribution, however, is found by

$$\begin{aligned} U_5(z) &= \left(1 - p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\right)z^0 + \left(p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\right)z^1 \\ &= \left(1 - p_{5,1,2}p_{5,2,2}p_{5,3,2}p_{5,4,2}p_{5,5,2}\right)z^0 + (0.97 \times 0.972 \times 0.969 \times 0.971 \times 0.965)z^1 \\ &= 0.14393z^0 + 0.85607z^1 \end{aligned}$$

leading to,

$$R_5 = \frac{d}{dz}U_{g_5}(1) = 0.85607$$

***System*$_{ISMM}$ analysis.** Finally, the *u*-function of the overall security system $U_{\mathrm{ISMM}}(z)$ is evaluated as follows.

$$\begin{aligned} U_{ISMM}(z) &= \Omega_{\omega}\left(u_1(z), u_2(z), u_3(z), u_4(z), u_5(z)\right) \\ &= \left(1 - p_{1,2}p_{2,2}p_{3,2}p_{4,2}p_{5,2}\right)z^0 + \left(p_{1,2}p_{2,2}p_{3,2}p_{4,2}p_{5,2}\right)z^1 \\ &= 0.32462z^0 + 0.67538z^1 \end{aligned}$$

leading to its reliability,

$$\begin{aligned} R_{ISMM} &= U'_{g_{ISMM}}(z)\big|_{z=1} \\ &= 0.67538 \end{aligned}$$

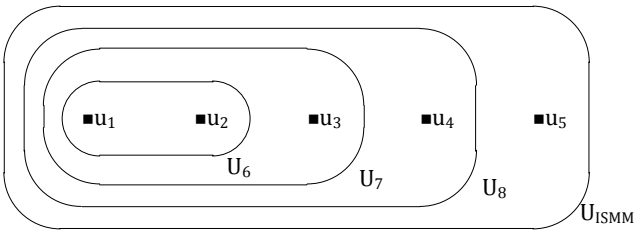

Figure 5-9: Intermediate *u*-functions for $system_{ISMM}$

Similarly, different subsystem-level permutations of the overall system structure can be established for evaluation, as depicted in Figure 5-9. For instance, in order to study the combined performance of $subsystem_1$ and $subsystem_2$, we analyze $U_6(z) = \Omega_{\omega_s}\left(u_1(z), u_2(z)\right)$. The reliability of this configuration equals $\frac{d}{dz}U_{g_6}(1) = 0.96201$. Similarly, we evaluate $U_7(z) = \Omega_{\omega_s}\left(u_6(z), u_3(z)\right)$ for

studying the performance of $subsystem_1$, $subsystem_2$, and $subsystem_3$ together, which leads to the reliability, $\frac{d}{dz}U_{g_7}(1) = 0.85533$. The function $U_8(z) = \Omega_{\omega_s}\big(u_7(z), u_4(z)\big)$ is evaluated for studying $subsystem_4$ and below, which leads to the combined reliability, $\frac{d}{dz}U_{g_8}(1) = 0.78893$. Various *u*-function evaluations along with the corresponding maturity adequacy bounds are shown in Figure 5-10.

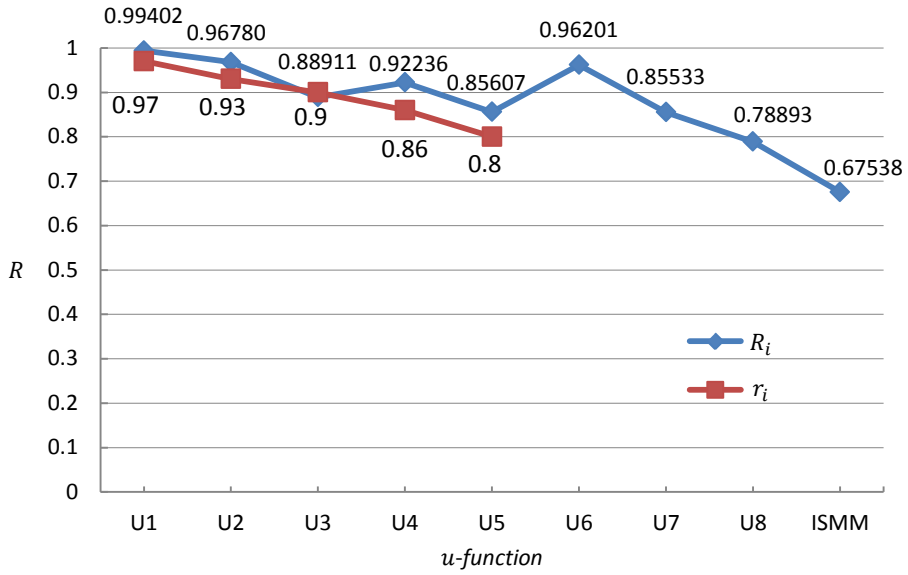

Figure 5-10: ISMM-based reliability analysis for time-independent reliabilities using UGF method

Maturity analysis: Because we are using the UGF method for reliability evaluation, where reliability is founded by the derivative of the *u*-function at $z = 1$, i.e., $\frac{d}{dz}U_{g_i}(1)$, the maturity calculation should follow analytically and graphically to the same results and observations using the reliability-theoretic method found earlier in section 4.4.3 for this example. The maturity acceptability test based on *u*-function representation, however, is written by

$$I\big(R_i \geq threshold(R_i)\big) = I\left(\frac{d}{dz}U_{g_i}(1) \geq threshold\left(\frac{d}{dz}U_{g_i}(1)\right)\right)$$

$$= I\left(\frac{d}{dz}U_{g_i}(1) \geq r_i\right), for\ all\ i = 1, \ldots, 5$$

Applying this leads to us constructing the following tests

$$I\left(\frac{d}{dz}U_{g_1}(1) \geq r_1\right) = I\left(\frac{d}{dz}U_{g_2}(1) \geq r_2\right) = 1$$
$$I\left(\frac{d}{dz}U_{g_3}(1) \geq r_3\right) = 0$$
$$I\left(\frac{d}{dz}U_{g_4}(1) \geq r_4\right) = I\left(\frac{d}{dz}U_{g_5}(1) \geq r_5\right) = 1$$

As a result,

$$\mathcal{M}_{ISMM} = \mathcal{M}_{UGF} = \left\{max(m)\, s.t. \left\{\prod_{i=1}^{m} I\left(\frac{d}{dz}U_{g_i}(1) \geq threshold(R_i)\right)\right\} = 1\right\} = 2$$

Time-dependent reliabilities: In this part we demonstrate the same example exponential model under the same assumptions and failure rates used earlier in Section 4.4.3 for explaining the reliability-based approach. One can see that the *u*-function representation can be used directly to represent the performance of various selections of controls for a given subsystem and the system overall. The analytical derivations of $U_1(z)$ function lead to naturally building and computing six different transient *u*-functions, representing different structures before concluding the resulting $U_1(z)$ for overall performance of $subsystem_1$. These *u*-functions show the progressive buildup of the final *u*-function. Recall that the analysis of $subsystem_1$ presented in Section 4.4.3 considered the structure as a series-parallel arrangement by four series 1-out-of-2 structures, named $S_{1,1}, S_{1,2}, S_{1,3}$, and $S_{1,4}$. Using the UGF method, these structures correspond to $U_{1,9}(z), U_{1,10}(z), U_{1,11}(z)$, and $U_{1,12}(z)$, respectively. The remaining intermediate *u*-functions, namely, $U_{1,13}(z)$, and $U_{1,14}(z)$, model the remaining steps of the structural growth towards building $U_1(z)$. To show this, recall that the derivative of the $U_1(z)$ function at state $z = 1$ leads to

$$\begin{aligned}\frac{d}{dz}U_{g_1}(1) &= p_{1,2}(t)\\ &= R_1(t) = E[G_1]\\ &= \left(1 - p_{1,1,1}(t)p_{1,2,1}(t)\right)\left(1 - p_{1,3,1}(t)p_{1,4,1}(t)\right)\\ &\quad \left(1 - p_{1,5,1}(t)p_{1,6,1}(t)\right)\left(1 - p_{1,7,1}(t)p_{1,8,1}(t)\right)\end{aligned}$$

Substituting the exponential model described in Section 4.4.3,

$$R_1(t) = \left[2e^{-\lambda_{1,1}t} - e^{-2\lambda_{1,1}t}\right]\left[2e^{-\lambda_{1,3}t} - e^{-2\lambda_{1,3}t}\right]\left[2e^{-\lambda_{1,5}t} - e^{-2\lambda_{1,5}t}\right]\left[2e^{-\lambda_{1,7}t} - e^{-2\lambda_{1,7}t}\right]$$

which is the same result found earlier using the reliability-based approach. Thus, the remaining analysis and calculations shall follow directly, such as the reliability for the mission time of one week $R_1(1) = 0.9727$, the mean time to failure $MTTF_1 = 6.7308\, weeks$, the reliability for the mission time of one day $R_1(1/7) = 0.9994$, the reliability for a mission time of 30 days $R_1(30/7) = 0.6675$, and the reliability equals three nines, i.e., $R_1(t) = 0.999$ when the mission time $t = 30.2\, hours$ instead of $t = 1\, week$. Moreover, this analysis can be carried out similarly on intermediate u-functions. The reliability of intermediate structures and the overall $subsystem_1$ structure derived from their corresponding u-functions is depicted in Figure 5-11. Table 5-1 also outlines some reliability-specific analysis for such functions.

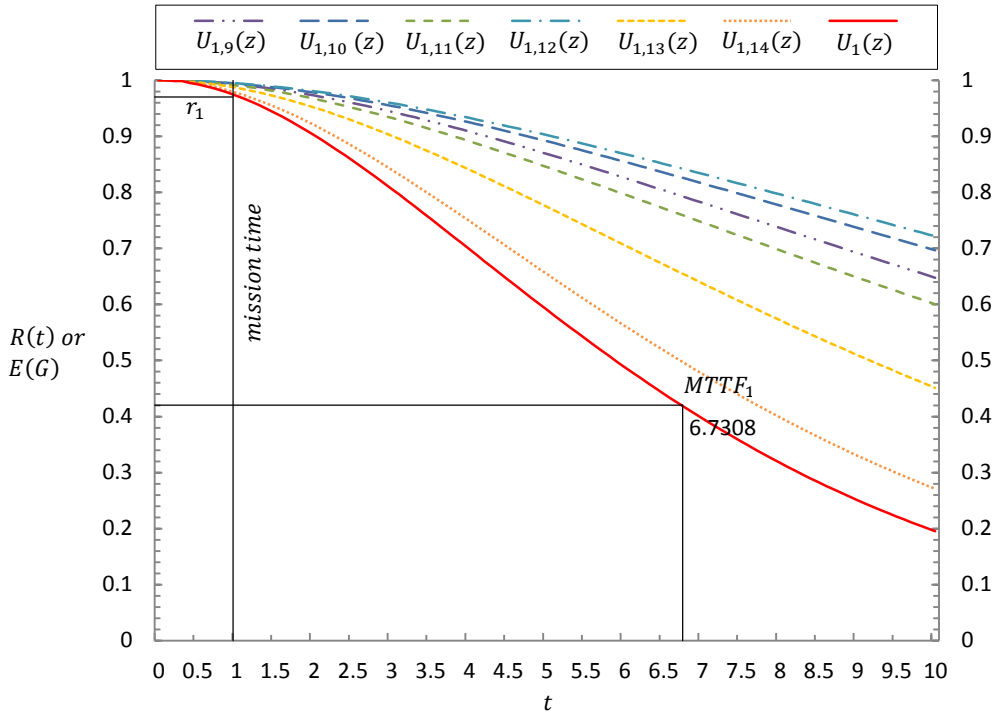

Figure 5-11: ISMM-based time-dependent reliability analysis for $subsystem_1$ using UGF method

Table 5-1: Time-dependent reliability analysis of intermediate and final u-functions for $subsystem_1$

u-function	$R_1(1)$	$MTTF_1$	$R_1(1/7)$	$R_1(30/7)$	$t;\ where\ R_1(t) = 0.999$
$U_{1,9}(z)$	0.9926	16.6667	0.9998	0.8976	0.35
$U_{1,10}(z)$	0.9941	18.75	0.9999	0.9157	0.4
$U_{1,11}(z)$	0.9909	15	0.9998	0.8785	0.32
$U_{1,12}(z)$	0.9948	20	0.9999	0.9244	0.42
$U_{1,13}(z)$	0.9867	10.7783	0.9997	0.8220	0.26
$U_{1,14}(z)$	0.9778	7.7650	0.9995	0.7221	0.2
$U_1(z)$	0.9727	6.7308	0.9994	0.6675	0.18

The reliability of intermediate structures and overall subsystem structure for $subsystem_2, subsystem_3$, $subsystem_4$, and $subsystem_5$ derived from their corresponding u-functions are depicted in Figure 5-12, Figure 5-13, Figure 5-14, and Figure 5-15, respectively. Table 5-2, Table 5-3, Table 5-4, and Table 5-5 also outline some reliability-specific analysis for such functions. At system level, the evaluation of intermediate u-functions of subsystems is depicted in Figure 5-16, whereas the corresponding reliability-specific analysis is shown in Table 5-6.

Observe how the progressive buildup of intermediate u-functions shows a faster decay when moving serially towards the resulting u-function. For instance, $U_{3,15}(z)$ decays faster than $U_{3,14}(z)$ but slower than $U_{3,16}(z)$. Furthermore, the decay gap between $U_{3,14}(z)$ and $U_{3,15}(z)$ is smaller than the gap between $U_{3,15}(z)$ and $U_{3,16}(z)$. This gap is due to the effect of adding the single control $C_{3,7}$ represented by $U_{3,7}(z)$ to the arrangement. This behaviour is also confirmed by observing the corresponding reliability calculations in the corresponding table, namely, $R_3(1), MTTF_3, R_3(1/7), R_3(30/7)$, and $R_3(t) = 0.999$. For example at system level, the curve of $U_2(z)$ is bounded between the curves of $U_1(z)$ and $U_3(z)$. The decay gap also between $U_1(z)$ and $U_2(z)$ is smaller than the gap between $U_2(z)$ and $U_3(z)$, signifying the effect of the single controls at $subsystem_3$ compared to the effect of the redundancy at $subsystem_1$ and $subsystem_2$.

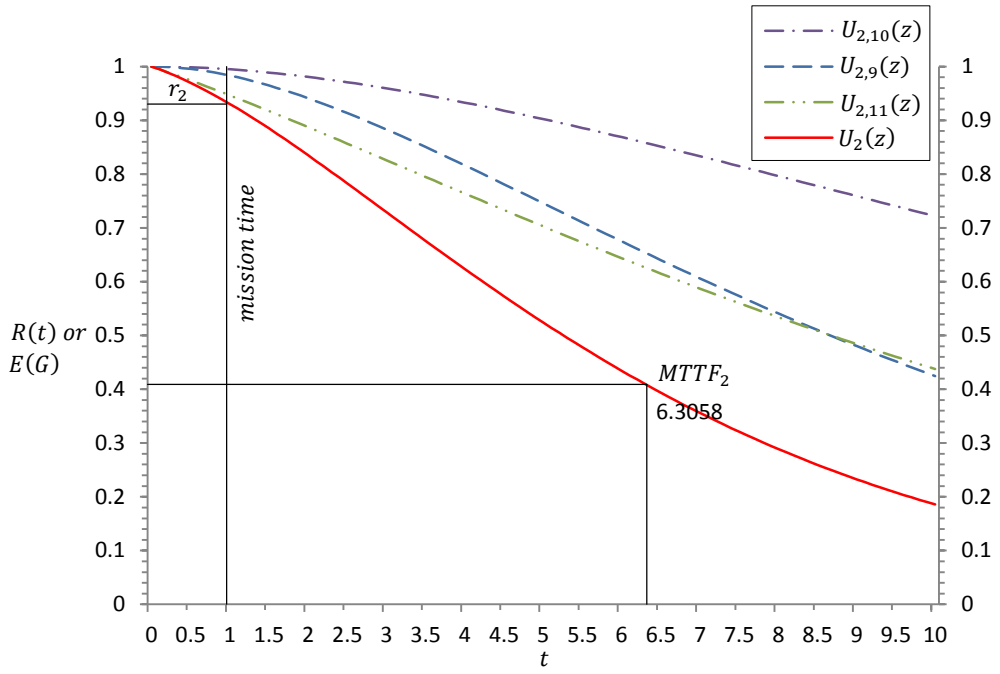

Figure 5-12: ISMM-based time-dependent reliability analysis for $subsystem_2$ using UGF method

Table 5-2: Time-dependent reliability analysis of intermediate and final u-functions for $subsystem_2$

u-function	$R_2(1)$	$MTTF_2$	$R_2(1/7)$	$R_2(30/7)$	t; $where$ $R_2(t) = 0.999$
$U_{2,9}(z)$	0.9832	10.4167	0.9996	0.7962	0.22
$U_{2,10}(z)$	0.9948	20	0.9999	0.9244	0.41
$U_{2,11}(z)$	0.9463	11	0.9928	0.7461	0.02
$U_2(z)$	0.9303	6.3058	0.9924	0.5940	0.018

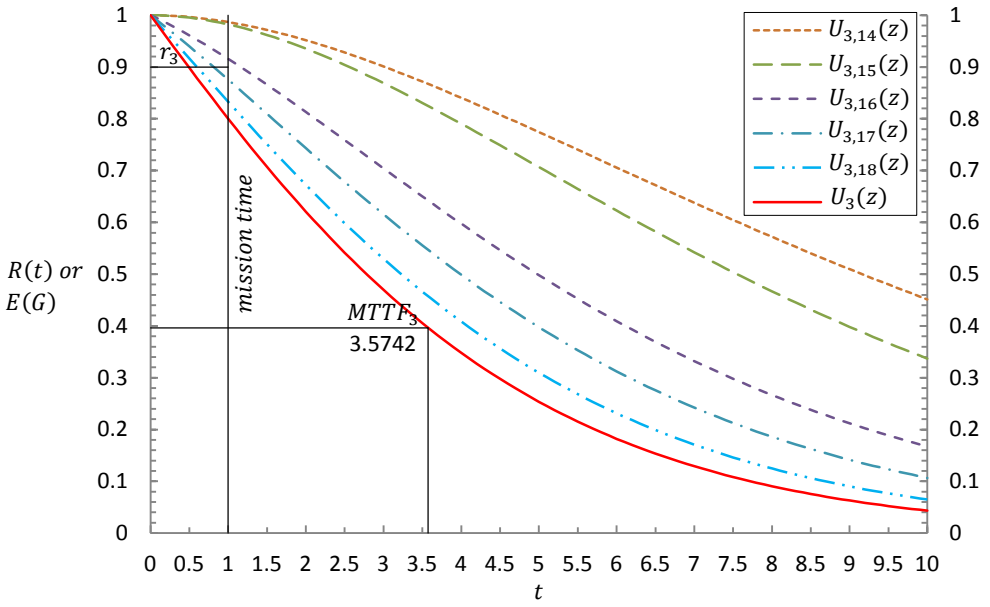

Figure 5-13: ISMM-based time-dependent reliability analysis for $subsystem_3$ using UGF method

