Improved Internet Protocol Multimedia Subsystem Authentication for Long Term Evolution

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Engineering

Lili Gu
B.Eng.

ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF SCIENCE, ENGINEERING AND HEALTH
RMIT UNIVERSITY

DECEMBER 2011

Dedicated to my parents, my husband and my daughters

Preface

Abstract

Long Term Evolution (LTE) is a major technology to be used in the 4th generation (4G) mobile network and the core network is evolving towards a converged packet based framework for all services. As a part of the evolved core network, Internet Protocol (IP) Multimedia Subsystem (IMS) provides multimedia services (data, voice, video and variations) over packet switched networks. LTE and IMS are both defined by the 3rd Generation Partnership Project (3GPP) group, and the specification identifies that a LTE user device has to carry out two authentication steps to access IP multimedia services. The first authentication step is used to gain LTE network admission and the second authentication step is the IMS authentication used to gain access to the multimedia services. It is observed that the 4G standardized authentication protocols include double execution of the Authentication and Key Agreement (AKA) which increases the system's complexity, results in significant authentication delay and high terminal energy consumption. Authentication is very important for a terminal to gain access to a network and therefore considerable previous research into this topic has occurred. However a common limitation of previously proposed authentication systems is either a lack of security or significant system modification.

This research proposes the Improved AKA (IAKA) authentication protocol which binds the two layer's authentication procedures by using the unified IP Multimedia Private-user Identity (IMPI). The proposed IAKA only executes the AKA protocol once in the network layer and generates authentication credentials which would be used in the second IMS service layer authentication.

This research work included providing IAKA authentication protocol, developing a LTE IMS integrated network by using OPNET Modeller, simulation of the IAKA and the legacy 3GPP defined 4G LTE AKA authentication protocol under different environments, and in-depth analysis of the system performance, security and terminal's energy consumption. It is shown that the proposed IAKA carries out

terminal authentication correctly, improves security, reduces IMS layer authentication delay by up to 38%, and provides an 81.82% terminal energy consumption saving.

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Signature: _____

Lili Gu

Date: 15/12/2011

Acknowledgement

This research would not have been possible without the support of my supervisor. I am deeply indebted and grateful to my supervisor, Dr. Mark Gregory from the School of Electrical and Computer Engineering for his patient guidance, encouragement, suggestions and support during the progress and realisation of the research. This also extends to all the staff of the School of Electrical and Computer Engineering and RMIT University.

I would also like to thank to my parents, my husband and my daughters for all their support and patience during the time it took to complete this thesis.

Table of Contents

Preface iii
Abstract iii
Declaration v
Acknowledgement vi
Table of Contents vii
Table of Figures xi
Table of Tables xiv
List of Abbreviations xv
List of definitions xix
1 Introduction 1
1.1 Scope 2
1.2 Purpose 4
1.3 Publications 4
1.4 Thesis Outline 5
2 Background 6
2.1 Evolution of the mobile network 6
2.1.1 1G and 2G 6
2.1.2 3G 7
2.1.3 4G 7
2.2 The 3GPP group 8
2.3 IP Multimedia Subsystem 9
2.4 Long Term Evolution 10
2.4.1 Network architecture 11
2.4.2 Reference Points 12
2.5 4G LTE two-pass AKA authentication 13
2.5.1 LTE key hierarchy 13
2.5.2 IMS key hierarchy 14
2.5.3 EPS AKA authentication 15
2.5.4 IMS AKA authentication 17
2.5.5 Two-pass AKA authentication 19
2.5.6 Early IMS authentication 20
2.6 Related work 20

2.6.1 Huang and Li 20
2.6.1 Ntantogian 23
3 Objectives 26
3.1 Research Limitations 27
3.2 Assumption 28
4 Improved one pass authentication protocol 30
4.1 The proposed IAKA authentication protocol 30
4.1.1 Outline 30
4.1.2 The proposed IMS key hierarchy 31
4.1.3 Improved LTE EPS AKA authentication 32
4.1.4 The improved IMS AKA authentication 33
4.2 System Model 36
4.2.1 OPNET Modeller 36
4.2.2 Model Methodology 39
4.2.3 Network Topology 40
4.2.4 Network Scenarios 42
4.2.5 Application 45
4.2.6 SIP registration 47
4.2.6.1 sip_UAC 50
4.2.6.2 sip_UAS 51
4.2.7 Diameter server 55
4.2.7.1 Diameter packets 56
4.2.7.2 Diameter_UAS_mgr 58
4.2.7.3 Diameter_UAS 58
4.2.8 AKA authentication protocol 60
4.2.9 Key derivation of $K_{PCSCFenc}$ and $K_{PCSCFint}$ 63
4.2.10 IPSEC 63
4.2.11 Other system processes 63
4.2.12 System Load 67
4.2.13 Statistics 67
4.2.14 Node Configuration 68
4.2.15 Run Simulation 70
4.2.16 Simulation Results 70