Table 5-3: Time-dependent reliability analysis of intermediate and final u-functions for $subsystem_3$

u-function	$R_3(1)$	$MTTF_3$	$R_3(1/7)$	$R_3(30/7)$	$t;\ where$ $R_3(t) = 0.999$
$U_{3,11}(z)$	0.9941	18.75	0.9999	0.9158	0.4
$U_{3,12}(z)$	0.9926	16.6667	0.9998	0.8976	0.35
$U_{3,13}(z)$	0.9954	21.4286	0.9999	0.9328	0.46
$U_{3,14}(z)$	0.9867	10.7783	0.9997	0.8220	0.26
$U_{3,15}(z)$	0.9822	8.7318	0.9996	0.7667	0.22
$U_{3,16}(z)$	0.9158	5.9754	0.9897	0.5680	0.013
$U_{3,17}(z)$	0.8755	4.9120	0.9833	0.4684	0.009
$U_{3,18}(z)$	0.8328	4.0751	0.9763	0.3780	0.006
$U_3(z)$	0.8002	3.5742	0.9708	0.3185	0.005

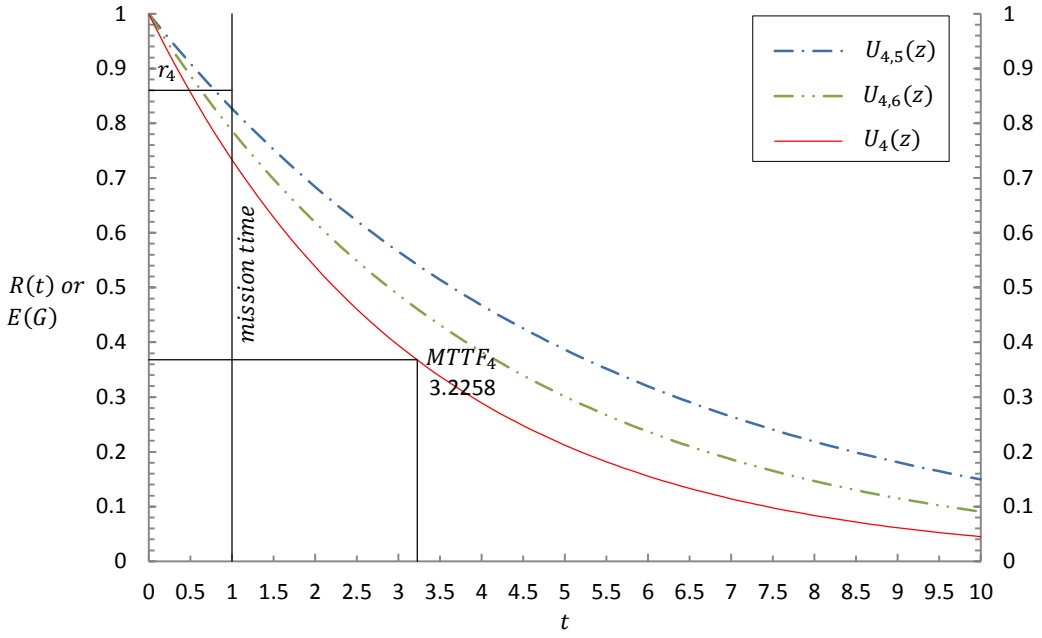

Figure 5-14: ISMM-based time-dependent reliability analysis for $subsystem_4$ using UGF method

Table 5-4: Time-dependent reliability analysis of intermediate and final u-functions for $subsystem_4$

u-function	$R_4(1)$	$MTTF_4$	$R_4(1/7)$	$R_4(30/7)$	t; $where$ $R_4(t) = 0.999$
$U_{4,5}(z)$	0.8270	5.2632	0.9732	0.4430	0.005
$U_{4,6}(z)$	0.7866	4.1667	0.9663	0.3575	0.004
$U_4(z)$	0.7334	3.2258	0.9567	0.2649	0.0032

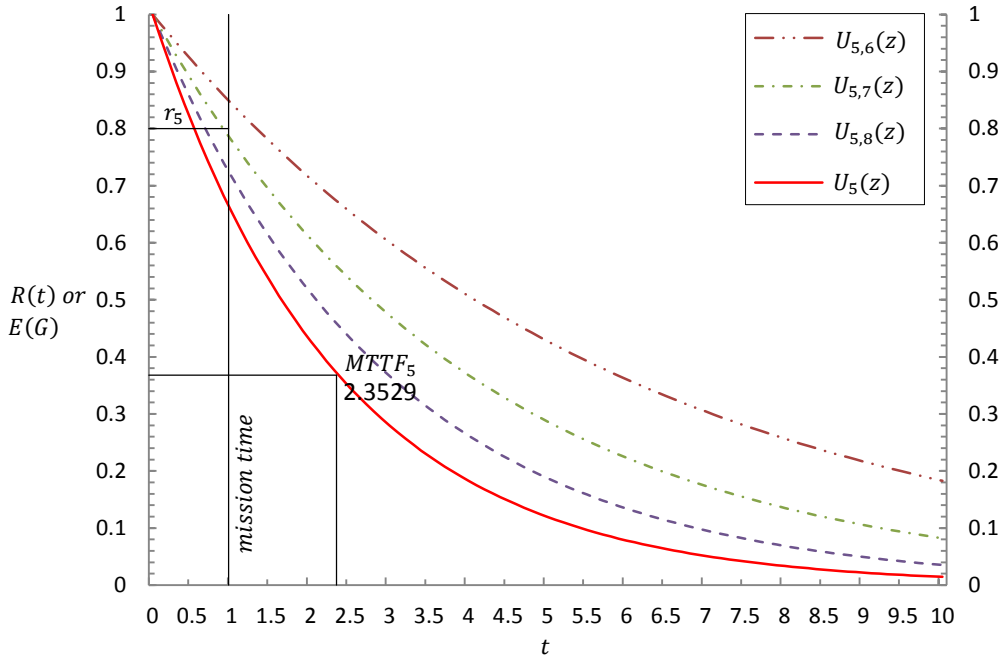

Figure 5-15: ISMM-based time-dependent reliability analysis for $subsystem_5$ using UGF method

Table 5-5: Time-dependent reliability analysis of intermediate and final u-functions for $subsystem_5$

u-function	$R_5(1)$	$MTTF_5$	$R_5(1/7)$	$R_5(30/7)$	t; *where* $R_5(t) = 0.999$
$U_{5,6}(z)$	0.8437	5.8824	0.9760	0.4826	0.006
$U_{6,7}(z)$	0.7788	4	0.9649	0.3425	0.004
$U_{5,8}(z)$	0.7153	2.9851	0.9533	0.2379	0.0029
$U_5(z)$	0.6538	2.3529	0.9411	0.1618	0.0024

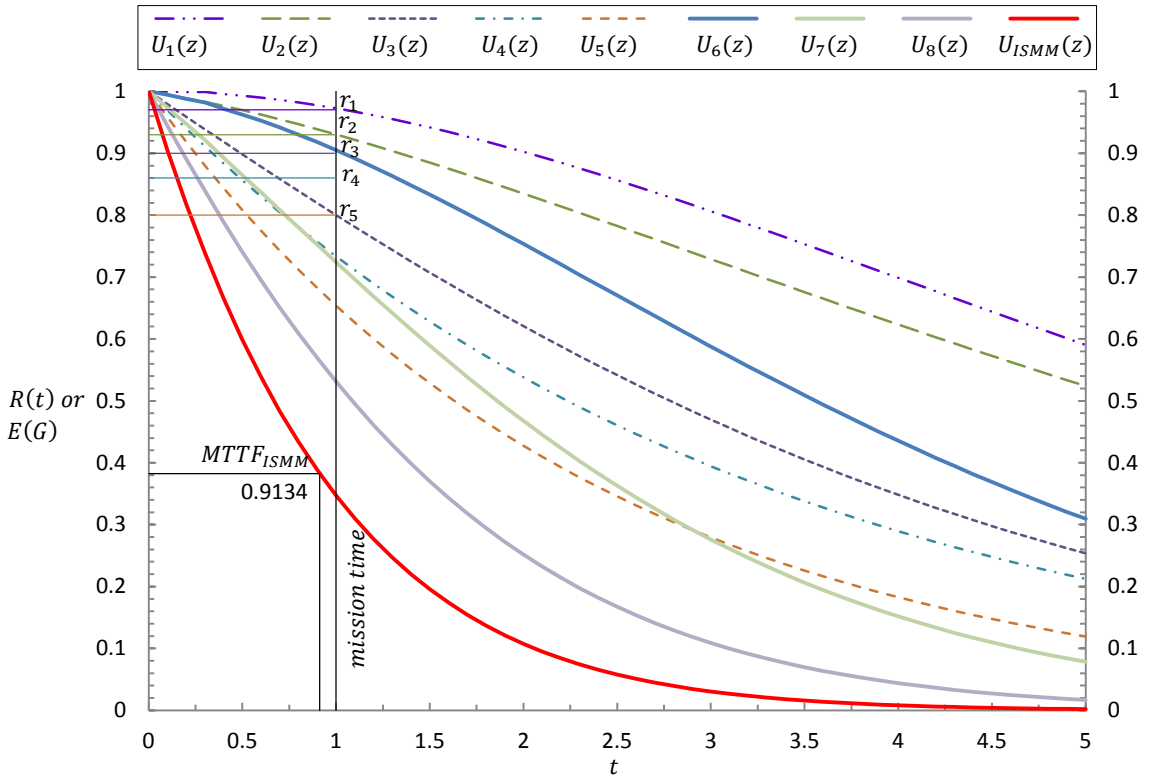

Figure 5-16: ISMM-based time-dependent reliability analysis for $system_{ISMM}$ using UGF method

Table 5-6: Time-dependent reliability analysis of intermediate and final u-functions for $system_{ISMM}$

u-function	$R_{ISMM}(1)$	$MTTF_{ISMM}$	$R_{ISMM}(1/7)$	$R_{ISMM}(30/7)$	t; *where* $R_{ISMM}(t) = 0.999$
$U_1(z)$	0.9727	6.7308	0.9994	0.6675	0.18
$U_2(z)$	0.9303	6.3058	0.9924	0.5940	0.018
$U_3(z)$	0.8002	3.5742	0.9708	0.31849	0.005
$U_4(z)$	0.7334	3.2258	0.9567	0.2649	0.0032
$U_5(z)$	0.6538	2.3529	0.9412	0.1618	0.0024
$U_6(z)$	0.9049	4.0814	0.9918	0.3965	0.0195
$U_7(z)$	0.7241	2.2479	0.9628	0.1263	0.0039

$U_8(z)$	0.5311	1.414	0.9211	0.0334	0.0017
$U_{ISMM}(z)$	0.3472	0.9134	0.8668	0.0054	0.0012

Maturity analysis: As mentioned earlier, the derivative of the u-function at $z = 1$, i.e., $\frac{d}{dz}U_{g_i}(1)$, gives the corresponding reliability, which represents the performance measure for establishing the maturity function. Since we are using the same exponential model assumptions and failure rates, maturity calculation should follow analytically and graphically to the same results and observations found earlier of this example in section 4.4.3. The similarity in the analysis also includes optimizing the mission time t to reach certain reliability and maturity goals. According to the u-function representation, the maturity acceptability test based in this setting, however, is written by

$$I\big(R_i \geq threshold(R_i)\big) = I\left(\frac{d}{dz}U_{g_i}(1) \geq threshold\left(\frac{d}{dz}U_{g_i}(1)\right)\right)$$
$$= I\left(\frac{d}{dz}U_{g_i}(1) \geq r_i\right), for\ all\ i = 1, \dots, 5;\ mission\ time\ t = 1$$

Applying this procedure leads us to construct the following tests,

$$I\left(\frac{d}{dz}U_{g_1}(1) \geq r_1\right) = I\left(\frac{d}{dz}U_{g_2}(1) \geq r_2\right) = 1$$
$$I\left(\frac{d}{dz}U_{g_3}(1) \geq r_3\right) = I\left(\frac{d}{dz}U_{g_4}(1) \geq r_4\right) = I\left(\frac{d}{dz}U_{g_5}(1) \geq r_5\right) = 0$$

As a result,

$$\mathcal{M}_{ISMM} = \mathcal{M}_{UGF} = \left\{max(m)\ s.t. \left\{\prod_{i=1}^{m} I\left(\frac{d}{dz}U_{g_i}(1) \geq threshold\left(\frac{d}{dz}U_{g_i}(1)\right)\right)\right\} = 1\right\} = 2$$

5.6 Summary

The advantage of extending reliability analysis methods to solve the evaluation problem of operational capabilities of security systems has been established in Chapter 4. Such an extension allows one particularly to establish availability and reliability measures, and build accordingly quantitative maturity analysis.

In this chapter, we have introduced another evaluation approach to address the evaluation problem of security systems more universally, i.e., in multistate settings using multiple performance measures.

This approach is based on previous work of MSS systems and the UGF method. Using the same case study as in Section 3.11, we have demonstrated analytically and numerically how to establish structural evaluation of various permutations of intermediate *u*-functions towards the product *u*-function in a systematic and progressive manner. Being based on the same model inputs, this demonstration has also shown that both the reliability-theoretic method and the MSS UGF-based method lead to the same results, mutually validating each other. This method, however, allows one to design and audit security systems considering multistate controls and using a wider range of performance measures.

Chapter 6
Risk Assessment Using Asset-Control Bayesian Networks

6.1 Introduction

As discussed earlier, the development of system-level, quantitative security evaluation methods fills a gap in both research and industry. These methods can be used to build more-secure, reliable systems. The abstraction of a computing system into a set of assets and a set of controls represents a fundamental abstraction paradigm for this work. Previous chapters have approached the evaluation problem by modeling the relationships among the set of controls only. More specifically, Chapter 3 has shown how to build the logical arrangement of a security system into the ISMM model, capturing the relationships among its controls into RBDs, vectors, and structure functions. Then, Chapter 4 has shown how to build on such system representation by extending minimal path and cut sets methods and reliability analysis from reliability theory, as a means to evaluating operational capabilities of a security system and establish its maturity analysis. To make the modeling and evaluation method universal in terms of performance measures and controls states, Chapter 5 has extended the MSS UGF method into ISMM model, demonstrating both system representation and analysis aspects as well.

This chapter approaches the evaluation problem by examining the relationship of both sets of a computing system together: assets and controls. To establish the system model, we employ BNs to capture and bound the failure dependency among such entities. We then show how to use this representation to develop a new risk assessment method. An illustrative case study addressing risk mitigation under the cloud paradigm is presented.

The rest of this chapter is organized as follows. Section 6.2 describes the overall approach of the proposed risk assessment method. Then, the necessary notation and definition of this part are introduced in Section 6.3. Section 6.4 describes the asset-control BN as the candidate system model. Section 6.5 then explains the risk assessment method, showing both node- and system-level analysis equations, followed by features of this model in Section 6.6. The illustrative case study is presented in Section 6.7, showing its system description, model representation, and risk assessment. This work is further supported by two mathematical proofs in Section 6.8. The first is related to a specific bound on the risk function, and the other is on the distinction property between high-frequency low-impact and low-frequency high-impact events.

6.2 Approach

Computing paradigms are evolving by nature, leading to new forms of capabilities and challenges. The new paradigm of cloud computing, as an example, has led to new forms of complexity, uncertainty, and associated risks in security and privacy. It requires threat and risk system-level models that can quantitatively incorporate a cloud's resources, its offered services, and associated Service Level Agreement (SLA) into the assessment process. Nevertheless, regardless of advancements in the computing field, a system's components can still be intrinsically abstracted into two main classes: assets and controls. The interaction between these two classes creates a certain dependency with respect to failure that defines the security posture of the system, including its risk manageability. This work employs this abstraction and demonstrates how Bayesian Networks (BNs) can model such failure dependency to develop a new approach for risk assessment. This approach tackles three main challenges in current risk models. The first involves security failure quantification using the impact of failure statistics as opposed to underlying failure mechanisms, reducing the complexity of the failure space. The second resolves the distinction problem between high-frequency low-impact events and low-frequency high-impact ones by employing the casual effect property defined by BN topology. The third facilitates various useful inferences and types of analysis using BN formalism.

Traditional probabilistic risk assessment, however, employs both inductive and deductive approaches. Deductive methods are used to analyse underlying causes of an undesired event, whereas inductive methods are used to analyse the resulting effects and enumerate possible scenarios of undesired events [74]. The proposed modeling approach offers a new risk assessment method that bounds the relationships among undesired events using BNs. Being based on BNs, the method encompasses both inductive and deductive logic [142].

This work required the extension and integration of four studies, leading to the steps depicted in Figure 6-1: system-level abstraction into assets and controls, introduced in Section 3.4; the failure model from dependability theory, presented in Section 3.5; system model using Bayesian networks from graph theory, explained in Section 6.4; and risk assessment from security engineering, explained in section 6.5. The first three components of this approach have been published in [149], [150], [151] to introduce asset-control graph and evaluate failure dependency in access control models.

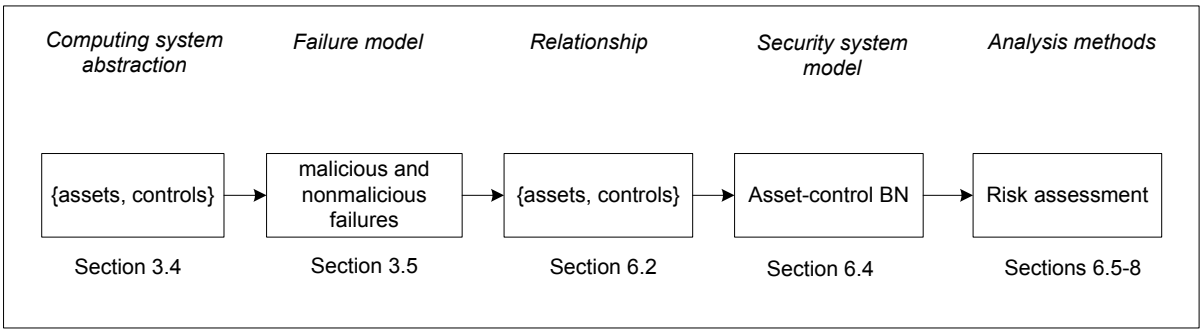

Figure 6-1: Approach of asset-control risk assessment

6.3 Notation and Definitions

Appendix B summarises the notation necessary to demonstrate this work. Recall the definitions of *assets* and *controls* in Section 3.3, and that this work is centered on analyzing the failure relationship among assets and controls when both are seen as the distinct abstracts of a computing system, as depicted in Figure 3-2. Furthermore, recall the failure definition in Section 3.5, which is the deviation from correct service, including both malicious and nonmalicious failures, with the level of abstraction based on the impact, not the underlying failure mechanisms.

Asset tag: A label associated with each asset to represent certain attributes of interest to the analysis. In this work, the tag of asset i is represented by

$$tag(A_i) = \big(Val(A_i), Imp(A_i)\big),$$

where the term $Val(A_i) \in \Re^+$ is the asset value, represented in countable units, say monetary units. Impact factor $Imp(A_i) \in \Re^+$ represents the extent of the damage of asset A_i in the case of failure, including tangible and intangible losses.