4.3 Summary 70
5 Analysis 71
5.1 Performance Analysis 71
5.1.1 Fixed Network Delay Configuration 71
5.1.1.1 System Load SPT 1 71
5.1.1.2 System Load SPT 20 72
5.1.1.3 System Load SPT 30 74
5.1.1.4 System Load SPT 40 76
5.1.1.5 Summary 77
5.1.2 Varying Network Delay Configuration 80
5.2 Security Analysis 81
5.2.1 Authentication Accuracy 81
5.2.2 Mutual authentication 83
5.2.3 Confidential and integrity protection 83
5.2.4 Possible attacks 84
5.3 Energy Consumption Analysis 85
6 Conclusion 88
7 Future Work 91
References: 92
Appendix A 98
A.1: Source Code for application SIP REGISTER 98
A.1.1: Function “sip_request_register” 98
A.1.2: Function “sip_request_register_finish” 99
A.2: Source Code for DIAMETER_UAS_mgr model 100
A.3: Source Code for DIAMETER_UAS 103
A.4: Source Code for AKA authentication 116
Appendix B 127
I. INTRODUCTION 127
II. Related Work 128
III. 3GPP two-pass authentication 128
IV. Improved one-pass authentication 130
V. IAKA AUTHENTICATION SIMULATION 131
VI. Security Analysis 132

VII. IAKA authentication 133
VIII. Conclusion 133
IX. References 133
Appendix C 135
I. introduction 135
II. Related Work 136
III. 3GPP security architecture 136
IV. Improved AKA Authentication 137
V. Energy Cost Analysis 139
VI. Security Analysis 140
VII. Conclusion 142
VIII. References 142
Appendix D 143
I. Introduction 143
II. Related Work 144
III. Background 145
IV. Improved One-pass Authentication 147
V. Analysis 148
VI. Conclusion 153
VII. References 153

Table of Figures

Figure 2-1 IMS three layer architecture 9

Figure 2-2 LTE and IMS integrated system architecture 11

Figure 2-3 LTE key hierarchy 13

Figure 2-4 IMS key hierarchy 14

Figure 2-5 LTE EPS AKA authentication procedure 15

Figure 2-6 IMS AKA authentication procedure 17

Figure 2-7 One pass IMS AKA [Huang 2009] 21

Figure 2-8 Proposed IMS AKA for 3G-WLAN in [Ntantogian 2010] 24

Figure 4-1 the principle of the improved IAKA authentication protocol 30

Figure 4-2 Proposed IMS key hierarchy 31

Figure 4-3 Improved LTE EPS AKA authentication procedure 32

Figure 4-4 the improved IMS AKA authentication procedure 34

Figure 4-5: OPNET hierarchical GUI editors 38

Figure 4-6 OPNET Modeller development methodology 39

Figure 4-7 Simulation network topology 40

Figure 4-8 Node attribute “Authentication Mode” configuration 43

Figure 4-9: Node attribute “Authentication Mode” definition 44

Figure 4-10 Application “REGISTER” process diagram 45

Figure 4-11: Voice application state transition diagram 46

Figure 4-12: SIP packet format 47

Figure 4-13: SIP registration diagram of SIP UAC 50

Figure 4-14: sip_UAC state transition diagram 51

Figure 4-15: sip_UAS SIP registration diagram 52

Figure 4-16: SIP packet process 53

Figure 4-17: Diameter receiving process 54

Figure 4-18: sip_UAS state transition diagram 55
Figure 4-19: ethernet_server_adv node model 56
Figure 4-20: Diameter packet format 56
Figure 4-21: State transition diagram of process Diameter_UAS_mgr 58
Figure 4-22: Process diagram of Diameter_UAS 59
Figure 4-23: Diameter_UAS state transition diagram 60
Figure 4-24: f1, f1*, f2, f3, f4, f5, and f5* definition 60
Figure 4-25 Configuration of Node Attribute “delay loop” 65
Figure 4-26 Configuration of system processing delay of sip_proxy_server 66
Figure 4-27 Configuration of system processing delay of HSS model 66
Figure 4-28 Configuration of SPT 67
Figure 4-29 Authentication delay 67
Figure 4-30 EPS bearer “platinum” configuration 69
Figure 4-31 UE EPS bearer configuration 69
Figure 4-32 Simulation runtime configuration 70
Figure 5-1 Authentication Delay (SPT = 1) 72
Figure 5-2 Authentication Delay Saving Rate (SPT = 1) 72
Figure 5-3 Authentication Delay (SPT = 20) 73
Figure 5-4 Authentication Delay Saving Rate (SPT = 20) 74
Figure 5-5 Authentication Delay (SPT = 30) 75
Figure 5-6 Authentication Delay Saving Rate (SPT = 30) 76
Figure 5-7 Authentication Delay (SPT = 40) 77
Figure 5-8 Authentication Delay Saving Rate (SPT = 40) 77
Figure 5-9 IMS layer Authentication Delay under fixed network delay configuration 78
Figure 5-10 IMS layer Authentication Delay Saving Rate under fixed network delay configuration 79
Figure 5-11 Authentication Delay with varying network delay configuration 80