Control tag: A label associated with each control to represent its attributes of interest. The tag of control i is represented by

$$tag(C_i) = (Cst(C_i), Imp(C_i), Gol(C_i), Typ(C_i)),$$

where $Cst(C_i) \in \Re^+$ is the control cost[8], represented in the same unit as that used for asset value. The impact factor $Imp(C_i) \in \Re^+$ is similar to the impact factor of assets, since controls are part of the total assets too. The control process goal is represented by $Gol(C_i) \in \{Prevention\ PGol, Detection\ DGol, Recovery\ RGol\}$, and $control\ type\ Typ(C_i) \in \{Physical\ PTyp, Technical\ TTyp, Administrative\ ATyp\}$.

As with existing risk management practices, the information contained in these tags can be compiled during the early phases of the risk management process of the system of interest [10], [11]. In [10] for instance, costs of controls are calculated based on estimates of annual implementation costs, including worker times, hardware equipment, software packages, administrative overhead, and awareness programs.

6.4 Asset-control Bayesian Network

The main idea of asset-control BN is to use graph theory to capture probabilistically the topology of the system configuration and associated failure dependency among its components. To show this, consider a system $\boldsymbol{S}$ of n components, i.e., $\boldsymbol{S} = \{S_1, S_2, \dots, S_n\}$, where S_i is an asset or control, the mapping procedure into system model is established as follows.

$$\boldsymbol{S} \xrightarrow{maps\ to} \boldsymbol{X}, \boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}) \tag{6-1}$$

where,

1. $\boldsymbol{G}$ is a directed acyclic graph.
2. Vertices $\boldsymbol{V} = \{Assets\ \boldsymbol{A}, Controls\ \boldsymbol{C}\}$, represented by the set of random variables $\boldsymbol{X}$ that makes up the nodes of the network.
3. Edges $\boldsymbol{E} = \{failure\ dependency\}$ are defined by the failure dependency among BN nodes where a directed link reflects the probable impact of failure of the initial vertex on the terminal vertex, i.e., of a control or asset on another control or asset.
4. $\boldsymbol{P}$ is conditional probability distribution over $\boldsymbol{V}$, represented by the conditional probability tables (CPT). It basically quantifies the effect of the parents' failure on each node.

The goal of this mapping method is to bind, qualitatively and quantitatively, the failure behaviour, i.e., dependency and impact, among assets and controls using BN topology and CPTs, respectively.

[8] Note that C_i is control i. $Cst(C_i)$ is cost function of control i.

Moreover, additional feature space pertaining to various attributes of assets and controls is attached to graph nodes. These attributes are called asset and control tags and are specific to the evaluation method in use, which is the risk assessment in this work. Example attributes of assets are asset value $Val(A_i)$ and impact factor $Imp(A_i)$); and example attributes of controls are cost $Cst(C_i)$, impact factor $Imp(C_i)$, process goal $Gol(C_i)$, and control type $Typ(C_i)$.

The combined use of this BN-based modeling and the failure model makes a suitable match between the level of abstraction of failure and corresponding BN input parameter values. In particular, contrary to the intractable enumeration of all possible space of failure causes of system components (i.e., the full space of fault-error-failure chain), realisable failure effects are used in the model (i.e., failure statistics). In the Bayesian sense, these effects are modeled by BN random variables, which can be observed, hypothesized, or latent variables. As practiced in BN reasoning, initial failure prior probabilities of individual system components might come from historical data, operational field data, experiments, expert knowledge, and/or engineering estimates. Then, as new information arrives, nodes belief can be updated through their prior or posterior probabilities [165], making the subjective interpretation of probability in the Bayesian approach an attractive choice in the case of security [9].

In this work, we restrict ourselves to explaining the proposed evaluation method, excluding the structure and parameter learning problems of BNs. However, BN capabilities, including learning and reasoning techniques, are readily adoptable by the proposed method; therefore, their limitations are inherited too. The interested reader is referred to [32], [165] for various exact and approximate algorithms in BN learning.

Figure 6-2 depicts an example of a BN with 12 nodes of assets and controls. Controls $\{X_1, X_2, X_3\}$ are root nodes. The parents of asset X_6 are controls $\{X_1, X_4\}$. The children of asset X_6 are assets $\{X_9, X_{10}\}$. The descendants of asset X_6 are assets $\{X_9, X_{10}, X_{12}\}$. The nondescendants of asset X_6 are assets $\{X_7, X_8, X_{11}\}$ and controls $\{X_1, X_2, X_3, X_4, X_5\}$. The general flow of dependency relationships normally initiates from root controls protecting, and hence influencing, other control and/or asset nodes, which in turn, propagate through the remaining asset nodes, and eventually influence the system's threshold node. Note that it is not necessary that all controls influence all assets. Rather, the relationship is defined according to the protection/failure causal dependency; similarly, not all nodes, whether controls or assets, should influence the system node.

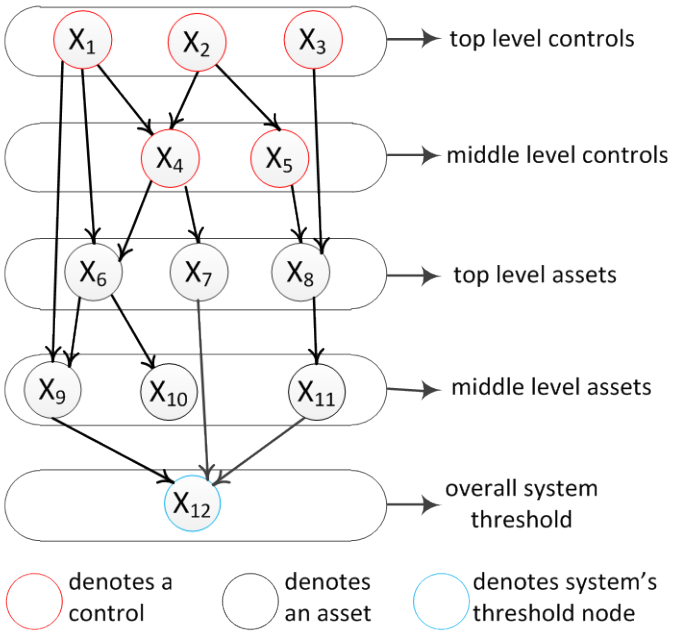

Figure 6-2: An example of an asset-control Bayesian network

6.5 Risk Assessment Method Using Asset-Control BNs

Risk is generally defined as the product of the likelihood of an event occurring and the impact that this event would have on an asset. The traditional method for measurement of computer-related risks is usually calculated by

$$ALE = ARO \times SLE, \qquad for\ each\ risk\ scenario, \tag{6-2}$$

which involves finding the quantities of threats, vulnerabilities, and corresponding assets and impacts [10]. This method, however, has been criticized for several reasons, among them, and most importantly, 1) its heavy reliance on likelihood quantities that are subjective and very hard to measure [19], [30]; 2) its inability to distinguish between high-frequency low-impact events and low-frequency high-impact ones[9] [10]; and 3) its inability to predict threats or failures [30].

The proposed approach, however, offers a different method for assessing risk, using asset-control BN. The asset-control BN, as the system model, captures inductively the failure dependency

[9] While in many cases high-frequency low-impact events can be manageable as incremental costs, low-frequency, high-impact events can be catastrophic [10].

relationship between assets and controls of a computing system. The risk assessment method then employs this representation to extract the likelihood and impact quantities necessary to establish quantitative risk assessment at both the component and system levels. Thus, this method provides two groups of equations addressing various aspects of risk: the first addresses risk calculations at the node level, and the second addresses risk calculations at the system level. Both groups employ the BN topology, CPTs, BN reasoning, and asset and control tags to quantify risks in question.

Generally, each risk calculation in this method involves the multiplication of the quantities $P(.), Imp(.)$, and $Val(.)$. The likelihood term ARO in (6-2) is found by the probability $P(.)$, which is derived from BN. The impact term SLE is found by $Imp(.) \times Val(.)$, given by the asset and control tags. The quantities $Imp(.)$ and $Val(.)$, however, can still be established using common techniques in current risk assessment methods.

6.5.1 Node-level Equations

The objective of this section is to show how one can compute node-level risk for three different forms of scenarios: 1) node risk, 2) node risk with evidence, and 3) node risk with exclusions.

To demonstrate the first form, let $ALE(X_i)$ denote the annual loss expectancy of node, or component, i. Recall that failure behaviour for any node (including impacted nodes) is now bounded qualitatively and quantitatively by BN topology and CPT, respectively. Using this representation, we define $ALE(X_i)$ as the risk quantity resulting from the failure of node X_i and its consequent failures, and it is calculated by[10]

$$\begin{aligned} ALE(X_i) &= ALE_n(X_i) + ALE_n\{dec(X_i)/X_i\} \\ &= ALE_n(X_i) + \sum_{all\ X_j \in dec(X_i)} ALE_n(X_j/X_i) \end{aligned} \tag{6-3}$$

where

$$ALE_n(X_i) = \begin{cases} P(x_i) \times Imp(X_i) \times Val(X_i), & if\ X_i\ is\ an\ asset, \\ P(x_i) \times Imp(X_i) \times Cst(X_i), & if\ X_i\ is\ a\ control. \end{cases} \tag{6-4}$$

and

[10] We use $P\{X_i = "success\ state"\} = P(x_i)$, where success state means failure event in this context.

$$ALE_n(X_j/X_i) = \begin{cases} P(x_j/x_i) \times Imp(X_j) \times Val(X_j), & if\ X_j\ is\ an\ asset, \\ P(x_j/x_i) \times Imp(X_j) \times Cst(X_j), & if\ X_j\ is\ a\ control. \end{cases} \quad (6\text{-}5)$$

The first term in (6-3), $ALE_n(X_i)$, represents the direct risk quantity from X_i failure, and is calculated by (6-4). The probability $P(x_i)$ is derived from the CPT, and the quantities $Imp(X_i)$ and $Val(X_i)$ (or $Cst(X_i)$) are given by the node tag. The second term, $ALE_n\{dec(X_i)/X_i\}$, represents the consequent risk quantity from X_i failure, which is bounded by BN topology, and is calculated using (6-5). Similarly, the conditional probability $P(x_j/x_i)$ is derived from the CPT, and the quantities $Imp(X_j)$ and $Val(X_j)$ (or $Cst(X_i)$) are given by the corresponding node tag. Thus, risk is generally represented by the direct risk of the node in question plus the consequent risk from its descendants. The latter term is sometimes called descendant risk.

The cases in (6-4) and (6-5) represent similar calculations, but $Val(.)$ is used to denote that the node in question is an asset, whereas $Cst(.)$ is used to denote that the node is essentially a control. Henceforth, $Val(.)$ is used to represent both cases.

The second form of queries involves inference of risk given some evidence of failure (or non-failure) as follows. To find the annual loss expectancy of X_i conditioned on particular evidence, say X_j's occurrence, we write

$$\begin{aligned} ALE(X_i/X_j) &= ALE_n(X_i/X_j) + ALE_n\{dec(X_i)/X_iX_j\} \\ &= P(x_i/x_j) \times Imp(X_i) \times Val(X_i) \\ &+ \sum_{all\ X_k \in dec(X_i)} ALE_n(X_k/X_iX_j) \end{aligned} \quad (6\text{-}6)$$

where

$$ALE_n(X_k/X_iX_j) = P(x_k/x_ix_j) \times Imp(X_k) \times Val(X_k) \quad (6\text{-}7)$$

The first term in (6-6), $ALE_n(X_i/X_j)$, can be computed using (6-5), and the second term, $ALE_n\{dec(X_i)/X_iX_j\}$ is found using (6-7). This form allows us to calculate the risk when we have evidence about some node X_j, i.e., failure or non-failure, affecting the node in question, i.e., X_i. Therefore, the conditioning on X_j starts at the top node and propagates downward along with X_i to the impacted nodes, i.e., its descendants, as well. For risk diagnosis, the inference is made from children to parents, and for risk prediction, the inference is made from parents to children.

The third form involves exclusions as follows. To find the annual loss expectancy of node i, excluding a certain set of nodes $\boldsymbol{X}_{-i}$, we write[11]

$$\begin{aligned} ALE(X_i \backslash \boldsymbol{X}_{-i}) &= ALE_n(X_i) + ALE_n\{dec(X_i) \backslash \boldsymbol{X}_{-i}/X_i\} \\ &= P(x_i) \times Imp(X_i) \times Val(X_i) \\ &+ \sum_{all\ (X_j \backslash \boldsymbol{X}_{-i}) \in dec(X_i)} ALE_n(X_j/X_i) \end{aligned} \tag{6-8}$$

Similarly, the first term in (6-8), $ALE_n(X_i)$, can be found using (6-4), and the second term, $ALE_n\{dec(X_i) \backslash \boldsymbol{X}_{-i}/X_i\}$, can be found using (6-5). This form of queries can be used to evaluate alternative design options and mitigation measures to control risk. For example, it can be used to measure the residual risk in the case of moving some applications or platforms into the cloud.

Note that both conditioning and exclusion keys, represented in (6-6) and (6-8), respectively, can be combined into the same risk query.

Furthermore, two remarkable bounds of the risk associated with the system threshold in the case of evidence of its failure, $ALE(X_s/X_s)$[12], can be established with respect to its parents $pa(X_s)$: 1) it is lower than any of its diagnosis-based risks, $ALE(X_i/X_s)$, i.e.,

$$ALE(X_s/X_s) < ALE(X_i/X_s), \qquad for\ all\ X_i \in pa(X_s); \tag{6-9}$$

and, 2) it is higher than (or equal to) any of its prediction-based risks, $ALE(X_s/X_i)$, i.e.,

$$ALE(X_s/X_s) \geq ALE(X_s/X_i), \qquad for\ all\ X_i \in pa(X_s). \tag{6-10}$$

The intuitions behind these two bounds make consistent arguments with their calculations. In (6-9), when the failure of a system's threshold node is realised (i.e., is not a probabilistic figure anymore), its direct SLE is incurred. Thus, finding the likelihood of parents' risk given this evidence (i.e., $ALE(X_i/X_s)$) will include the probable conditional risk terms of those parent nodes in addition to the certain risk term of the system threshold. In (6-10), the risk associated with the probable system threshold failure in the evidence of failure of any of its parents is less than (or equal to) the risk associated with the system threshold certain failure (i.e., when $P(x_s) = 1$). The equality in this bound occurs when $P(x_s/x_i) = 1$. The mathematical proof for these two bounds is provided later in Section 6.8.

[11] We use backslash to denote the exclusion part of the argument of $ALE(.)$ function.

[12] $ALE(X_s/X_s)$ is different than $ALE(X_s)$. The former represents system threshold risk in the evidence of system-level failure, i.e., $P(x_s) = 1$, whereas the later represents system threshold risk when failure is probable.

6.5.2 System-level Equations

The objective of this section is to show how to compute system-level risk for three different tasks: 1) a system's threshold risk, 2) a system's direct risk (i.e., risk of individual nodes alone, not including descendants' risks), and 3) a system's total risk paths (i.e., both direct and descendant risk quantities).

The first task is finding the risk of the system reaching its threshold failure point. This task is denoted as $ALE(X_s)$ and defined by the calculated risk of the system's leaf node, i.e.,

$$\begin{aligned} ALE(X_s) &= ALE_n(X_s) \\ &= ALE_n(X_i) \\ &= P(x_i) \times Imp(X_i) \times Val(X_i) \end{aligned} \tag{6-11}$$

where X_i in this case is the leaf node that represents the system threshold. Note that the probability $P(x_i)$, by definition of BN, absorbs the probabilities of nodes that have a risk path toward system-level failures.

The second task for a system is finding the total number of direct risk quantities of its individual nodes. This task is denoted as $ALE_n(\boldsymbol{X})$ and defined by

$$\begin{aligned} ALE_n(\boldsymbol{X}) &= \sum_{all\ X_i \in \boldsymbol{X}} ALE_n(X_i) \\ &= ALE_n(X_1) + ALE_n(X_2) + \cdots + ALE_n(X_s) \end{aligned} \tag{6-12}$$

This equation sums up the risks associated with individual nodes of the system, not including their descendants' risks (i.e., consequent risks).

The third task is finding the number of total risk paths of the system. This task is denoted as $ALE(\boldsymbol{X})$ and calculated by

$$\begin{aligned} ALE(\boldsymbol{X}) &= \sum_{all\ X_i \in \boldsymbol{X}} ALE(X_i) \\ &= ALE(X_1) + ALE(X_2) + \cdots + ALE(X_s) \end{aligned} \tag{6-13}$$

This equation sums up the risks associated with individual nodes of the system, including their descendants' risks.

Similarly to node-level queries, both conditioning and exclusion keys can be used in system-level queries. It is also remarkable that $ALE(.)$ is a monotonic increasing function of the values $P(.), Imp(.)$, and $Val(.)$, and is additive over its nodes.

6.6 Features

The proposed risk assessment offers a plausible method for assigning numeric values to the likelihood and impact of risk and to the costs and benefits of alternative configurations of the system.

In light of the criticisms of traditional risk assessment methods mentioned earlier, we present three features of this approach. First, contrary to traditional risk models, this formalism extends our system abstraction and failure model, defined earlier, to BN representation. This combination makes the domain of $P(X_i)$ inclusive of malicious and nonmalicious incidents of the likelihood statistics of threats and vulnerabilities summarized into the links going to X_i, with the cascaded effect of failure summarized in the links outgoing from X_i. In this formalism, BN topology is employed in defining failure and its cascaded impact, thereby bounding the qualitative part of the risk behavior. CPTs are employed in deriving associated likelihood and impact calculations, thus bounding the quantitative part of the risk behavior. As a result, this bounding technique addresses the difficulty associated with traditional risk assessment methods with respect to calculating likelihood quantities for each pair of a threat and an asset in the system individually, which are subjective and very hard to measure. It also incorporates the operational dynamics of controls in the analysis as more failure data feeds lead to updating BN prior and posterior probabilities.

Second, in addition to calculating the direct risk quantity of a given node, say X_i (by computing $ALE_n(X_i)$), consequent risk quantities are added as defined by BN topology (by computing $ALE_n\{dec(X_i)/X_i\}$). The addition of other risk terms as a result of the risk of the initial node addresses the distinction problem between high-frequency low-impact events and low-frequency high-impact ones. Further, it shows that the probabilities of failure and risk calculations associated with high-frequency low-impact events are lower than the corresponding figures in the opposite scenario. This property builds an agreement with the observation that, while in many cases high-frequency low-impact events can be manageable as incremental costs, low-frequency high-impact events can be catastrophic [10]. The mathematical proof of this property is presented later in Section 6.8.

Third, the adoption of BNs inherently facilitates consistent reasoning, inferences, and prediction of threats, failures, and associated risks. This facilitation comes naturally with existing, well-founded algorithms and tools for such BN primitives.

6.7 Case Study

In this section, we demonstrate the proposed risk assessment method using a common scenario influenced by the transition toward cloud services. However, we emphasize that the proposed concepts hold and the method is applicable to computing systems in general.