Figure 5-12 ADSR with varying network delay configuration 81

Figure 5-13 Energy Consumption Comparison..... 87

Table of Tables

Table 2-1 Comparison of LTE, LTE advanced, and IMT advanced 10

Table 2-2 Huang and Li and 3GPP defined IMS AKA authentication procedure comparison 22

Table 4-1 Algorithm type distinguishers 31

Table 4-2 IMS LAYER AUTHENTICATION COMPARISON 35

Table 4-3 The Node Model and Link Model used in the simulation 42

Table 4-4: Node Attribute “Authentication Mode” definition 43

Table 4-5: Attribute “Traffic Modelling” definition 45

Table 4-6: The definition of Node Attributes “delay loop” 64

Table 4-7: Standardized QCI characteristics [TS 23.203 2011] 68

Table 5-1 Average simulation results (SPT = 20) 73

Table 5-2 Average simulation results (SPT = 30) 75

Table 5-3 Average simulation results (SPT = 40) 76

Table 5-4 Average simulation results during simulation period (500 second~599 second) 78

List of Abbreviations

1G	first generation
2G	second generation
3G	third generation
3GPP	Third generation partnership project
4G	fourth generation
ADSR	Authentication Delay Saving Rate
AES	Advanced Encryption Standard
AIA	Authentication Information Answer
AIR	Authentication Information Request
AKA	Authentication and Key Agreement
AMF	Authentication Management Field
AMPS	Advanced Mobile Phone System
ARIB	the Association of Radio Industries and Businesses
ARPU	Average Revenue Per User
AS	Application Server
ATIS	the Alliance for Telecommunications Industry Solutions
AuC	Authentication Center
AV	Authentication Vector
B3G	Beyond 3rd Generation
BSS	Business Support System
CCSA	China Communications Standards Association
CDMA	Code Division Multiple Access
CK	Cipher Key
CRM	Customer Relationship Management
CSCF	Call Session Control Function
DOS	Denial Of Service
DS	Direct Speed
eNB	Evolved Node B
EPB	Energy consumption Per Byte
EPC	Evolved Packet Core
EPS	Evolved Packet System
ESP	Encapsulating Security Payload

ESR	Energy Saving Rate
ETSI	the European Telecommunications Standards Institute
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EV-DO	Evolution Data Optimized
FDD	Frequency Division Duplex
FMC	Fixed Mobile Convergence
GSM	Global System for Mobile Communications
GTP	General packet radio services Tunnelling Protocol
GUTI	Global Unique Temporary Identity
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
I-CSCF	Interrogating CSCF
IAKA	Improved Authentication and Key Agreement protocol
ICI	Interface Control Information
ICV	Integrity Checking Value
IETF	Internet Engineering Task Force
IK	Integrity Key
IMPI	IP Multimedia Private-user Identity
IMPU	IP Multimedia Public User identity
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IMT	International Mobile Telecommunications
IP	Internet Protocol
IPSec	Internet Protocol Security
ITU	International Telecommunication Union
KDF	Key Derivation Function
KSI	Key Set Identifier
LTE	Long Term Evolution
MAC	Message Authentication Code
MC	Multi Carrier
MME	Mobility Management Entity
NAS	Non Access Stratum
NGN	Next Generation Network

NMT	Nordle Mobile Telephone
OMA	Open Mobile Alliance
OSI	Open Systems Interconnection
P-CSCF	Proxy CSCF
PDC	Personal Digital Cellular
PDG	Packet Data Gateway
PDN	Packet Data Network
PGW	PDN Gateway
PSN	Packet Switched Network
PSTN	Public Switched Telephone Network
QCI	QOS Class Identifier
QOS	Quality of Service
S-CSCF	Serving CSCF
SA	Security Association
SAA	Server Assignment Answer
SAR	Server Assignment Register
SGW	Serving Gateway
SIP	Session Initiation Protocol
SPI	Security Parameters Index
SPT	Simultaneous rate Per Terminal
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TFT	Traffic Flow Template
TIA	Telecommunication Industry Association
TISPAN	Telecommunication and Internet converged Services and Protocols for Advanced Networking
ToS	Type of Service
TTA	Telecommunications Technology Association
TTC	Telecommunication Technology Committee
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UE	User Equipment

UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
WiMAX	Worldwide Interoperability for Microwave Access

List of definitions

4G LTE authentication	in this paper, 4G LTE authentication denotes the current 4G LTE IMS two-pass authentication protocol defined by 3GPP standardization group
IAKA authentication	in this paper, IAKA authentication denotes the proposed the improved 4G LTE IMS one-pass AKA authentication protocol

1 Introduction

Driven by the rapid growth in mobile broadband networks and the insatiable bandwidth requirements of new applications, mobile networks are evolving towards converged Internet Protocol (IP) based 4th generation (4G) mobile broadband networks. As a major 4G technology, Long Term Evolution (LTE) aims to provide a high data rate, low latency and all IP mobile networks. Correspondingly, the core network is evolving towards a converged packet-based framework for all services. As a part of the evolved core network the IP Multimedia Subsystem (IMS) provides multimedia services (data, voice, video, and combinations) over Packet Switched Networks (PSN) for wired and wireless access networks. However, with the integration of different networks, the authentication procedure becomes more and more complex which raises performance concerns and affects IMS and LTE deployment scenarios.

4G LTE and IMS are both defined by the 3rd Generation Partnership Projects (3GPP) group. The 3GPP specification requires that a LTE user terminal authenticate twice to access the multimedia service layer. The first authentication step is the LTE network layer authentication which enables the user to access the LTE network layer and packet data network by checking the user's International Mobile Subscriber Identity (IMSI); and the second authentication step is the IMS service layer authentication which authenticates the user by checking the user's IP Multimedia Private-User Identity (IMPI) to enable the user to access the multimedia service layer. This authentication approach is also called the two-pass Authentication and Key Agreement (AKA) authentication protocol. However, it was observed that the two authentication steps use the same AKA protocol including requesting, generation and distribution of the Authentication Vectors (AV), and the mutual authentication by using the received AVs increases the system's authentication delay and the terminal's energy consumption.

In order to reduce the overhead, delay and energy consumption, an Improved AKA (IAKA) authentication protocol is proposed which binds the two authentication steps by using the user's IMPI number without double execution of the AKA protocol. In the proposed IAKA, the Home Subscriber Server (HSS) generates the AV by using

the IMPI number and distributes the AV to the User Equipment (UE), the Mobility Management Entity (MME), and the Proxy Call Session Control Function (PCSCF). In both of the network layer authentication and the service layer authentication, the terminal and the network entities authenticate each other by using the same AV generated by the HSS using the IMPI number.

The security, performance and energy consumption of the proposed IAKA was analysed and the results are presented. The results demonstrate that the IAKA provides adequate authentication, and supports mutual authentication, confidential and integrity message protection. Furthermore, two possible security attacks are discussed: (1) replay attack; and (2) Denial of Service (DOS) attack. The proposed IAKA algorithm protects the system from the malicious replay attack and provides stronger protection against the DOS attack than the current 4G standardized approach. An OPNET Modeller simulation model was developed to simulate the proposed IAKA algorithm and the 4G LTE authentication protocols. The simulation results showed that the proposed IAKA algorithm could not only simplify the authentication protocol significantly, but also reduce the IMS layer authentication delay by up to 38%. The terminal energy consumption used in the authentication and security activities was calculated and the results showed the proposed IAKA algorithm could save up to 81.82% in the IMS service layer authentication and save up to 39.13% in the combined two layer authentication process.

1.1 Scope

It was observed that the legacy 4G LTE authentication protocol has the following vulnerabilities: (1) duplicated AKA authentication procedures in use; (2) long authentication delay; and (3) high terminal energy consumption. In order to improve the system's efficiency and save terminal energy consumption, the research questions were framed:

- How to avoid the double execution of the AKA protocol?
- How to improve the system's performance by simplifying the authentication process?
- How to provide a safe message channel to protect the critical user identities?

- How to decrease terminal energy consumption?

The research outcome is to design an authentication protocol which would be a good solution to the defined research questions. The scope of this research included:

- A literature review of current research on the authentication protocols.
- An initial investigation of LTE and IMS network concepts and technologies used.
- Detailed research and investigation of IMS authentication in LTE network:
    - LTE network architecture and service procedures
    - LTE network layer AKA authentication
    - IMS network architecture and service procedures
    - IMS service layer AKA authentication
    - LTE and IMS user identity management
    - SIP protocol
    - Diameter protocol
    - IPSec (Internet Protocol Security) protocol
    - Security
    - Energy consumption
- Network simulation by using OPNET Modeller Version 16
    - Developing LTE IMS integrated network
    - Developing the current 3GPP defined 4G LTE IMS registration procedure and the related AKA protocol
    - Developing proposed IAKA registration and the related AKA protocol

- Developing different network environment to run the 3GPP AKA authentication protocol and the IAKA authentication protocol
- Choosing statistics and collecting results

- An in-depth Performance analysis based on the comparison of the simulation results
- An in-depth security analysis
- An in-depth energy consumption analysis

1.2 Purpose

The purpose of this thesis is described as followings:

- To develop an improved authentication protocol which enables the LTE user to access the IP multimedia services with enhanced performance, security and low energy consumption.
- To develop the simulator of the LTE IMS integrated network which would be used to validate the proposed IAKA protocol.
- To simulate different authentication protocols under different environments to compare their performance.
- To compare and analyse the security level of different authentication approaches.
- To compare and analyse the energy consumption of different authentication approaches.

1.3 Publications

The research has been published in peer reviewed IEEE conferences and a journal publication has been submitted for review. The publications are shown below which will be included in the appendix.

- Lili Gu and Mark A Gregory, "Improved One-Pass IP Multimedia Subsystem Authentication for UMTS", International Conference on Information Networking, Kuala Lumpur Malaysia, Jan 2011

- Lili Gu and Mark A Gregory, “A Green and Secure Authentication for the 4Th Generation Mobile Network”, Australasian Telecommunication Networks And Applications Conference, Melbourne Australia, Nov. 2011, accepted

- Lili Gu and Mark A Gregory, “Optimized Authentication and Key Agreement Protocol for 4G Long Term Evolution and IP Multimedia Subsystem”, IEEE Transactions on Wireless Communications, Dec. 2011, submitted

1.4 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 provides a description of the evolution of the mobile network, the 3GPP defined 4G LTE AKA authentication protocol, and the current research work in this area; the research objectives, assumptions and limitations are provided in Chapter 3; Chapter 4 introduces the proposed IAKA authentication protocol including the principles, the authentication procedures, and the detailed description of the system model; Chapter 5 provides a comprehensive analysis of the system performance, system security and terminal’s energy consumption; Chapter 6 provides the conclusion; and possible future research opportunities are identified in Chapter 7.

2 Background

This chapter provides a brief introduction of the mobile network evolution which starts from first generation (1G) to fourth generation (4G). With the focus on 4G LTE and IMS technology, the chapter provides an in-depth description of the LTE and IMS network architecture, and the system's interfaces. A detailed description is provided of the current 4G LTE IMS authentication architecture and relevant protocols. Finally this chapter introduces the most recent research in related areas.

2.1 Evolution of the mobile network

2.1.1 1G and 2G

Wireless cellular communication has evolved since its emergence in the 1980s when the 1G analogue mobile network was commercially introduced. 1G systems included Advanced Mobile Phone System (AMPS) in the United States and Nordle Mobile Telephone (NMT) in the Europe. 1G systems suffered from performance limitations, poor coverage, limited roaming and large handsets; and was replaced by second generation (2G) systems in 1990s. The 2G mobile network was originally designed as a digital, circuit-switched network for voice services. 2G systems included key technologies:

- Global System for Mobile Communications (GSM) was defined by the European Telecommunications Standards Institute (ETSI) and deployed worldwide. GSM was the dominant 2G technology.
- cdmaOne was standardized by the Telecommunications Industry Association (TIA) as an Interim Standard (IS-95) [RAPPAPORT 2002] and deployed in Asia-Pacific, North and Latin America.
- Time Division Multiple Access (TDMA) is an evolution to AMPS and standardized in North America as IS-136. TDMA was mostly used in North America.
- Personal Digital Cellular (PDC) was standardized in Japan and was very similar to TDMA. PDC was mainly used in Japan and China.

2.1.2 3G

Due to demand and increasing capacity requirements for mobile Internet access and IP-based services, the first commercial third generation (3G) network was available in Japan at the end of 2001. Under the International Telecommunication Union (ITU) International Mobile Telecommunications 2000 (IMT-2000) framework, the two main 3G technologies developed included Universal Mobile Telecommunications System (UMTS) and Code Division Multiple Access 2000 (CDMA 2000).