First, we describe the scenario, and then demonstrate how to build the corresponding BN representation, followed by a brief demonstration of simple risk assessment exercises. We employed the Junction Tree inference algorithm using BN Toolbox (BNT) in MATLAB [166] to calculate BN inferences.

6.7.1 System Description

Consider the scenario in Figure 6-3. It basically represents a small web-based application where access control is implemented by an authentication server for computer security and by a biometric reader over a server farm for physical security. Assume that the failure relationship between assets and controls is summarized by the statement that the failure of either the authentication server S_1 or the biometric reader S_2 leads to failure of the web application S_3, which in turn, might lead to failure of system operations overall. In this scenario $\boldsymbol{S} = \{S_1, S_2, S_3\}$, and we call it Scenario 1.

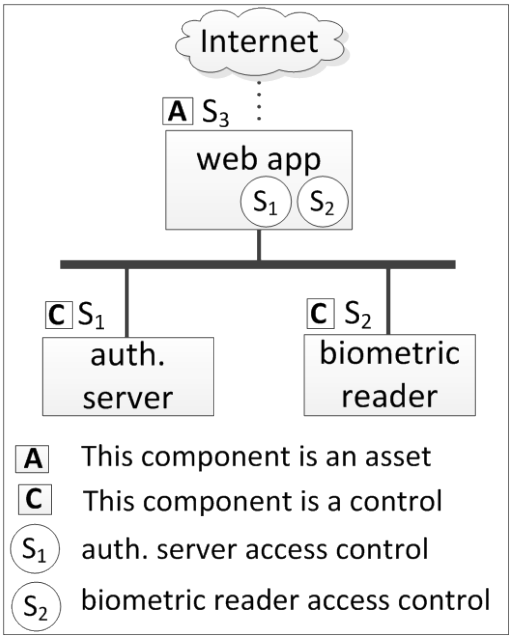

Figure 6-3: Scenario 1: outline of asset and control entities[13]

As an alternative design and a risk mitigation measure, the management wants to assess the risk transference option associated with moving the web-based application into the cloud. This option represents an application of the Software as a Service (SaaS) model of the cloud paradigm. With this risk mitigation measure, assume that there will be no need to keep the authentication server S_1 or the biometric reader S_2 anymore. To avoid confusion of notation, we add the subscript "c" to the notation under the cloud configuration, so, in this scenario $\boldsymbol{S_c} = \{S_{3c}\}$, and we call it Scenario 2. We also assume that all other parameters for these scenarios are fixed. This option is depicted in Figure 6-4. Note that we intentionally chose a small number of nodes to demonstrate the method and simplify the analysis of the web-based application's options. However, these nodes can be a subset of a larger set of nodes representing a complete computing environment, where the analysis will be more complex but should follow similarly.

[13] Note that squares denote component's class, and circles encode asset-control failure relationships (i.e., the failure of either S_1 or S_2 leads to failure of S_3).

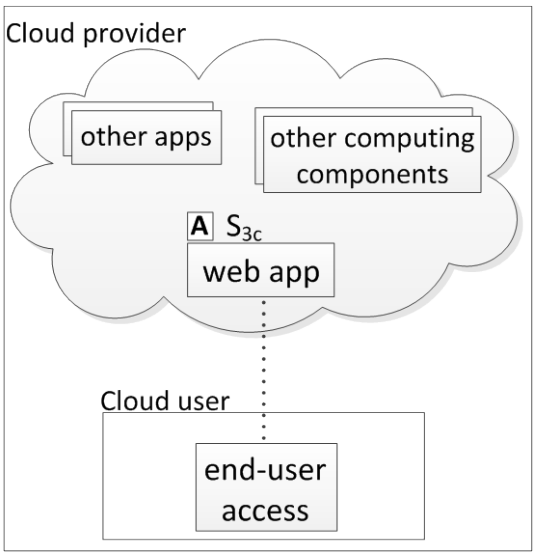

Figure 6-4: Scenario 2: cloud SaaS model[14]

6.7.2 Model Representation Using Asset-Control BN

First, we consider building the BN of the original configuration depicted in Figure 6-3. The corresponding BN topology is mapped directly using the qualitative failure behavior assumption defined earlier. Doing so leads us to define the BN conditional independence statements among system components, as depicted in Figure 6-5. To demonstrate some quantitative assessment, assume binary failure events of components, represented by binary random variables, which are mutually exclusive and collectively exhaustive states of the probability of the failure space; assume the knowledge of the conditional probability tables (CPTs) as compiled in Table 6-1; and also assume the knowledge of tags information as compiled in Table 6-2.

As a result, $\boldsymbol{S}$ is modeled by $\boldsymbol{X}, \boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, and $\boldsymbol{P}$ as follows:

1. $\boldsymbol{G}$ is a directed acyclic graph.
2. $\boldsymbol{X} = \{X_1, X_2, X_3, X_4\}$, Vertices $\boldsymbol{V} = \{Assets\ \boldsymbol{A}, Controls\ \boldsymbol{C}\}$, where

   $\boldsymbol{A} = \{X_3, X_4\}$, $\boldsymbol{C} = \{X_1, X_2\}$.
3. Edges $\boldsymbol{E} = \{failure\ dependency\}$, where

[14] The web-based application is moved into the cloud, including its protection cost and associated risks. Thus, S_1 and S_2 are eliminated from the configuration.

$$\boldsymbol{E} = \{(X_1, X_3), (X_2, X_3), (X_3, X_4)\}.$$

4. P is conditional probability distribution over V, represented by CPTs in Table 6-1.

Note that the overall system X_4 is a leaf node, a virtual one used to represent the collective failure of system components when these are considered as a whole. This representation can be useful to define the paths and threshold point that can lead to determining the system to be in a failure state.

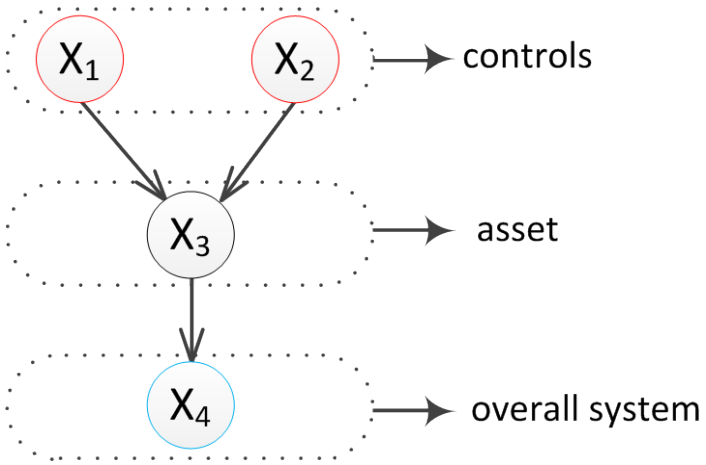

Figure 6-5: Scenario 1: BN representation[15]

[15] Note that X_4 is added to model the failure threshold of the system.

Table 6-1: Example data for conditional probability tables of the BN

X_1

Y	N
0.05	0.95

X_2

Y	N
0.15	0.85

X_3/X_1X_2

		X_3	
X_1	X_2	Y	N
Y	Y	0.40	0.60
Y	N	0.10	0.90
N	Y	0.22	0.78
N	N	0.05	0.95

X_4/X_3

	X_4	
X_3	Y	N
Y	0.23	0.77
N	0.08	0.92

Table 6-2: Scenario 1: example data for asset and control tags of the BN

Node	Description	Type	Tags: $tag(A_i) = (Val(A_i), Imp(A_i))$ $tag(C_i) = \big(Cst(Ci), Imp(C_i), Gol(C_i), Typ(C_i)\big)$
X_1	authentication server	Control	$(12, 0.46, PGol, TTyp)$
X_2	biometric reader	Control	$(11, 0.50, PGol, PTyp)$
X_3	web application	Asset	$(30, 0.80)$
X_4	overall system	Asset	$(55, 1.20)$

Alternatively, when the cloud solution is considered, the risks associated with the web application S_3, the authentication server S_1, and the biometric reader S_2 are understood to have been transferred to the cloud. Let X_{3c} be a r.v. representing the state of failure of the aggregation of the web application and associated controls under the cloud, and X_{4c} be a r.v. representing the new system threshold state of failure. The corresponding BN is depicted in Fig. 7, and $\boldsymbol{S_c}$ is modeled by $\boldsymbol{X_c, G_c} = (\boldsymbol{V_c, E_c})$, and $\boldsymbol{P_c}$ as follows:

1. $\boldsymbol{G_c}$ is a directed acyclic graph.

2. $\boldsymbol{V_c} = \boldsymbol{X_c} = \{X_{3c}, X_{4c}\}$.
3. $\boldsymbol{E_c} = \{(X_{3c}, X_{4c})\}$.
4. $\boldsymbol{P_c}$ is CPT over $\boldsymbol{V_c}$

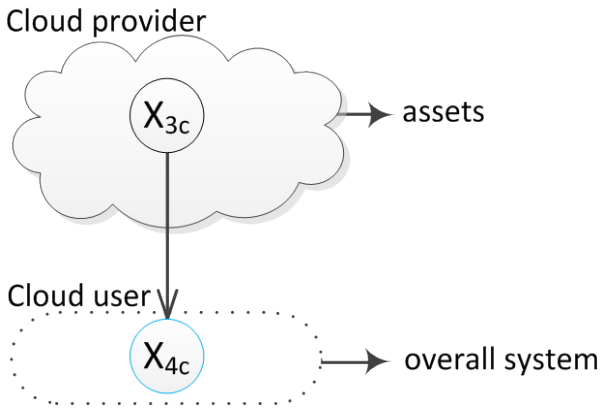

Figure 6-6: Scenario 2: BN representation[16]

For consistency of the analysis, we further assume that $tag(X_{4c}) = tag(X_4)$ and $Imp(X_{3c}) = Imp(X_3)$, while the quantities $Val(X_{3c})$, $P(x_{3c})$, $P(x_{4c}/x_{3c})$, $P(x_{4c}/\overline{x}_{3c})$, and $P(x_{4c})$ remain in question. These quantities represent key decision points when determining the feasibility of the alternative solution. They can be used as input figures for the Quality of Service (QoS) metrics and SLA protocols between the cloud provider and user. Therefore, we do not intend to assign hypothetical values for these variables; rather, we present them in the analysis as decision bounds with respect to the calculations performed on Scenario 1.

6.7.3 Risk Assessment and Analysis

We briefly explore some demonstration queries. We first demonstrate the calculations based on Scenario 1, followed by comparisons with ideal bounds for Scenario 2. The ideal region in this context is defined for cases in which the cloud configuration could outperform the original configuration operationally and economically, i.e., where failure probability and risk measures of X_{3c} equal or exceed the corresponding measures of the worst-case scenario among X_1, X_2, and X_3.

[16] Note that X_{4c} is added to model the new failure threshold of the system.

Node-level analysis: Recall that the first form of node-level risk calculations is finding $ALE(X_i)$. For Scenario 1, to calculate the risk (i.e., annual loss expectancy) associated with the failure of the biometric reader X_2, we write[17]

$$\begin{aligned} ALE(X_2) &= ALE_n(X_2) + \sum_{all\ X_j \in dec(X_2)} ALE_n(X_j/X_2) \\ &= P(x_2) \times Imp(X_2) \times Val(X_2) \\ &+ \mathrm{P}(x_3/x_2) \times Imp(X_3) \times Val(X_3) \\ &+ \mathrm{P}(x_4/x_2) \times Imp(X_4) \times Val(X_4) \\ &= (0.15 \times 0.50 \times 11) \\ &+ (0.229 \times 0.80 \times 30) \\ &+ (0.114 \times 1.20 \times 55) \\ &= 14\ units. \end{aligned}$$

Based on this result, we can say that there is a probability of losing a prevention access control function implemented by the physical control X_2 of $P(x_2) = 0.15$. This probable event might cause damage of about $ALE(X_2) = 14$ units, whereby the control current direct investment, or cost, $Cst(C_2) = 11$ units, designed to contribute to the protection of assets[18] of a total value of

$$Val\big(dec(X_2)\big) = 30 + 55 = 85\ units.$$

Applying the same calculations on the authentication server, we find that there is a probability of losing a prevention access control function implemented by the technical control X_1 of $P(x_1) = 0.05$. This probable event might cause damage of $ALE(X_1) = 10$ units, whereby the control current direct cost $Cst(C_1) = 12$ units, designed to contribute to the protection of the same assets of a total value of 85 units.

In addition, the probability of failure of the web application $P(x_3) = 0.08$, with a probable damage $ALE(X_3) = 17$ units, whereby the asset direct value $Val(X_3) = 30$ units, impacting another asset of a total value of 55 units.

[17] Risk calculations are rounded.

[18] As controls can themselves be assets, we use "assets" to denote direct system assets, i.e., $\{X_3, X_4\}$, of a total value of 85 units, and we use "total assets" to denote all assets, including controls, i.e., $\{X_1, X_2, X_3, X_4\}$, of a total value of 108 units.

The second form of node-level calculations is evidence-based risk. For example on risk diagnosis, in the case of evidence of failure of the system threshold X_s, the probability that the cause was due to the biometric reader is found by $P(x_2/x_s) = 0.187$, and the associated risk is calculated by

$$\begin{aligned} ALE(X_2/\mathrm{X}_s) &= ALE_n(X_2/\mathrm{X}_s) + \sum_{all\ X_k \in dec(X_2)} ALE_n(X_k/X_2\mathrm{X}_s) \\ &= P(x_2/x_4) \times Imp(X_2) \times Val(X_2) \\ &+\mathrm{P}(x_3/x_2x_4) \times Imp(X_3) \times Val(X_3) \\ &+\mathrm{P}(x_4/x_2x_4) \times Imp(X_4) \times Val(X_4) \\ &= (0.187 \times 0.50 \times 11) \\ &+(0.461 \times 0.80 \times 30) \\ &+(1 \times 1.20 \times 55) \\ &= 78\ units. \end{aligned}$$

In contrast, the probability that the system failure X_s was due to a failure initiated at the authentication server X_1 is given by $P(x_1/x_s) = 0.055$, with the accumulated risk $ALE(X_1/X_s) = 74$ units. However, the probability of failure of the web application X_3 given the evidence of system failure is $P(x_3/x_s) = 0.20$, with the associated risk $ALE(X_3/X_s) = 71$ units.

For example on risk prediction, in the case of evidence of failure of the biometric reader X_2, the probability of system failure $P(x_s/x_2) = 0.114$, and the associated risk is calculated by[19]

$$\begin{aligned} ALE(X_s/\mathrm{X}_2) &= ALE_n(X_s/\mathrm{X}_2) + \sum_{all\ X_k \in dec(X_s)} ALE_n(X_k/X_s\mathrm{X}_2) \\ &= P(x_s/x_2) \times Imp(X_s) \times Val(X_s) + \mathrm{ALE}_n\{\emptyset\} \\ &= (0.114 \times 1.20 \times 55) + 0 \\ &= 8\ units. \end{aligned}$$

In contrast, the probability of system failure X_s, given the evidence of failure of the authentication server X_1, is given by $P(x_s/x_1) = 0.102$, with the accumulated risk $ALE(X_s/X_1) = 7$ units. Nonetheless, the probability of system failure given the evidence of failure of the web application X_3 is $P(x_s/x_3) = 0.23$, with the associated risk $ALE(X_s/X_3) = 15$ units.

[19] Note that the second term in the equation here equals 0 because the prediction is made about a leaf node; otherwise, the descendants will not be the empty set, and therefore, additional consequent risks will be incurred.

The third form of node-level calculations involves risk exclusions. To further analyze the risk associated with the biometric reader $ALE(X_2)$ when eliminating the risks associated with the web application $ALE(X_3)$ and the overall system failure $ALE(X_4)$ from the risk path of the biometric reader X_2, we compute

$$\begin{aligned} ALE(X_2 \backslash \mathrm{X}_3\mathrm{X}_4) &= ALE_n(X_2) + \sum_{all\ (X_j \backslash \mathrm{X}_3\mathrm{X}_4) \in dec(X_2)} ALE_n(X_j/\mathrm{X}_2) \\ &= P(x_2) \times Imp(X_2) \times Val(X_2) + \mathrm{ALE}_n\{\emptyset\} \\ &= (0.15 \times 0.50 \times 11) + 0 \\ &= 1\ unit. \end{aligned}$$

This result suggests a significant decrease of the risk associated with the biometric reader $ALE(X_2)$ by 92.86%, its diagnosis-based risk $ALE(X_2/X_s)$ by 98.72%, and its prediction-based risk $ALE(X_s/X_2)$ by 87.50% if more protection (e.g., an alternative design) is implemented to eliminate only the risk associated with the web application X_3 from the risk path of X_2, as X_3 already cuts off the link (i.e., consequent risk) to X_4.

Similarly, $ALE(X_1 \backslash X_3X_4) = 0.30$ unit, which shows a decrease of the risk associated with the authentication server $ALE(X_1)$ by 97%, its diagnosis-based risk $ALE(X_1/X_s)$ by 99%, and its prediction-based risk $ALE(X_s/X_1)$ by 95.71% if only the web application's risk is eliminated. In addition, $ALE(X_3 \backslash X_4) = 2$ units, which shows a decrease of the risk associated with the web application $ALE(X_3)$ by 88.24%, its diagnosis-based risk $ALE(X_3/X_s)$ by 97.18%, and its prediction-based risk $ALE(X_s/X_3)$ by 86.67% if the web application's consequent risk is eliminated.

To evaluate the bounds of the realised risk of system threshold when it fails $ALE(X_s/X_s)$, proofed in the next section, we compute the quantity

$$\begin{aligned} ALE(X_s/\mathrm{X}_s) &= P(x_4/x_4) \times Imp(X_4) \times Val(X_4) \\ &= 1 \times 1.20 \times 55 \\ &= 66\ units. \end{aligned}$$

As shown, the bounds of $ALE(X_s/\mathrm{X}_s)$ hold with respect to its parents for both diagnosis-based risks, i.e., $ALE(X_1/\mathrm{X}_s), ALE(X_2/\mathrm{X}_s)$, and $ALE(X_3/\mathrm{X}_s)$; and prediction-based risks, i.e., $ALE(X_s/\mathrm{X}_1), ALE(X_s/\mathrm{X}_2)$, and $ALE(X_s/\mathrm{X}_3)$.

With respect to maintaining the security controls locally (Scenario 1), these results suggest that the biometric reader has a higher probability of cause of failure to the system and a higher magnitude of risk, thus significance, compared to the authentication server. However, it also shows that the consequent risk quantities of both controls and the web application, due to their interdependency with system failure, represent the biggest contribution to system risk overall.

To evaluate the plausibility of Scenario 2, the above results are summarized in Table 6-3 with the corresponding ideal region under the cloud configuration. These statistics show the anticipated ideal bounds of the probability of failure $P(x_{3c})$, investment $Val(X_{3c})$, and associated risk quantity $ALE(X_{3c})$, as well as diagnosis-, prediction-, and exclusion-based probabilities and risks.