- UMTS: UMTS was defined by the 3GPP and was an evolution of 2G GSM technology to provide improved services. Two radio interfaces were defined; one was CDMA Direct Spread (DS) which corresponds to the Frequency Division Duplex (FDD) mode of UMTS; another was CDMA Time Division Duplex (TDD) which corresponds to the UMTS TDD mode. However, with 2Mbps bitrate, UMTS quickly fell behind meeting the requirements of the fast developing mobile Internet services. Therefore, only several years later, High Speed Packet Access (HSPA) was launched providing a downlink peak rate of 14.4Mbps and uplink data rate of 5.7Mbps. HSPA+ was defined in the 3GPP release 7 and could provide data rates of 84Mbps on the downlink and 22Mbps on the uplink. Telstra, a major carrier in Australia, was the first carrier in the world to launch a HSPA+ mobile network in 2008.
- CDMA 2000 is also known as CDMA Multi Carrier (MC) and was defined by the 3GPP2. CDMA 2000 provided backward compatibility with the 2G cdmaOne technology. CDMA 2000 has evolved considerably over time and some of the revisions include CDMA2000 1X Rev0, CDMA2000 1xEV-DO Rev 0 / Rev A / Rev B / Rev C, CDMA2000 Rev A / Rev B / Rev C / Rev D.

2.1.3 4G

With the emergence of the mobile broadband network, smart mobile phones/ tablet computers, and the open source application platform, the need for bandwidth has doubled every year. In order to meet the evolving user needs, the ITU International Mobile Telecommunications-Advanced (IMT-Advanced) specified requirements for the 4G mobile network and the key points in the specification are [ITU-R M.2134]:

- All IP packet based network
- Enhanced peak data rates to support advanced services and applications (100 Mbit/s for high and 1 Gbps for low mobility)
- Cell spectrum efficiency of downlink up to 3 bit/s/Hz/cell and uplink up to 2.25 bit/s/Hz/cell
- The minimum requirements for peak spectral efficiencies are 15 bit/s/Hz for downlink and 6.75 bit/s/Hz for uplink
- Scalable bandwidth up to and including 40 MHz. Wider bandwidths (e.g. up to 100 MHz) are encouraged
- Control plane latency shall be less than 100ms and user plane latency shall be less than 10ms

By the end of 2009, different technologies from different countries were submitted to the ITU as 4G candidates including the Worldwide Interoperability for Microwave Access (WiMAX) and Long Term Evolution (LTE). In October 2010, the 3GPP defined LTE release 10 & beyond (LTE_advanced) was accepted as a 4G technology which is the successor of the 2G GSM and 3G UMTS technologies. Since 2G GSM and 3G UMTS have a dominant user base, the LTE technology will follow this trend and be prevalent.

2.2 The 3GPP group

The 3GPP unites telecommunication standard bodies, known as “Organization Partners”. The organization partners are the Association of Radio Industries and Businesses (ARIB), the Alliance for Telecommunications Industry Solutions (ATIS), China Communications Standards Association (CCSA), the European Telecommunications Standards Institute (ETSI), the Telecommunications Technology Association (TTA), and Telecommunication Technology Committee (TTC) [3GPP 2011].

The original scope of 3GPP was to produce technical specifications and technical reports for a 3G mobile system, and now it is extended to include the development and maintenance the 4G LTE systems.

2.3 IP Multimedia Subsystem

The IP Multimedia Subsystem (IMS) has been widely adopted as the multimedia system architecture for the three networks: (1) 4G mobile network; (2) Next Generation Network (NGN); (3) and Internet access. The IMS provides multimedia services (i.e., audio, video, text, chat, image, and a combination of them) over Packet Switched Networks (PSN) for multiple access networks such as fiber, LTE, and WiMAX [Copeland 2009]. The IMS supports interoperating with other IP networks and interworking with legacy networks such as the Public Switched Telephone Network (PSTN) and GSM. The IMS also defines the underlying standards, including standards for security, Quality of Service (QoS), resilience, and chargeability.

The IMS was first specified by 3GPP in 2003 for mobile communication session control over IP transport for 3G networks and then it was extended to include the Telecommunication and Internet converged Services and Protocols for Advanced Networking (TISPAN) as a subsystem of NGNs. Most of the IMS protocols are standardized by Internet Engineering Task Force (IETF) including the Session Initiation Protocol (SIP) which has become an important part of many Voice over Internet Protocol (VoIP) systems. Other standardization bodies are also involved in the development of IMS, such as the Open Mobile Alliance (OMA).

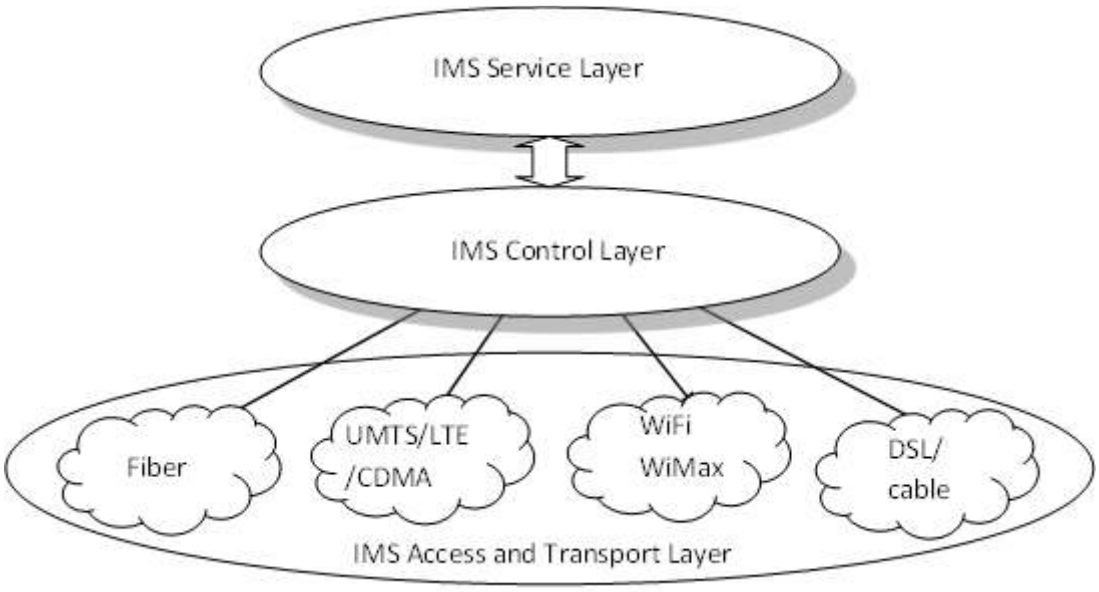

Figure 2-1 IMS three layer architecture

Fig. 2-1 shows the IMS the layer architecture. The three layers are independent with each other and each layer is responsible for different functions which make the IMS an access-agnostic, service-agnostic, and location-agnostic core network. With the IMS, network operators can save costs and increase average revenue per user (ARPU) and subscriber stickiness which increase the incentive for IMS to be deployed. The advantages of IMS can be summarized as follows [Khilifi 2008].

- Integrated delivery of Multimedia Services
- Flexible service environment with rapid service creation & deployment; easy service customization and tailoring
- Fixed-Mobile convergent services (FMC)
- One IMS for multiple access network: Provide same application, single sign-in, consistent user interface, unified charging policy and etc., for different handsets or clients of different access networks by sharing the centralized resources such as Home Subscriber Server (HSS), Customer Relationship Management (CRM), and Business Support System (BSS)
- Support of mobility including service mobility and user mobility for both mobile users and fixed users
- End to end QoS and security

2.4 Long Term Evolution

LTE is defined by the 3GPP and is an evolution of the 3G UMTS technology. Although LTE is always marketed as 4G, the first version of LTE (3GPP release 8) doesn't meet IMT 4G requirements. This pre-4G standard step toward to LTE advanced (3GPP release 10) which increases the speed and capacity of the mobile network, and is backward compatible to the LTE technologies. The key parameters of LTE, LTE advanced, and IMT advanced is compared in Table 2-1.

Table 2-1 Comparison of LTE, LTE advanced, and IMT advanced

	LTE (Rel. 8)	**LTE advanced (Rel. 10)**	**IMT advanced**
Bandwidth (Max)	20MHz	100MHz	40MHz~100MHz

Peak data rate	DL	300Mbps	1Gbps	1Gbps
	UL	75Mbps	500Mbps	
Peak spectrum efficiency	DL	15	30	15
	UL	3.75	15	6.75

2.4.1 Network architecture

To the LTE core network, IMS is the only candidate for the all IP network. Fig. 2-2 shows the architecture of the LTE IMS integrated system [TS 23.401 2010][TS 23.228 2010]. The functionalities of the basic system components are briefly described as following.

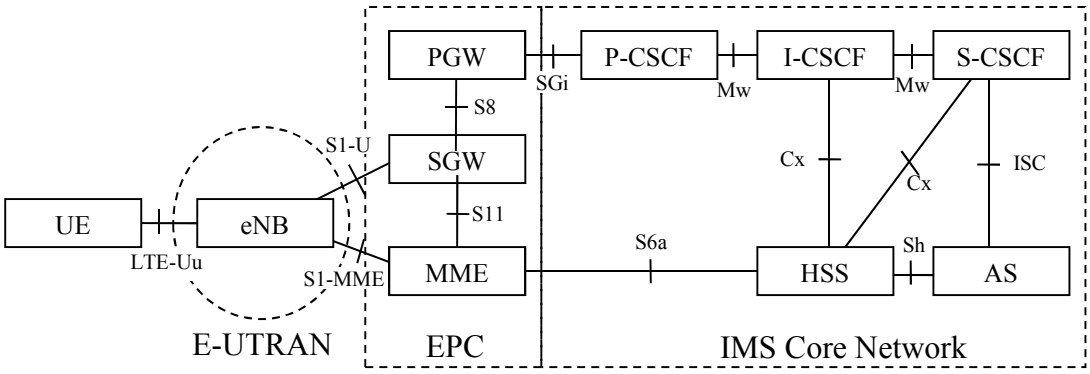

Figure 2-2 LTE and IMS integrated system architecture

- Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Node B (eNB): The E-UTRAN consists of eNBs providing Layer 1 and Layer 2 features of the Open Systems Interconnection (OSI) reference model. Briefly, eNB hosts the functions of the Radio bearer control, Radio resource management, Connection mobility control, Radio admission control, and Dynamic resource allocation