Table 6-3: Node-level risk analysis: original vs. cloud configuration[20]

	measure	original configuration			cloud configuration
		X_1	X_2	X_3	X_{3c} ideal bound
node risk	$Cst(X_i)\ or\ Val(X_i)$	12	11	30	$Val(X_{3c}) \leq (12+11+30)$
	$P(x_i)$	0.05	0.15	0.08	$P(x_{3c}) \leq 0.05$
	$ALE(X_i)$	10	14	17	$ALE(X_{3c}) \leq 10$
diagnosis-based node risk	$P(x_i/x_s)$	0.055	0.187	0.20	$P(x_{3c}/x_{sc}) \leq 0.055$
	$ALE(X_i/X_s)$	74	78	71	$ALE(X_{3c}/X_{sc}) \leq 71$
prediction-based node risk	$P(x_s/x_i)$	0.102	0.114	0.23	$P(x_{sc}/x_{3c}) \leq 0.102$
	$ALE(X_s/X_i)$	7	8	15	$ALE(X_{sc}/X_{3c}) \leq 7$
exclusion-based node risk	$P(x_i)$	0.05	0.15	0.08	$P(x_{3c}) \leq 0.05$
	$ALE(X_i \backslash dec(X_i))$	0.30	1	2	$ALE(X_{3c} \backslash X_{4c}) \leq 0.30$

System-level analysis: The analysis can be further augmented by performing some comparisons to system-level risk statistics. Recall that the first task is finding $ALE(X_s)$, which represents the risk of a system reaching its threshold failure point. For Scenario 1, this task is defined at node X_4 and computed by[21]

[20] Note that the column of cloud configuration shows the region of values as opposed to specific values, indicating where the cloud configuration could outperform the original one considering the cloud's best-case scenario.

[21] In the context of this case study, X_4 and X_s are used interchangeably, as well as X_{4c} and X_{sc}.

$$
\begin{aligned}
ALE(X_s) &= ALE_n(X_4) \\
&= P(x_4) \times Imp(X_4) \times Val(X_4) \\
&= 0.092 \times 1.20 \times 55 \\
&= 6\ units.
\end{aligned}
$$

The second task of system-level calculations is finding the system's direct risk, $ALE_n(\boldsymbol{X})$, which can be found as follows

$$
\begin{aligned}
ALE_n(\mathbf{X}) &= \sum_{all\ X_i \in \boldsymbol{X}} ALE_n(\mathrm{X}_i) \\
&= ALE_n(\mathrm{X}_1) + ALE_n(\mathrm{X}_2) + ALE_n(\mathrm{X}_3) + ALE_n(\mathrm{X}_4) \\
&= 9\ units.
\end{aligned}
$$

The third task is finding the system's total risk paths (i.e., direct and consequent risk quantities), $ALE(\boldsymbol{X})$, which can be found as follows

$$
\begin{aligned}
ALE(\boldsymbol{X}) &= \sum_{all\ X_i \in \boldsymbol{X}} ALE(X_i) \\
&= ALE(X_1) + ALE(X_2) + ALE(X_3) + ALE(X_4) \\
&= 47\ units.
\end{aligned}
$$

Additionally, some ratio statistics can be added to the analysis. For example, to find the ratio of risk contribution of the biometric reader to the overall anticipated risk of the system, for a system's direct risk, we compute

$$\frac{ALE_n(\mathrm{X}_2)}{ALE_n(\mathbf{X})} = \frac{1}{9} = 11.11\%,$$

and for overall system risk paths, we write

$$\frac{ALE(\mathrm{X}_2)}{ALE(\mathbf{X})} = \frac{14}{47} = 29.79\%.$$

The ratio of risk contribution of the authentication server to the overall anticipated risk of the system, for system direct risk, we compute

$$\frac{ALE_n(\mathrm{X}_1)}{ALE_n(\mathbf{X})} = \frac{0.30}{9} = 3.33\%,$$

and for overall system risk paths, we write

$$\frac{ALE(\mathrm{X}_1)}{ALE(\mathbf{X})} = \frac{10}{47} = 21.28\%.$$

Similarly, the ratio of risk contribution of the web application to the overall anticipated risk of the system, for system direct risk, we compute

$$\frac{ALE_n(\mathrm{X}_3)}{ALE_n(\mathbf{X})} = \frac{2}{9} = 22.22\%,$$

and for overall system risk paths, we write

$$\frac{ALE(\mathrm{X}_3)}{ALE(\mathbf{X})} = \frac{17}{47} = 36.17\%.$$

Adding up these risk quantities to see how much risk is being transferred to the cloud, for system direct risk, leads to

$$\begin{aligned}\frac{\sum_{transferable\ nodes} ALE_n(\mathrm{X}_i)}{ALE_n(\mathbf{X})} &= \frac{ALE_n(\mathrm{X}_1) + ALE_n(\mathrm{X}_2) + ALE_n(\mathrm{X}_3)}{ALE_n(\mathbf{X})} \\ &= \frac{1 + 0.30 + 2}{9} \\ &= 36.70\%,\end{aligned}$$

and for total risk paths, we find

$$\begin{aligned}\frac{\sum_{transferable\ nodes} ALE(\mathrm{X}_i)}{ALE(\mathbf{X})} &= \frac{ALE(\mathrm{X}_1) + ALE(\mathrm{X}_2) + ALE(\mathrm{X}_3)}{ALE(\mathbf{X})} \\ &= \frac{14 + 10 + 17}{47} \\ &= 87\%.\end{aligned}$$

To find the ratio of the total investment of system protection over the value of its total assets, we compute

$$\begin{aligned}\frac{Cst(\mathbf{C})}{Val(\mathbf{X})} &= \frac{\sum_{all\ C_j} Cst(C_j)}{\sum_{all\ X_i} Val(X_i)} \\ &= \frac{23}{108} = 21.30\%,\end{aligned}$$

which provides the system $\boldsymbol{S}$ with a protection level up to the probability of failure:

$$P(x_s) = 0.0918.$$

Table 6-4 summarises some of the above statistics at system level, indicating the cloud's ideal region of bounds based on the calculated risk transference quantities.

Table 6-4: System-level risk analysis: original vs. cloud configuration[22]

	measure	original configuration $\boldsymbol{X}$	cloud configuration $\boldsymbol{X_c}$ ideal bound
investment (total assets)	$\sum_{all\ X_i} Val(X_i)$	108	$\sum_{all\ X_{ic}} Val(X_{ic}) \leq 108$
protection (controls)	$\sum_{all\ C_j} Cst(C_j)$	23	$\sum_{all\ C_{jc}} Cst(C_{jc}) \leq 23$
controls to total assets ratio	$\frac{\sum_{all\ C_j} Cst(C_j)}{\sum_{all\ X_i} Val(X_i)}$	$\frac{23}{108} = 21.30\%$	$\frac{\sum_{all\ C_{jc}} Cst(C_{jc})}{\sum_{all\ X_{ic}} Val(X_{ic})} \leq 21.30\%$
threshold failure probability	$P(x_s)$	0.0918	$P(x_{sc}) \leq 0.0918$
threshold risk	$ALE(X_s)$	6	$ALE(X_{sc}) \leq 6$
direct risk	$ALE_n(\boldsymbol{X})$	9	$ALE_n(\boldsymbol{X_c}) \leq 9$
total risk paths	$ALE(\boldsymbol{X})$	47	$ALE(\boldsymbol{X_c}) \leq 47$
transferable investment	$\sum_{transferable\ nodes} Val(X_i)$	$12 + 11 + 30 = 53$	$Val(X_{3c}) \leq 53$
direct risk ratio: transferable to overall nodes	$\frac{\sum_{transferable\ nodes} ALE_n(X_i)}{ALE_n(\boldsymbol{X})}$	36.70%	$\frac{ALE_n(X_{3c})}{ALE_n(\boldsymbol{X_c})} \leq 36.70\%$
total risk paths ratio: transferable to overall nodes	$\frac{\sum_{transferable\ nodes} ALE(X_i)}{ALE(\boldsymbol{X})}$	87%	$\frac{ALE(X_{3c})}{ALE(\boldsymbol{X_c})} \leq 87\%$

Figure 6-7 illustrates how the inferred $P(x_s)$ value and risk function $ALE(X_s)$ behave across the range of probability values for each of $P(x_1)$ and $P(x_2)$ individually, when all other parameters are fixed. Given the current case study inputs, the slopes of both $P(x_s)$ and $ALE(X_s)$ when X_2 is considered—0.03 and 1.75, respectively—are higher than the corresponding slopes when X_1 is considered: $0.01, 0.69$. In addition, the y-intercept values of both $P(x_s)$ and $ALE(X_s)$ when X_2 is considered—0.08 and 5.80, respectively—are smaller than their values when X_1 is considered: $0.09, 6.03$. Thus, the effect of the operational capabilities of the biometric reader X_2 is more significant than the authentication server X_1 with respect to both system-level metrics $P(x_s)$ and $ALE(X_s)$.

[22] Note that the column of cloud configuration shows the region of values as opposed to specific values, indicating where the cloud configuration could outperform the original one considering the cloud's best-case scenario.

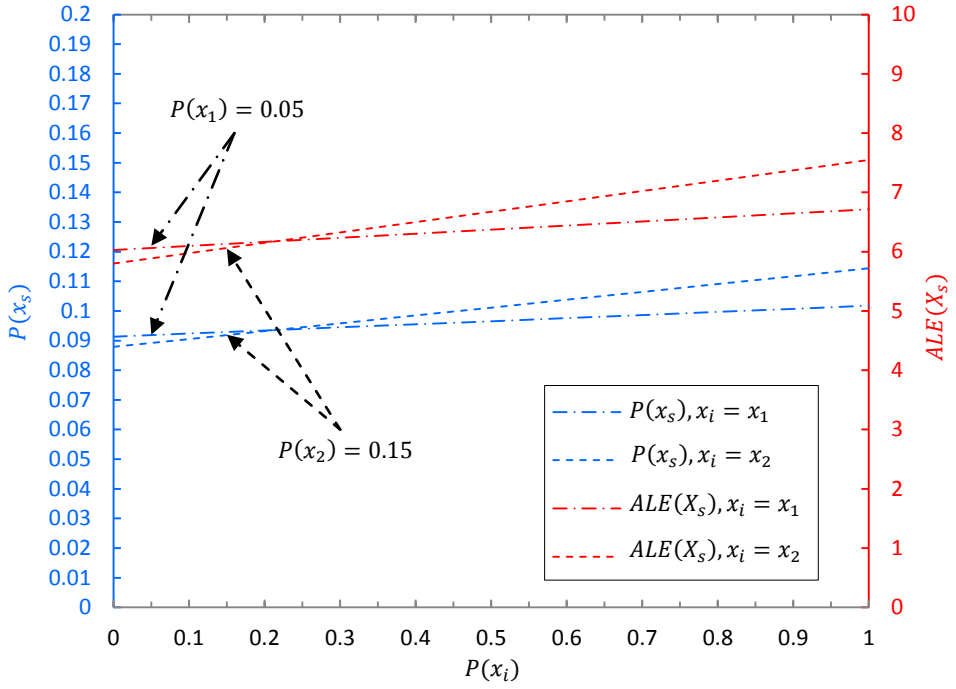

Figure 6-7: Evaluation of $P(x_s)$ and $ALE(X_s)$ over the range of probability values for both $P(x_1)$ and $P(x_2)$ individually[23]

As a result, the investment in control X_2 leads to increasing system-level security (by lowering $P(x_s)$ more and faster), decreasing risk (by lowering $ALE(X_s)$ more and faster), and therefore, increasing economical return more than the investment in control X_1. Furthermore, the figure shows consistency and prediction features in the model using $P(.)$ and $ALE(.)$ functions. Figure 6-8 and Figure 6-9 further support such observations when the full range of $P(x_1)$ and $P(x_2)$ values are considered together.

[23] Current case study values are marked showing corresponding values on $P(x_s)$ and $ALE(X_s)$ axes. The graph shows that X_2 is more significant than X_1 with respect to both $P(x_s)$ and $ALE(X_s)$.

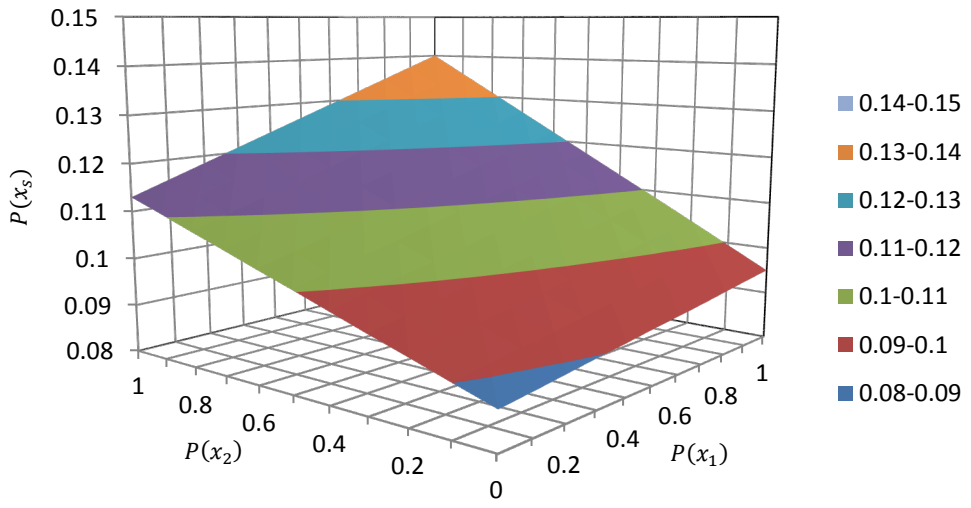

Figure 6-8: Evaluation of $P(x_s)$ over the range of probability values of both $P(x_1)$ and $P(x_2)$ together[24]

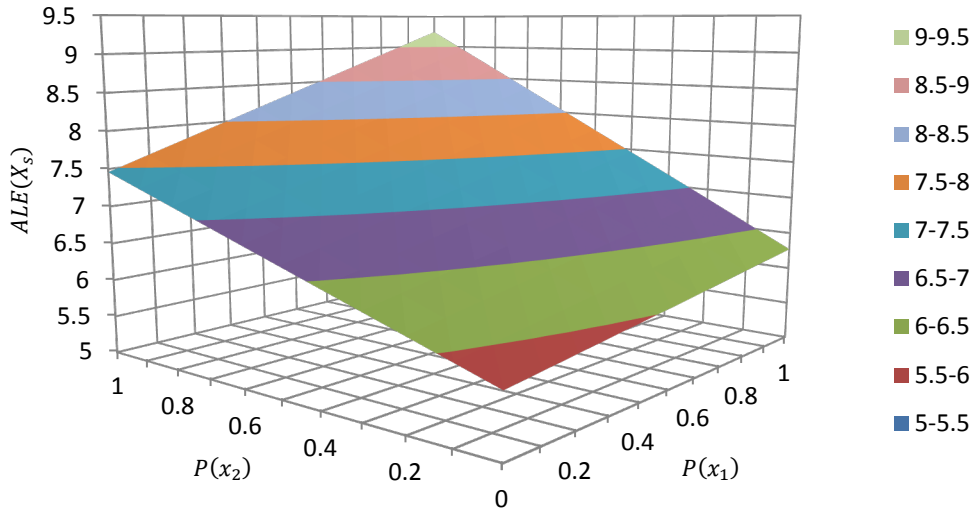

Figure 6-9: Evaluation of $ALE(X_s)$ over the range of probability values of both $P(x_1)$ and $P(x_2)$ together[25]

[24] The graph shows how X_2 increases/decreases $P(x_s)$ more and faster than X_1 does. It also shows the consistency of such functions.

6.8 Mathematical Proofs

Two proofs are presented in this part. The first is for the property of $ALE(X_s/X_s)$ bounds, and the second is for the distinction property of high-frequency low-impact from low-frequency high-impact events.

6.8.1 Proof of $ALE(X_S/X_s)$ Bounds

We want to prove the relationship:

$$ALE(X_s/X_i) \leq ALE(X_s/X_s) < ALE(X_i/X_s), \qquad for\ all\ X_i \in pa(X_s), pa(X_s) \notin \{\emptyset\}$$

The first part:

$$\begin{aligned} ALE(X_s/X_i) &= ALE_n(X_s/X_i) + \sum_{all\ X_k \in dec(X_s)} ALE_n(X_k/X_s X_i) \\ &= P(x_s/x_i) \times Imp(X_s) \times Val(X_s) \\ &+ ALE_n\{\emptyset\}, \qquad since\ X_s\ is\ a\ leaf\ nde \\ &\leq ALE(X_s/X_s) \\ &= P(x_s/x_s) \times Imp(X_s) \times Val(X_s) \\ &= 1 \times Imp(X_s) \times Val(X_s). \end{aligned}$$

Note that the equality occurs when $P(x_s/x_i) = 1$.

The second part:

$$\begin{aligned} ALE(X_i/X_s) &= ALE_n(X_i/X_s) + \sum_{all\ X_k \in dec(X_i)} ALE_n(X_k/X_i X_s) \\ &= P(x_i/x_s) \times Imp(X_i) \times Val(X_i) \\ &+ \cdots, \qquad for\ some\ middle\ nodes, if\ any \\ &+ ALE_n(X_s/X_i X_s), \qquad since\ X_s \in dec(X_i) \\ &> ALE(X_s/X_s). \end{aligned}$$

Thus, the relationship holds ■

[25] The graph shows how X_2 increases/decreases $ALE(X_s)$ more and faster than X_1 does. It also shows the consistency of such functions.

6.8.2 Proof of $ALE(.)$ Distinguishing High-frequency Low-impact from Low-frequency High-impact Events

We want to show that for all $X_i \in \boldsymbol{X}, dec(X_i) \notin \{\emptyset\}$, the function $ALE(X_i)$ returns two different yet consistent risk values for the cases: high $P(x_i)$, low $Imp(X_i)$; and low $P(x_i)$, high $Imp(X_i)$. To show that, let the subscripts l, h denote low value and high value, respectively, taken by $P(x_i)$ and $Imp(X_i)$. This means l is close to 0 and h is close to 1. We prove the distinction property in the worst-case scenario, which occurs when $P_h(x_i) + Imp_l(X_i) = 1$ and $P_l(x_i) + Imp_h(X_i) = 1$.

For high-frequency low-impact events, we write $ALE(X_i)$ as follows

$$
\begin{aligned}
ALE(X_i) &= ALE_n(X_i) + \sum_{all\ X_j \in dec(X_i)} ALE_n(X_j/X_i) \\
&= P_h(x_i) \times Imp_l(X_i) \times Val(X_i) \\
&+ P_l(x_m/x_i) \times Imp(X_m) \times Val(X_m), \qquad for\ some\ middle\ nodes\ m, if\ any \\
&+ P_l(x_k/x_i) \times Imp(X_k) \times Val(X_k), \qquad where\ k\ is\ the\ leaf\ node
\end{aligned}
$$

Note that in high-frequency low-impact events, consequent probabilities take low values representing low impact. Thus, $P_l(x_m/x_i)$ and $P_l(x_k/x_i)$ are considered.

Similarly, for low-frequency high-impact events, denoted as $\widehat{ALE}(X_i)$, we write

$$
\begin{aligned}
\widehat{ALE}(X_i) &= \widehat{ALE}_n(X_i) + \sum_{all\ X_j \in dec(X_i)} \widehat{ALE}_n(X_j/X_i) \\
&= P_l(x_i) \times Imp_h(X_i) \times Val(X_i) \\
&+ P_h(x_m/x_i) \times Imp(X_m) \times Val(X_m), \qquad for\ the\ same\ middle\ nodes\ m, if\ any \\
&+ P_h(x_k/x_i) \times Imp(X_k) \times Val(X_k), \qquad fore\ the\ same\ leaf\ node\ k
\end{aligned}
$$

Subtracting both risk terms leads to

$$
\begin{aligned}
&ALE(X_i) - \widehat{ALE}(X_i) \\
&= [P_h(x_i) \times Imp_l(X_i) \times Val(X_i) - P_l(x_i) \times Imp_h(X_i) \times Val(X_i)] \\
&+ [P_l(x_m/x_i) \times Imp(X_m) \times Val(X_m) - P_h(x_m/x_i) \times Imp(X_m) \times Val(X_m)] \\
&+ [P_l(x_k/x_i) \times Imp(X_k) \times Val(X_k) - P_h(x_k/x_i) \times Imp(X_k) \times Val(X_k)] \\
&= 0 \\
&+ [P_l(x_m/x_i) - P_h(x_m/x_i)] \times Imp(X_m) \times Val(X_m) \\
&+ [P_l(x_k/x_i) - P_h(x_k/x_i)] \times Imp(X_k) \times Val(X_k)
\end{aligned}
$$

Setting the result to 0 to check for equality conditions, where all the terms $P(.), Imp(.), Val(.)$ take positive values, $ALE(X_i) - \widehat{ALE}(X_i) = 0$ only when

$$P_l(x_m/x_i) = P_h(x_m/x_i) = 0.5$$

and

$$P_l(x_k/x_i) = P_h(x_k/x_i) = 0.5$$

Thus, the distinction property holds ■

The proof not only shows the distinction between high-frequency low-impact events and low-frequency high-impact events, but it further reflects the behaviours of these two scenarios. For any given node, the curves of $ALE(X_i), ALE(X_s)$, and $P(x_s)$ in the case of low-frequency high-impact events are higher than those of high-frequency low-impact ones. This property coincides with the observation that, while in many cases high-frequency low-impact events can be manageable as incremental costs, low-frequency high-impact events can be catastrophic, as reported in [10]. Figure 6-10, Figure 6-11, and Figure 6-12 demonstrate this property on X_1 in the original configuration of the case study presented earlier.

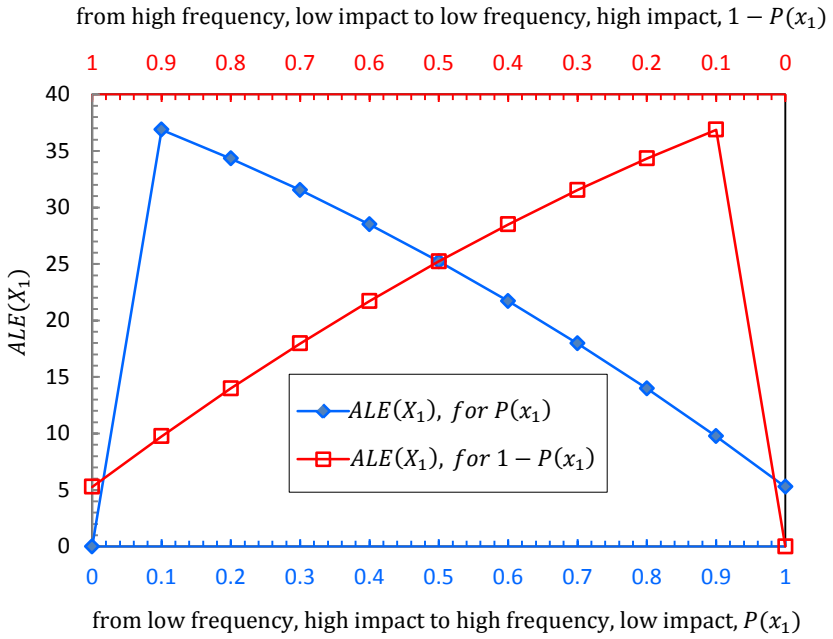

Figure 6-10: Evaluation of distinction property proof based on $X_1, ALE(X_1)$ scenario

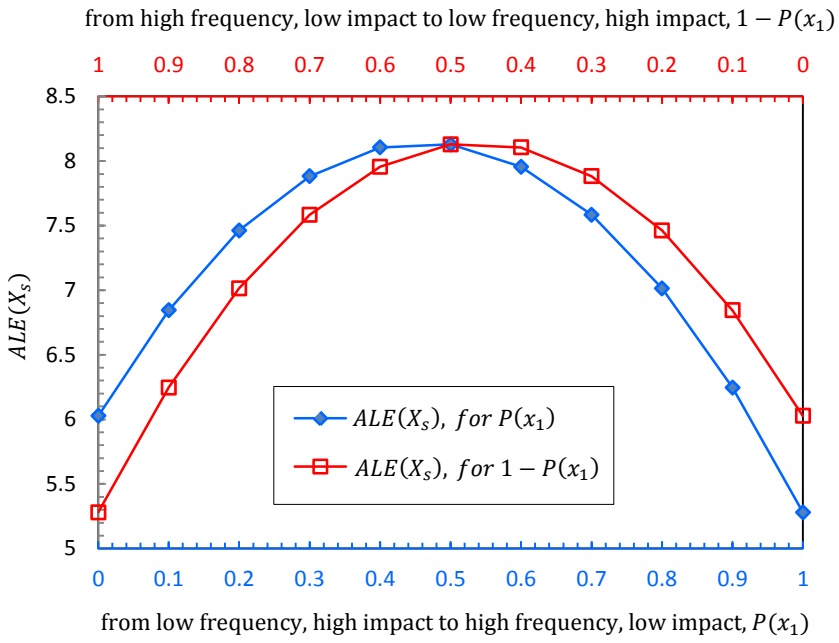

Figure 6-11: Evaluation of distinction property proof based on $X_1, ALE(X_s)$ scenario

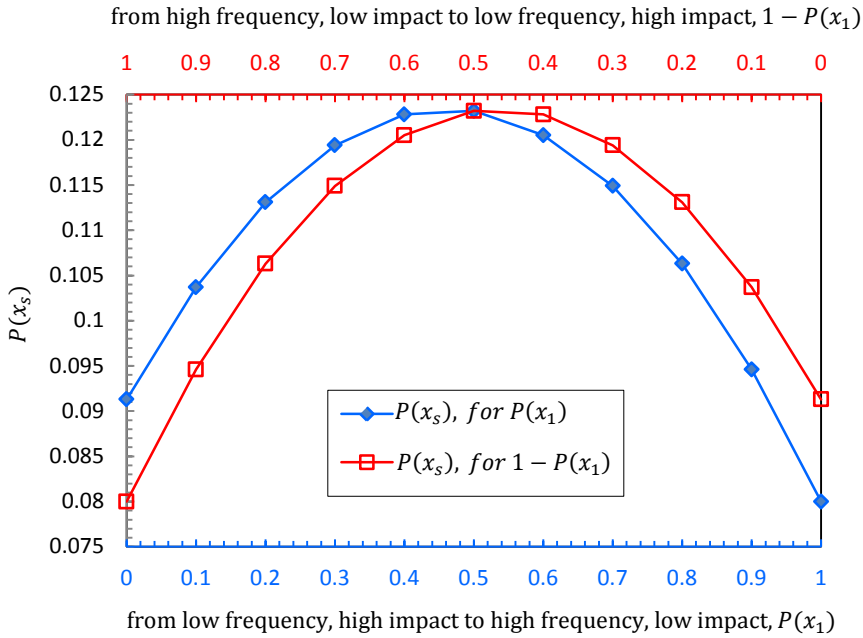

Figure 6-12: Evaluation of distinction property proof based on $X_1, P(x_s)$ scenario

6.9 Summary

The advantages of employing the logical structure of controls to solve the evaluation problem of operational capabilities of security systems have been established in the previous chapters. More specifically, system-level representation has been established in Chapter 3 using the ISMM model and representation methods from reliability theory; the extension of evaluation methods from reliability theory has also been demonstrated Chapter 4; and the extension of multi-state systems evaluation using the UGF method has been shown in Chapter 5. Such extensions allow one to particularly establish various performance measures, such as availability and reliability measures, and to build accordingly quantitative maturity analysis.

In this chapter, we have shown how the abstraction of a computing system into assets and controls can be further used to establish a different evaluation method in security studies. Building on this abstraction and the failure model explained earlier, this chapter contributes to existing work in risk assessment by considering the fact that computing paradigms are evolving by nature. We have first shown how to establish the system model of assets and controls using Bayesian networks, forming what we have called asset-control BN. Then we have explained and demonstrated the proposed risk assessment approach, addressing major challenges found in traditional risk assessment methods. Particularly, we have addressed the challenge of security failure quantification by using the impact of failure statistics as opposed to underlying failure mechanisms, reducing the complexity of the failure space. We have also illustrated how the model facilitates various useful inferences and types of analysis using BNs primitives. Finally, we have provided two proofs: 1) a specific bound property of the risk function; and 2) a resolution of the distinction problem between high-frequency low-impact events and low-frequency high-impact ones, by employing the casual effect property defined by BN topology.

The analysis and examples given illustrate that the use of the proposed risk assessment is considerably more practical and reliable than common risk assessment methods, especially when underlying failure mechanisms, whether due to malicious attacks or accidental events, are unattainable. Thus, for many scenarios, this method can be a promising approach.

Chapter 7
Contributions and Future Work

In this chapter we summarise the overall contribution of this document and present some open problems that can be used as basis for advancing this research direction.

7.1 Summary

This work addresses the problem of system-level security quantification. Across the various security domains, this problem is hard, as the "physics" of the amount of security is ambiguous, the operational state is defined by two confronting parties, the problem is about multiple conceptual boundaries across multiple abstraction levels, computing paradigms are evolving by nature, protecting and breaking systems is a cross-disciplinary mechanism, security is achieved by comparable security strength and breakable by the weakest link, the human factor is unavoidable, and no universally accepted abstraction of a computing system or failure model is available, to say the least.

To tackle these issues, we have examined the problem at its abstraction level with comparison to related, well-founded engineering disciplines. We have realised that the following components are essential to arriving at a consistent evaluation methodology: a unified abstraction of computing systems, a standardized failure model, bounding system model(s), performance measure(s), and evaluation technique(s). We have also found that the effect of security controls and the impact of their failures in a computing system can be captured in a structural way, by which these modeling components can be established. Subsequently, we have shown how the logical network and failure statistics of individual system components, in addition to design bounds, can be used as the main input to carry out various quantitative analysis tasks. The proposed evaluation methods, however, are independent from their corresponding system models, facilitating a wider range of evaluation techniques. In what follows, we summarise the contribution of each chapter individually.

In Chapter 3, we have first addressed the problem of a unified paradigm of security modeling, particularly establishing computing system abstraction and a failure model. This has led to abstracting any computing system into the two sets of assets and controls, bounding what defines a computing system, and the definition of failure based on its impact consequent to malicious and/or nonmalicious causes, reducing the complexity of the failure space. These components have been consistently used as the foundation for the evaluation methods throughout this work.

Then, we have addressed the issue of developing a bounding system model for evaluating security systems as a separate entity. To do so, we have reconstructed the ISMM model to map the structural arrangement among the set of controls with respect to failure so that quantitative evaluation, including maturity, can be established. We have shown that various concepts and representation and analysis techniques can be extended from dependability evaluation. In particular, we have extended the use of Reliability Block Diagrams (RBDs), state vectors, and structure functions to bound and build the logical network of a security system. To evaluate these ideas and forthcoming evaluation methods, we have built a case study applying such extensions.

Chapter 4 has followed with the analysis techniques from reliability engineering. We have shown the mathematical formulation of minimal path sets, minimal cut sets, and reliability analysis based on both random events and random variables. The minimal sets allow one to examine the critical controls that affect the operational state of security, whereas reliability analysis can be used to calculate reliability and maturity measures of the security system. We have demonstrated different redundancy configurations and failure rates of controls, along with different ways to find the mean time to failure ($MTTF$) and the mission time (t) that increases reliability and maturity values, at subsystem and system levels. Such analysis can be used to build more reliable security systems.

Chapter 5 has addressed the issue of security quantification when multi-state security systems, beyond binary systems, with multiple performance measures, beyond reliability and availability, are considered. It has been recognised that Multi-State Systems (MSS) representation and the Universal Generating Function (UGF) provide a suitable methodology to resolve the issue. Building on the same modeling paradigm, we have shown the formulation necessary to establish these methods on the ISMM model, establishing the multi-layer MSS (MLMSS) model. In a parallel analysis to the previous chapter, we have also demonstrated the reliability and maturity analysis. Moreover, we have shown how to perform structural evaluation of various permutations of intermediate u-functions towards the product u-function in a systematic and progressive manner.

Chapter 6 has addressed the problem of security quantification when both sets of a computing system, i.e., assets and controls, are considered. Alternatively to the ISMM system model, which only considers the set of controls, and building on the same modeling paradigm, we have adopted a graph-theoretic approach to model the relationships among both sets. We have proposed asset-control BNs as the bounding system model to capture the failure interdependency among such entities. In this representation, besides failure analysis of individual nodes, we have found that this model facilitates

the study of failure consequences and estimated damage using BNs properties, rendering it a useful model for risk assessment applications. Consequently, we have proposed a new risk assessment method and have shown its mathematical formulation with a brief example demonstration. Various diagnosis and prediction inferences have been demonstrated, addressing several aspects of security and its economics, at both node and system levels. We have also provided proof of certain bounds on risk function, and proof that the method resolves the distinction problem between high-frequency low-impact events and low-frequency high-impact ones by employing the casual effect property defined by BN topology.

The proposed evaluation methods can be applied to both systems under design and existing systems, and can be used by both security engineers and dependability engineers. For systems under design, they can be used to engineer security and dependability requirements, including the development of alternative design options and the analysis of potential threat and failure models. For existing systems, they can be employed to provide various operational measures of security and dependability, and to audit, monitor, and predict security behavior. They can also be used to identify and evaluate alternative countermeasures and enhancement upgrades. Overall evaluations can be performed at component-, subsystem-, and system levels, addressing key aspects from both security-related analysis and dependability-related analysis.

Nevertheless, while this research has exploited the security modeling paradigm and provided various evaluation methods, some limitations exist. For instance, a classical limitation occurs when the number of controls in a given system grows, rendering the corresponding mathematical problem intractable. This difficulty arises due to the exponential growth of the number of variables involved in building the system model. Fortunately, various approximation methods can be used to overcome such a limitation in a way analogous to that of reliability engineering. One might also notice the limitation in validating this type of work due to the unavailability of representative datasets, as explained earlier in Section 2.1.5.

7.2 Future Work

In the following sections, we briefly describe some of the potential research directions for the core components of this research.

7.2.1 The ISMM Model

Side note #1. In this work, ISMM structures play a major role to establish the system model as they bound the structural relationships among the set of controls. We have only considered simple structures to introduce the extended evaluation methods. Being based on reliability block diagrams, these structures will eventually restrict any representation to mixed combinations of series and parallel connections. One direction in this regard is to extend this representation to complex structures, such as bridge, star, and delta [47], [105], to represent more-complex relationships of controls. Furthermore, another direction of research is to extend graph-theoretic structures to model much more-complex relationships. This extension can be approached using reliability graphs, non-series-parallel block diagrams, which can be found in [33] or Bayesian networks, which can be found in [96], [97].

Side note #2. Recall that the general form of the maturity function in Section 3.10 is $\mathcal{M}_{ISMM} = \mathcal{F}(\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_l)$, a function defined over various performance measures. Each measure $\mathcal{M}_k$ involves the evaluation of the two sets M_{i_k} and $threshold\left(\mathrm{M}_{i_k}\right)$, representing a certain performance and demand, respectively. These two sets can be formulated using the MSS UGF method, particularly, using the demand function $F\big(G(t), W(t)\big) = G(t) - W(t)$ and the corresponding u-function form, where $G(t)$ and $W(t)$ represent performance and demand functions, respectively. The interested researcher is referred to [116], [117] for further details on these functions.

Side note #3. The use of importance analysis in multi-component systems can be very useful. For instance, the Birnbaum Importance measure [102], [105], [167] in reliability evaluation represents the rate at which a system's reliability increases when the reliability of a particular component increases. Analytically, it is defined by $I_k = \frac{\partial R_s(t)}{\partial R_k(t)}$, where I_k is the reliability importance of the kth component, $R_s(t)$ and $R_k(t)$ are system reliability and component k reliability at time t, respectively. These kinds of measures merit further study in this work and current studies in security evaluation in general.

Side note #4. For decades, maturity models have been qualitative in nature. The most prominent ones are those developed by the Software Engineering Institute (SEI) at Carnegie Mellon University. Three examples from the SEI are the Software CMM (SW-CMM), the Systems Engineering CMM (SE-CMM), and the CMM Integration (CMMI). In addition, the Systems Security Engineering Capability Maturity Model CMM (SSE-CMM) developed by The International Systems Security Engineering Association (ISSEA, established in 1999), which has been accepted by the International Organization

for Standardization (ISO) as ISO/IEC 21827 standard. However, while these CMMs have been useful in their applications, they have remained of a qualitative nature, only reflecting how the process is implemented, disallowing them from providing quantitative, operational measurement of the delivery itself. As such, the context of the ISMM model, including its quantifying semantics, demonstrated in this work has a research potential for adoption by those qualitative maturity models.

Side note #5. The main focus of ISMM structures has been on security controls, without a concrete consideration of privacy, which is sometimes addressed distinctly. One possible research direction is to exploit this avenue.

7.2.2 Reliability-theoretic ISMM-based Analysis

Side note #1. A natural extension to the demonstrated reliability-based analysis is to introduce the availability measure and repairable systems in the security context. These aspects have a great potential to address the maintainability of security systems. Recall that availability is defined by $\frac{up\ time}{operational\ cycle} = \frac{MTTF}{MTTF+MTTR}$, requiring more statistical information about the recovery of failing controls. As such, that this of analysis normally requires models based on stochastic processes such as Markov and semi-Markov [47], [80], [168].

Side note #2. There are various techniques to evaluate the reliability of complex systems that can be extended to this work. Examples include Fault Tree Analysis (FTA) and Event Tree Analysis (ETA). FTA is considered a popular deductive method (top-down) to evaluate reliability of complex systems [105] with respect to the failure space [74]. This method is designed to use simple logical relationships (i.e., AND, OR, XOR, PRIORITY AND, INHIBIT, DELAY) to represent the structure of different relationships among lower-level events and ultimately faults or failures of the top event whereby probabilistic analysis is performed [47], [168]. Thus, FTA can be useful for examining failure scenarios [75], [76]. Alternatively, event trees are constructed using the complete event space of all possible events representing components in a system. Event trees are built based on an inductive approach (bottom-up), and thus can be useful to study the holistic view of systems [47], [168].

Side note #3. Many useful optimization applications in reliability engineering can be useful to this study, including the economics of security. Such formulations and solution techniques can be extended to address various optimization objectives in this work, such as redundancy, cost, maturity, and reliability. The following resources can be consulted for more details in optimization: [47], [118], [169]

7.2.3 ISMM-based MSS Evaluation Using UGF

Side note #1. k-out-of-n systems receive special attention in various system-evaluation applications. These systems exist in many forms with different models. In the general case, a k-out-of-n system functions (fails) if and only if at least k of its components function (fail). A special case is the Consecutive k-out-of-n system, which adds the restriction of working (failing) components being in consecutive order. In this thesis, however, we have introduced the concept of *precedence* to the ISMM work to distinguish it from the simple ordering of security layers. Building on this property, a new case of k-out-of-n systems, perhaps the Precedence k-out-of-n system, can be established. This model should reflect the special feature of security systems where the protection (failure) occurs when a set of controls in certain precedence functions (fails), not necessarily in consecutive order of the controls. To this end, precedence needs to be defined by some rules added to the set of k. Details on several k-out-of-n models, however, can be found in [116], [118], [132].

Side note #2. In this work we have addressed the case of MSS systems with independent controls. Similar to reliability-based analysis, a natural direction is to extend the modeling of dependent components, the availability measure, and repairable systems too [116], [117].

Side note #3. The MSS UGF technique facilitates various optimization problems as it allows a fast evaluation of the system performance distribution. Optimization using genetic algorithms, which is based on evolutionary searching of a solution space, represents a particular match to this setting. More techniques, however, can be found in [116], [117], [118], [119].

7.2.4 Asset-control Graphs

Side note #1. Above, we have not considered how to build the BN graphs by applying structure and parameter learning algorithms. Existing literature on BNs offers well-founded learning methods and algorithms that can be constructively extended to this work. The interested researcher is referred to [32], [142], [165].

Side note #2. In this work, we have only addressed the case when there are no cycles between asset and control nodes in the graph. While this simplifying assumption brings the great benefits of BN inference algorithms and associated analysis, it remains a limitation on the mutual relationships between system nodes. As a direction of research to overcome this limitation, we think allowing cycles and directed and undirected graphs, whereby more system configurations and asset-control

behaviours can be modeled, merits further extensions. Details on various graph classes can be found in [170], [171].

Side node #3. In uncertain environments such as security, reasoning over time becomes of great benefit to better analysing the changing world. To facilitate such study, Dynamic Bayesian Networks (DBNs) can be used, and can be found in [32], [142], [165]. In this setting, asset and control variables are related to each other over adjacent time steps.

Side node #4. The proposed asset-control BNs formalism allows us to map qualitatively and quantitatively the dependency and impact of security failures among assets and controls using BN topology and CPTs. This representation provides a system bounding that can be useful in formulating various optimization problems. For example, for a given expenditure of money units and a set of configuration alternatives with a set of asset-control dependency options, how much of a decrease in a particular risk term, perhaps system risk $ALE(X_S)$, can be achieved? What is the minimum protection cost required to reach an acceptable, predefined level of risk? How much should be set to align the incentives to protecting a system with the "suffer [143]" when it fails? Other queries might also involve identifying latent failures and associated risks and their paths, for example, finding the most (least) probable path of system threshold failure, affected nodes, associated risks, and their probabilities. In general, such an analysis can be useful in engineering a system's configuration, given some risk and economic constraints, whereby the analysis can be performed at the individual component level, system level, or a subset of its components. Various optimization resources, such as [172], [173], can be consulted for these researchable problems.

Appendix A: Notation for ISMM-based Analysis

$\mathcal{M}_{ISMM}$	ISMM overall maturity value
$\mathcal{M}_k$	system maturity value based on measure type k
$\mathcal{M}_R$	system maturity value based on reliability measure
M_{i_k}	the value of measure k at layer i
$threshold(\mathrm{M}_{i_k})$	adequacy or maturity minimum bound for measure k at layer i
$\mathcal{F}$	maturity adequacy function
n_i	the total number of security controls at $layer_i$ or $subsystem_i$
$C_{i,j}$	security control j at layer i, $where\ i = layers\ 1,\dots,5;\ \ j = controls\ 1,\dots,n_i\ at\ layer\ i$
$\boldsymbol{C_i}$	$\{C_{i,j}, i = 1,\dots,5; j = 1,\dots,n_i\}$, set of controls for $subsystem_i$
$\boldsymbol{C}$	$\{C_i, i = 1,\dots,5\}$, set of subsystems of the system
$\emptyset(\boldsymbol{x_i})$	structure function of the vector $\boldsymbol{x_i}$ for structure or $subsystem_i$
$\boldsymbol{A_{i,k}}$	the *k*th minimal path set for $subsystem_i$
$\propto_{i,j}(\boldsymbol{x_i})$	minimal path set function for the set $\boldsymbol{A_{i,j}}$
$\boldsymbol{C_{i,k}}$	the *k*th minimal cut set for $subsystem_i$
$\beta_{i,j}(\boldsymbol{x_i})$	minimal cut set function for the set $\boldsymbol{C_{i,j}}$
$R_{i,j}$	reliability of control j at layer i
$\boldsymbol{A_1}$	the minimal path set for $system_{ISMM}$
$\boldsymbol{C_i}$	the *i*th minimal cut set for $system_{ISMM}$
$x_{i,j}$	the event of control $C_{i,j}$ is working
$\mathrm{X}_{i,j}$	a random variable representing the state of control $C_{i,j}$, working if $\mathrm{X}_{i,j} = 1$
$R_{i,j}$	reliability of control $C_{i,j}$
R_i	reliability of $subsystem_i$
R_{ISMM}	reliability of the overall security system
r_i	adequacy or maturity minimum bound for reliability measure at layer i. That is, the minimum reliability bound necessary for layer i to meet the reliability criterion set for the maturity qualification of layer i
$R_{i,j}(t)$	reliability of control $C_{i,j}$ as a function over time

$R_i(t)$	reliability of $subsystem_i$ as a function over time
$R_{ISMM}(t)$	reliability of the overall security system as a function over time
$f_{i,j}(t)$	failure density function for control $C_{i,j}$
$f_i(t)$	failure density function for $subsystem_i$
$F_{i,j}(t)$	failure distribution control $C_{i,j}$
$F_i(t)$	failure distribution $subsystem_i$
$\lambda_{i,j}$	constant failure rate for control $C_{i,j}$
$\lambda_{i,j}(t)$	hazard function for control $C_{i,j}$
$MTTF_{i,j}$	mean time to failure for control $C_{i,j}$
MSS	multi-state system
MLMSS	multi-layer multi-state system
UGF	universal generating function
ODP	output performance distribution
$g_{i,j,l}$	performance rate with associated probability $p_{i,j,l}$ for control $C_{i,j}$ in the state l, $l \in \{1,2,\dots,K_{i,j}\}$
$K_{i,j}$	number of different performance rates for control $C_{i,j}$
$\boldsymbol{g_{i,j}}$	set of performance rates for control $C_{i,j}$
$\boldsymbol{p_{i,j}}$	set of probabilities of performance rates for control $C_{i,j}$
$G_{i,j}(t)$	performance rate in the stochastic domain with associated probability $P_{i,j}(t)$ for control $C_{i,j}$
$g_{i,l}$	performance rate with associated probability $p_{i,l}$ for $subsystem_i$ (or $\boldsymbol{C_i}$) in the state $l, l \in \{1,2,\dots,K_i\}$
K_i	number of different performance rates for $subsystem_i$ (or $\boldsymbol{C_i}$)
$\boldsymbol{g_i}$	set of performance rates for $subsystem_i$
$\boldsymbol{p_i}$	set of probabilities of performance rates for $subsystem_i$
$G_i(t)$	performance rate in the stochastic domain with associated probability $P_i(t)$ for $subsystem_i$ (or $\boldsymbol{C_i}$)
g_l	performance rate with associated probability p_l for the system $\boldsymbol{C}$ in the state $l, l \in \{1,2,\dots,K\}$

k	number of different performance rates for $system_{ISMM}$ (or $\boldsymbol{C}$)
$\boldsymbol{g}$	set of performance rates for $system_{ISMM}$ (or $\boldsymbol{C}$)
$\boldsymbol{p}$	set of probabilities of performance rates for $system_{ISMM}$ (or $\boldsymbol{C}$)
$G(t)$	performance rate in the stochastic domain with associated probability $P(t)$ for $system_{ISMM}$ (or $\boldsymbol{C}$)
$\boldsymbol{L_i^{n_i}}$	space for all possible combinations of performance rates for all controls at layer $\boldsymbol{C_i}$
$\boldsymbol{M_i}$	space of all possible values of performance rates for layer $\boldsymbol{C_i}$
$\boldsymbol{L^n}$	space of all possible combinations of performance rates for all system $\boldsymbol{C}$ layers
$\boldsymbol{M}$	space of all possible values of the performance rates of the system $\boldsymbol{C}$

Appendix B: Notation for Asset-control Risk Assessment

$\boldsymbol{S}$	$\{S_i: S_i\ is\ a\ component\}$, set of system components
$\boldsymbol{A}$	$\{A_i: A_i\ is\ an\ asset\}$, set of assets
$\boldsymbol{C}$	$\{C_i: C_i\ is\ a\ control\}$, set of controls
$\boldsymbol{V}$	$\{Assets\ \boldsymbol{A}, Access\ controls\ \boldsymbol{C}\}$
$\boldsymbol{E}$	$\{failure\ or\ breach\ dependency\}$
$\boldsymbol{G}$	$(\boldsymbol{V}, \boldsymbol{E})$, Graph $\boldsymbol{G}$ of $\boldsymbol{V}$ nodes and $\boldsymbol{E}$ edges
$\boldsymbol{P}$	probability distribution over $\boldsymbol{V}$ for failure dependency
$\boldsymbol{X}$	$\{X_1, X_2, \dots, X_n\}$, system random variables
X_i	r.v. representing the state of failure of node i
$P(x_i)$	$P\{X_i = x_i\}$, probability of failure of node i
$pa(X_i)$	set of parents of X_i in $\boldsymbol{G}$
$dec(X_i)$	set of descendants of X_i in $\boldsymbol{G}$
$\boldsymbol{X_{-i}}$	set of nodes excluding node i
ALE	Annualized Loss Expectancy
SLE	Single Loss Expectancy
ARO	Annual Rate of Occurrence
$ALE_n(X_i)$	Annualized loss expectancy of node i alone
$ALE_n(X_i/X_j)$	Annualized loss expectancy of node i alone, given evidence of X_j occurrence
$ALE(X_i)$	Annualized loss expectancy of node i, including the set of its descendants
$ALE(X_i/X_j)$	Annualized loss expectancy of node i, including the set of its descendants, given evidence of X_j occurrence
$ALE(X_i \backslash \boldsymbol{X}_{-i})$	Annualized loss expectancy of node i, including the set of its descendants, excluding the set $\boldsymbol{X}_{-i}$
$ALE(X_s)$	Annualized loss expectancy of system threshold node X_s
$ALE_n(\boldsymbol{X})$	Annualized loss expectancy of system $\boldsymbol{X}$ nodes alone (excluding their descendants)
$ALE(\boldsymbol{X})$	Annualized loss expectancy of system $\boldsymbol{X}$ nodes, including the set of their descendants

$tag(A_i)$	$(Val(A_i), Imp(A_i))$, attributes of asset A_i
$Val(A_i)$	value of asset A_i
$Imp(A_i)$	impact factor of asset A_i
$Avl(A_i)$	availability of asset A_i
$tag(C_i)$	$(Cst(C_i), Imp(C_i), Gol(C_i), Typ(C_i))$, attributes of control C_i
$Cst(C_i)$	cost of control C_i
$Imp(C_i)$	impact factor of control C_i
$Gol(C_i)$	security process goal of control C_i
$Typ(C_i)$	type of control C_i

References

[1] A. Whitten and J. D. Tygar, "Why johnny can't encrypt: A usability evaluation of PGP 5.0," in *Proceedings of the 8th USENIX Security Symposium,* 1999.

[2] S. S. Alaboodi, "A New Approach for Assessing the Maturity of Information Security," *Information Systems Control Journal,* vol. 3, pp. 36, 2006.

[3] M. Dacier, Y. Deswarte and M. Kaaniche, "Quantitative Assessment of Operational Security: Models and Tools," *LAAS Research Report,* vol. 96493, pp. 5, 1996.

[4] T. Reid and S. W. Hamilton, *Essays on the Intellectual Powers of Man.* J. Bartlett, 1850.

[5] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid, "Recommendation for Key Management–Part 1: General," *NIST Special Publication 800-57,* 2005.

[6] R. Richardson, "CSI computer crime and security survey," *Computer Security Institute,* vol. 1, pp. 1-42, 2010/2011.

[7] D. M. Nicol, W. H. Sanders and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, pp. 48-65, 2004.

[8] A. Avizienis, J. -. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, pp. 11-33, 2004.

[9] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid and D. Gollmann, "Towards Operational Measures of Computer Security," *Journal of Computer Security,* vol. 2, pp. 3, 1993.

[10] K. J. S. Hoo, "How much is enough? A risk-management approach to computer security," in *Workshop on Economics and Information Security, UC Berkeley, CA,* 2000.

[11] J. Homer, X. Ou and D. Schmidt, "A sound and practical approach to quantifying security risk in enterprise networks," *Kansas State University Technical Report,* pp. 1-15, 2009.

[12] E. Jonsson, "An integrated framework for security and dependability," in *Proceedings of the 1998 Workshop on New Security Paradigms,* 1998, pp. 22-29.

[13] A. P. Moore, R. J. Ellison and R. C. Linger, "Attack modeling for information security and survivability," DTIC Document, 2001.

[14] M. Greenwald, C. A. Gunter, B. Knutsson, A. Scedrov, J. M. Smith and S. Zdancewic, "Computer Security is not a Science (but it should be)," 2003.

[15] C. Robb and D. Robinson, "Deloitte-NASCIO Cybersecurity Study," *Deloitte and the National Association of State Chief Information Officers,* vol. 1, pp. 1-40, 2012.
[16] CSO magazine, U.S. Secret Service, Software Engineering Institute CERT Program at Carnegie Mellon University and Deloitte, "Cybersecurity Watch Survey," 2011.
[17] L. A. Gordon and M. P. Loeb, "The economics of information security investment," *ACM Transactions on Information and System Security (TISSEC),* vol. 5, pp. 438-457, 2002.
[18] R. Anderson, "Why cryptosystems fail," in *Proceedings of the 1st ACM Conference on Computer and Communications Security,* 1993, pp. 215-227.
[19] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing.* Prentice Hall, 2006.
[20] K. S. Trivedi, D. S. Kim, A. Roy and D. Medhi, "Dependability and security models," in *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on,* 2009, pp. 11-20.
[21] E. Jonsson, "Towards an integrated conceptual model of security and dependability," *The First International Conference on Availability, Reliability and Security, ARES 2006.,* pp. 8 pp., 2006.
[22] S. S. Alaboodi, *The Information Security Maturity Model (ISMM): A Comprehensive Approach for Researchers and Practitioners* VDM Verlag, 2009.
[23] S. S. Alaboodi, "Towards evaluating security implementations using the Information Security Maturity Model (ISMM)," *Master's Thesis, University of Waterloo, Canada,* 2007.
[24] K. Sallhammar, B. E. Helvik and S. J. Knapskog, "A Game-Theoretic Approach to Stochastic Security and Dependability Evaluation," *2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing,* pp. 61-68, 2006.
[25] K. Lye and J. M. Wing, "Game strategies in network security," *International Journal of Information Security,* vol. 4, pp. 71-86, 2005.
[26] P. Liu, W. Zang and M. Yu, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," *ACM Transactions on Information and System Security (TISSEC),* vol. 8, pp. 78-118, 2005.
[27] K. Sallhammar, B. E. Helvik and S. J. Knapskog, "Towards a stochastic model for integrated security and dependability evaluation," in *The First International Conference on Availability, Reliability and Security, ARES 2006.* 2006, pp. 8 pp.
[28] The Official Google Blog. A new approach to china. *2010(03/29),* 2010. Available: http://googleblog.blogspot.com/2010/01/new-approach-to-china.html.

[29] B. Gertz. Cyber-attack on U.S. firms, google traced to chinese. *The Washington Times* 2010. Available: http://www.washingtontimes.com/news/2010/mar/24/cyber-attack-on-us-firms-google-traced-to-chinese/print/.

[30] B. Blakley, E. McDermott and D. Geer, "Information security is information risk management," in *Proceedings of the 2001 Workshop on New Security Paradigms,* 2001, pp. 97-104.

[31] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems.* Wiley, 2010.

[32] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques.* The MIT Press, 2009.

[33] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications.* New York: Wiley, 2002.

[34] G. Bolch, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications.* Wiley-Blackwell, 2006.

[35] B. C. Williams and P. P. Nayak, "A model-based approach to reactive self-configuring systems," in *Proceedings of the National Conference on Artificial Intelligence,* 1996, pp. 971-978.

[36] R. Jain, *The Art of Computer Systems Performance Analysis.* John Wiley & Sons New York, 1991.

[37] I. Koren and C. M. Krishna, *Fault-Tolerant Systems.* Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2007.

[38] J. McLean, "The specification and modeling of computer security," *Computer,* vol. 23, pp. 9-16, 1990.

[39] B. Schneier, *Secrets & Lies: Digital Security in a Networked World.* John Wiley & Sons, Inc. New York, NY, USA, 2000.

[40] B. Schneier, "Attack trees," *Dr.Dobb's Journal,* vol. 24, pp. 21-29, 1999.

[41] S. J. Prowell, C. J. Trammell, R. C. Linger and J. H. Poore, *Cleanroom Software Engineering: Technology and Process.* Addison-Wesley Longman Publishing Co., Inc., 1999.

[42] Systems Engineering Capability Maturity Model Project, "A description of the systems engineering capability maturity model appraisal method, version 1.0 (CMU/SEI-94-HB-005)," Software Engineering Institute, Carnegie Mellon University, 1995.

[43] J. W. Lainhart IV, "COBIT™: A methodology for managing and controlling information and information technology risks and vulnerabilities," *J. Inf. Syst.,* vol. 14, pp. 21-25, 2000.

[44] NIST, "Program review for information security management assistance (PRISMA)," National Institute of Standards and Technology, Computer Security Resource Center (CSRC), 2005.
[45] C. Alberts and A. Dorofee, "An introduction to the OCTAVE method," *Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.Http://www.Cert.org/octave/methodintro.Html,* 2001.
[46] UK CCTA, "CCTA risk analysis and management method CRAMM," United Kingdom Central Computer and Telecommunication Agency, 2005.
[47] R. Ramakumar, *Engineering Reliability: Fundamentals and Applications.* Englewood Cliffs, N.J. : Toronto: Prentice Hall ; Prentice-Hall Canada, 1993.
[48] S. M. Ross, *Introduction to Probability Models.* Amsterdam ; Boston: Academic Press, 2007.
[49] M. T. Todinov, *Reliability and Risk Models: Setting Reliability Requirements.* Hoboken, NJ: Wiley, 2005.
[50] V. Gupta, V. Lam, H. V. Ramasamy, W. H. Sanders and S. Singh, "Dependability and Performance Evaluation of Intrusion-Tolerant Server Architectures," *Lecture Notes in Computer Science,* pp. 81-101, 2003.
[51] O. Sheyner, J. Haines, S. Jha, R. Lippmann and J. M. Wing, "Automated generation and analysis of attack graphs," *Proceedings. 2002 IEEE Symposium on Security and Privacy,* pp. 273-284, 2002.
[52] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan and K. S. Trivedi, "Modeling and quantification of security attributes of software systems," *Proceedings. International Conference on Dependable Systems and Networks, DSN 2002.,* pp. 505-514, 2002.
[53] W. H. Sanders, M. Cukier, F. Webber, P. Pal and R. Watro, "Probabilistic validation of intrusion tolerance," in *Digest of Fast Abstracts: The International Conference on Dependable Systems and Networks, Bethesda, Maryland,* 2002, .
[54] R. Ortalo, Y. Deswarte and M. Kaaniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering,* vol. 25, pp. 633-650, 1999.
[55] D. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering,* pp. 222-232, 1987.
[56] T. M. P. Lee, "Statistical models of trust: TCBs vs. people," *IEEE Symposium on Security and Privacy, 1989. Proceedings.,* pp. 10-19, 1989.

[57] M. Frigault, L. Wang, A. Singhal and S. Jajodia, "Measuring network security using dynamic bayesian network," in *Proceedings of the 4th ACM Workshop on Quality of Protection,* 2008, pp. 23-30.
[58] Y. Liu and H. Man, "Network vulnerability assessment using bayesian networks," in *Defense and Security,* 2005, pp. 61-71.
[59] D. Wang, B. B. Madan and K. S. Trivedi, "Security analysis of SITAR intrusion tolerance system," in *Proceedings of the 2003 ACM Workshop on Survivable and Self-Regenerative Systems: In Association with 10th ACM Conference on Computer and Communications Security,* 2003, pp. 23-32.
[60] B. Awerbuch, D. Holmer, C. Nita-Rotaru and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of the 1st ACM Workshop on Wireless Security,* 2002, pp. 21-30.
[61] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 Workshop on New Security Paradigms,* 1998, pp. 71-79.
[62] M. Schiffman, G. Eschelbeck and S. Romanosky, "CVSS: A common vulnerability scoring system," *National Infrastructure Advisory Council (NIAC),* 2004.
[63] P. Mell, K. Scarfone and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," in *Published by FIRST-Forum of Incident Response and Security Teams,* 2007, pp. 1-23.
[64] M. Dacier, Y. Deswarte and M. Kaaniche, "Models and tools for quantitative assessment of operational security," 1996.
[65] M. Steinder and A. Sethi, "End-to-end service failure diagnosis using belief networks," in *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP,* 2002, pp. 375-390.
[66] R. Albert, H. Jeong and A. L. Barabási, "Error and attack tolerance of complex networks," *Nature,* vol. 406, pp. 378-382, 2000.
[67] R. Dantu, K. Loper and P. Kolan, "Risk management using behavior based attack graphs," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on,* 2004, pp. 445-449 Vol. 1.
[68] R. Sahner, K. S. Trivedi and A. Puliafito, *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach using the SHARPE Software Package.* Kluwer Academic Publishers, 1996.

[69] J. B. Dugan, K. S. Trivedi, M. K. Smotherman and R. M. Geist, "The hybrid automated reliability predictor," *Journal of Guidance, Control, and Dynamics,* vol. 9, pp. 319-331, 1986.
[70] G. Ciardo, J. Muppala and K. Trivedi, "SPNP: Stochastic petri net package," in *Petri Nets and Performance Models, 1989. PNPM89., Proceedings of the Third International Workshop on,* 1989, pp. 142-151.
[71] D. B. Parker, *Fighting Computer Crime.* Scribner, 1983.
[72] J. D. Howard, "An analysis of security incidents on the Internet 1989-1995," *DTIC Document,* 1997.
[73] J. Laprie, A. Avizienis and H. Kopetz, *Dependability: Basic Concepts and Terminology.* Springer-Verlag New York, Inc., 1992.
[74] M. Stamatelatos, W. Vesely, J. Dugan, J. Fragola, J. Minarick and J. Railsback, "Fault Tree Handbook with Aerospace Applications, Version 1.1," *National Aeronautics and Space Administration,* 2002.
[75] M. E. Pat-Cornell, "Fault trees vs. event trees in reliability analysis," *Risk Analysis,* vol. 4, pp. 177-186, 1984.
[76] M. Stamatelatos, H. Dezfuli, G. Apostolakis, C. Everline, S. Guarro, D. Mathias, A. Mosleh, T. Paulos, D. Riha and C. Smith, "Probabilistic risk assessment procedures guide for NASA managers and practitioners," 2011.
[77] M. Tripp, H. Bradley, R. Devitt, G. Orros, G. Overton, L. Pryor and R. Shaw, "Quantifying operational risk in general insurance companies," *British Actuarial Journal,* vol. 10, pp. 919, 2004.
[78] R. Cowell, R. Verrall and Y. Yoon, "Modeling operational risk with bayesian networks," *J. Risk Insur.,* vol. 74, pp. 795-827, 2007.
[79] S. Sklet, "Comparison of some selected methods for accident investigation," *J. Hazard. Mater.,* vol. 111, pp. 29-37, 2004.
[80] M. Rausand and A. Hoylanc, "System Reliability Theory: Models and Statistical Method," *Annals of Statistics,* vol. 6, pp. 701-726, 2004.
[81] K. Campbell, L. A. Gordon, M. P. Loeb and L. Zhou, "The economic cost of publicly announced information security breaches: empirical evidence from the stock market," *Journal of Computer Security,* vol. 11, pp. 431-448, 2003.
[82] A. Shiravi, H. Shiravi, M. Tavallaee and A. A. Ghorbani, "Towards developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.,* 2012.

[83] D. M. Cappelli, A. P. Moore and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud).* Addison-Wesley Professional, 2012.

[84] D. Coit, K. Dey and W. Turkowski, "Practical reliability data and analysis," *Reliability Engineering,* vol. 14, pp. 1-17, 1986.

[85] M. C. Paulk, B. Curtis, M. B. Chrissis and C. V. Weber, "Capability Maturity Model SM for Software, Version 1.1," *Pittsburgh, PA, Software Engineering Institute,* vol. 82, 1993.

[86] M. Dacier, "Towards Quantitative Evaluation of Computer Security," *Institut National Polytechnique De Toulouse,* 1994.

[87] IEC (International Electrotechnical Commission), "International Vocabulary, Chapter 191: Dependability and Quality of Service, IEC 50 (191)," 1991.

[88] B. Daniels, "Software reliability," *Reliability Engineering,* vol. 4, pp. 199-234, 1983.

[89] J. D. Musa, "The measurement and management of software reliability," *Proc IEEE,* vol. 68, pp. 1131-1143, 1980.

[90] G. R. Hudson, "Program errors as a birth and death process," *System Development Corp., Report SP-3011,* 1967.

[91] Z. Jelinski and P. B. Moranda, "Software reliability research," *Statistical Computer Performance Evaluation,* pp. 465-484, 1972.

[92] J. D. Musa, A. Iannino and K. Okumoto, *Software Reliability.* McGraw-Hill, Inc. New York, NY, USA, 1987.

[93] A. Costes, C. Landrault and J. C. Laprie, "Reliability and availability models for maintained systems featuring hardware failures and design faults," *Computers, IEEE Transactions on,* vol. 100, pp. 548-560, 1978.

[94] K. Suzuki, M. Alam, T. Yoshikawa and W. Yamamoto, "Two methods for estimating product lifetimes from only warranty claims data," in *Secure System Integration and Reliability Improvement, 2008. SSIRI '08. Second International Conference on,* 2008, pp. 111-119.

[95] J. Kalbfleisch, J. Lawless and J. Robinson, "Methods for the analysis and prediction of warranty claims," *Technometrics,* vol. 33, pp. 273-285, 1991.

[96] R. E. Barlow, "Using influence diagrams," *Accelerated Life Testing and Experts' Opinions in Reliability,* pp. 145-150, 1988.

[97] R. G. Almond, "An extended example for testing Graphical Belief," *Statistical Science Research Report,* vol. 6, pp. 1-18, 1992.

[98] J. C. Laprie, "Dependability of computer systems: Concepts, limits, improvements," in *Sixth International Symposium on Software Reliability Engineering, Proceedings,* 1995, pp. 2-11.
[99] N. F. Schneidewind, "Reliability- security model," *11th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2006,* pp. 9 pp., 2006.
[100] A. Avizienis, J. C. Laprie and B. Randell, "Fundamental Concepts of Dependability," *TECHNICAL REPORT SERIES-UNIVERSITY OF NEWCASTLE UPON TYNE COMPUTING SCIENCE,* 2001.
[101] R. Drenick, "The Failure Law of Complex Equipment," *Journal of the Society for Industrial and Applied Mathematics,* vol. 8, pp. 680, 1960.
[102] Z. W. Birnbaum, J. D. Esary and S. C. Saunders, "Multi-component systems and structures and their reliability," *Technometrics,* vol. 3, pp. 55-77, 1961.
[103] W. ARINC Research Corporation D.C. and W. H. Von Alven, *Reliability Engineering, Prepared by the Engineering and Statistical Staff of ARINC Research Corporation. Edited by William H. Von Alven.* Englewood Cliffs: N. J, 1964.
[104] B. L. Amstadter, *Reliability Mathematics: Fundamentals, Practices, Procedures.* McGraw-Hill, 1971.
[105] E. J. Henley and H. Kumamoto, *Reliability Engineering and Risk Assessment.* Prentice Hall, 1981.
[106] J. D. Musa, "A theory of software reliability and its application," *IEEE Trans. Software Eng.,* vol. 1, pp. 312-327, 1975.
[107] B. Littlewood and J. Verrall, "A Bayesian reliability growth model for computer software," *Applied Statistics,* pp. 332-346, 1973.
[108] C. E. Landwehr, A. R. Bull, J. P. McDermott and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Computing Surveys (CSUR),* vol. 26, pp. 211-254, 1994.
[109] E. E. Lewis, *Introduction to Reliability Engineering.* New York: J. Wiley, 1996.
[110] P. A. Tobias and D. C. Trindade, *Applied Reliability.* New York: Van Nostrand Reinhold, 1995.
[111] K. E. Murphy, C. M. Carter and S. O. Brown, "The exponential distribution: the good, the bad and the ugly. A practical guide to its implementation," *Annual Reliability and Maintainability Symposium, 2002. Proceedings.,* pp. 550-555, 2002.
[112] R. B. Abernethy, *The New Weibull Handbook.* RB Abernethy, 2006.
[113] B. V. Gnedenko, I. A. Ushakov and J. Falk, *Probabilistic Reliability Engineering.* New York: Wiley, 1995.

[114] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models.* DTIC Document, 1975.
[115] A. Birolini, *Quality and Reliability of Technical Systems: Theory, Practice, Management.* Springer, 1997.
[116] G. Levitin, *The Universal Generating Function in Reliability Analysis and Optimization.* Springer-Verlag New York Inc, 2005.
[117] A. Lisnianski and G. Levitin, *Multi-State System Reliability: Assessment, Optimization and Applications.* World Scientific Pub Co Inc, 2003.
[118] H. Pham, *Handbook of Reliability Engineering.* Springer Verlag, 2003.
[119] G. Levitin, A. Lisnianski, H. Ben-Haim and D. Elmakis, "Redundancy optimization for series-parallel multi-state systems," *IEEE Transactions on Reliability,* vol. 47, pp. 165-172, 1998.
[120] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design.* New York: Wiley, 2002.
[121] M. L. Shooman, *Software Engineering: Design, Reliability, and Management.* McGraw-Hill, 1983.
[122] C. Lie, C. Hwang and F. Tillman, "Availability of maintained systems: a state-of-the-art survey," *IIE Transactions,* vol. 9, pp. 247-259, 1977.
[123] I. Bazovsky, *Reliability Theory and Practice.* Dover Publications, 2004.
[124] J. Murchland, "Fundamental concepts and relations for reliability analysis of multi-state systems," *Reliability and Fault Tree Analysis,* pp. 581-618, 1975.
[125] E. El-Neweihi, F. Proschan and J. Sethuraman, "Multistate coherent systems," *J. Appl. Prob.,* vol. 15, pp. 675-688, 1978.
[126] W. S. Griffith, "Multistate reliability models," *J. Appl. Prob.,* vol. 17, pp. 735-744, 1980.
[127] T. Aven, "On performance measures for multistate monotone systems," *Reliab. Eng. Syst. Saf.,* vol. 41, pp. 259-266, 1993.
[128] I. Ushakov, "Universal generating function," *Sov J Comput Syst Sci,* vol. 24, pp. 118–129, 1986.
[129] I. A. Ushakov and R. A. Harrison, *Handbook of Reliability Engineering.* Wiley-Interscience, 1994.
[130] A. Lisnianski, G. Levitin, H. Ben-Haim and D. Elmakis, "Power system structure optimization subject to reliability constraints," *Electr. Power Syst. Res.,* vol. 39, pp. 145-152, 1996.
[131] W. Feller, "An Introduction to Probabability Theory and Its Applications, vol. 1," 1970.

[132] R. E. Barlow and K. D. Heidtmann, "Computing k-out-of-n System Reliability," *Reliability, IEEE Transactions on,* vol. R-33, pp. 322-323, 1984.
[133] G. Stoneburner, A. Goguen and A. Feringa, "Risk management guide for information technology systems," *Nist Special Publication,* vol. 800, pp. 800-830, 2002.
[134] J. Heiser and M. Nicolett, "Assessing the security risks of cloud computing," *Gartner Report,* 2008.
[135] Y. Chen, V. Paxson and R. H. Katz, "What's new about cloud computing security?" *University of California, Berkeley Report no.UCB/EECS-2010-5 January,* vol. 20, pp. 2010-2015, 2010.
[136] S. Pearson, "Taking account of privacy when designing cloud computing services," in *Software Engineering Challenges of Cloud Computing, 2009. CLOUD'09. ICSE Workshop on,* 2009, pp. 44-52.
[137] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and I. Stoica, "A view of cloud computing," *Commun ACM,* vol. 53, pp. 50-58, 2010.
[138] L. M. Kaufman, "Data security in the world of cloud computing," *Security & Privacy, IEEE,* vol. 7, pp. 61-64, 2009.
[139] ISO/IEC 27005, "Information technology — Security techniques — Information security risk management," 2011.
[140] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth and J. Cáceres, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development,* vol. 53, pp. 4: 1-4: 11, 2009.
[141] N. Friedman, M. Linial, I. Nachman and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology,* vol. 7, pp. 601-620, 2000.
[142] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Prentice hall, 2010.
[143] R. Anderson and T. Moore, "The Economics of Information Security," *Science,* vol. 314, pp. 610-613, 2006.
[144] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev,* pp. 167-256, 2003.
[145] H. Boudali and J. B. Dugan, "A discrete-time Bayesian network reliability modeling and analysis framework," *Reliab. Eng. Syst. Saf.,* vol. 87, pp. 337-349, 2005.
[146] A. Bobbio, L. Portinale, M. Minichino and E. Ciancamerla, "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks," *Reliab. Eng. Syst. Saf.,* vol. 71, pp. 249-260, 2001.

[147] E. S. Money, K. H. Reckhow and M. R. Wiesner, "The use of Bayesian networks for nanoparticle risk forecasting: Model formulation and baseline evaluation," *Sci. Total Environ.,* 2012.
[148] C. Bonafede and P. Giudici, "Bayesian networks for enterprise risk assessment," *Physica A: Statistical Mechanics and its Applications,* vol. 382, pp. 22-28, 2007.
[149] S. S. Alaboodi and G. B. Agnew, "A Novel Inference-based Approach to Evaluate Failure Interdependency in Access Control Models," *International Journal of Intelligent Computing Research (IJICR),* vol. 3, Sep/Dec 2012.
[150] S. S. Alaboodi and G. B. Agnew, "Intelligent secure autonomous systems (ISAS): Concept and modeling of failure and protection interdependencies using asset-control graphs," in *2011 Symposium on Advances in Intelligent Systems- Abstract and Poster,* Waterloo, Canada, 2011.
[151] S. S. Alaboodi and G. B. Agnew, "Bayesian networks for modeling failure dependency in access control models," in *2012 World Congress on Internet Security (WorldCIS),* Guelph, ON, Canada, 2012, pp. 176-182.
[152] ISO/IEC 13335-1, "Information technology -- Security techniques -- Management of information and communications technology security -- Part 1: Concepts and models for information and communications technology security management," 2004.
[153] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann, 2011.
[154] B. T. Contos, *Physical and Logical Security Convergence: Powered by Enterprise Security Management.* Syngress Publishing, 2007.
[155] B. B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," *Performance Evaluation,* vol. 56, pp. 167-186, 2004.
[156] A. H. Maslow, "A theory of human motivation." *Psychol. Rev.,* vol. 50, pp. 370, 1943.
[157] R. L. Nolan, "Managing the computer resource: a stage hypothesis," *Commun ACM,* vol. 16, pp. 399-405, 1973.
[158] P. B. Crosby, *Quality is Free: The Art of Making Quality Certain.* McGraw-Hill New York, 1979.
[159] W. S. Humphrey, *Managing the Software Process.* Addison-Wesley Reading, MA, 1989.
[160] M. C. Paulk, *The Capability Maturity Model: Guidelines for Improving the Software Process.* Addison-Wesley Reading, MA, 1995.

[161] S. S. Alaboodi, "Proposal of New Approach for Assessing the Maturity of Information Security," *Master's Thesis, Hull University, UK,* 2003.
[162] SANS Institute, "Critical controls for effective cyber defense: Version 4.0," SANS Institute, 2009.
[163] R. Ross, S. Katzke, A. Johnson, M. Swanson, G. Stoneburner, G. Rogers and A. Lee, "Recommended security controls for federal information systems," *NIST Special Publication,* vol. 800, pp. 53, 2005.
[164] G. Levitin, "A universal generating function approach for the analysis of multi-state systems with dependent elements," *Reliab. Eng. Syst. Saf.,* vol. 84, pp. 285-292, 2004.
[165] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.
[166] K. Murphy, "The bayes net toolbox for matlab," *Computing Science and Statistics,* vol. 33, pp. 1024-1034, 2001.
[167] Z. W. Birnbaum, "On the importance of different components in a multicomponent system ," 1968.
[168] E. G. Frankel, *Systems Reliability and Risk Analysis.* Dordrecht: Kluwer Academic, 1988.
[169] W. R. Blischke and D. N. P. Murthy, *Reliability: Modeling, Prediction, and Optimization.* Wiley-Interscience, 2000.
[170] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications.* Macmillan London, 1976.
[171] D. B. West, *Introduction to Graph Theory.* Prentice hall Englewood Cliffs, 2001.
[172] G. Strang and K. Aarikka, *Introduction to Applied Mathematics.* Wellesley-Cambridge Press Wellesley, MA, 1986.
[173] M. Minoux and S. Vajda, *Mathematical Programming: Theory and Algorithms.* Wiley New York, 1986.