- Mobility Management Entity (MME): The MME is in charge of the Evolved Packet System (EPS) mobility management and EPS session management. Authentication and authorization are two main functions of MME which request authentication vectors from HSS and initiate the mutual authentication procedures between the User Equipment (UE) and the MME
- Serving Gateway (SGW): The SGW serves as a mobility anchor which routes and forwards packets between eNBs and between eNB and Packet Gateway (PGW)
- PGW: The PGW is the Packet Data Network (PDN) Gateway and provides access for sessions to the external Packet Data Network
- Proxy Call Session Control Function (P-CSCF): The P-CSCF is the first contact point in the IMS
- Interrogating Call Session Control Function (I-CSCF): The I-CSCF is used to hide the network's topology and the main task is to identify the relevant Serving Call Session Control Function (S-CSCF)
- S-CSCF: The S-CSCF performs UE session control services
- Home Subscriber Server (HSS): The HSS is the master database for the whole system and contains subscription-related information
- Application Server (AS): The AS provides applications to the end users

2.4.2 Reference Points

The model reference points include:

- LTE-Uu: The radio protocol of E-UTRAN between the UE and the eNodeB
- S1-MME: Reference point between E-UTRAN and MME.
- S1-U: Reference point between E-UTRAN and SGW
- S6a: Reference point between MME and HSS
- S8: Reference point between the PGW and SGW
- S11: Reference point between MME and SGW
- SGi: Refernce point between the PGW and the packet data network
- Mw Reference Point between a CSCF and another CSCF

- Sh Reference Point between an AS and an HSS
- Cx Reference point between a CSCF and an HSS
- ISC Reference Point between a CSCF and an AS

The protocols related to the authentication procedures are:

- The S6a is based on Diameter protocol
- The Mw is based on SIP protocol
- The Cx is based on Diameter protocol
- The protocol between the UE and the MME is NAS (Non Access Stratum) protocol

2.5 4G LTE two-pass AKA authentication

2.5.1 LTE key hierarchy

In order to provide security protection for different traffic flows the 3GPP LTE introduced a five layer key hierarchy.

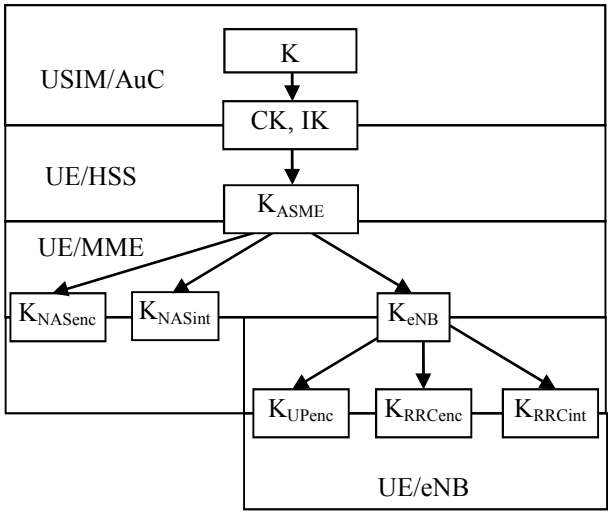

Figure 2-3 LTE key hierarchy

As shown in Fig. 2-3 [TS 33.401 2010], K is the pre-shared key between the UE and the HSS and is used to generate the Cipher Key (CK) and Integrity Key (IK). K is kept in the Universal Subscriber Identity Module (USIM) of the UE and the

Authentication Center (AuC) of the HSS. K_{ASME} is derived from CK and IK by the HSS and sent to the MME; and then, K_{ASME} is stored in the MME and is used to derive five keys to protect three different types of information flows which include the traffic between the terminal and the MME, the traffic between the terminal and the eNB ; and the traffic between the terminal and the SGW[Lescuyer 2008].

NAS [TS 24.301 2010] signaling between the terminal and the MME is confidentiality and integrity protected by K_{NASenc} and K_{NASint}. The traffic between the terminal and the eNB is confidentiality and integrity protected by K_{RRCenc} and K_{RRCint}. The user plane data between the terminal and the SGW is protected by K_{UPenc}.

2.5.2 IMS key hierarchy

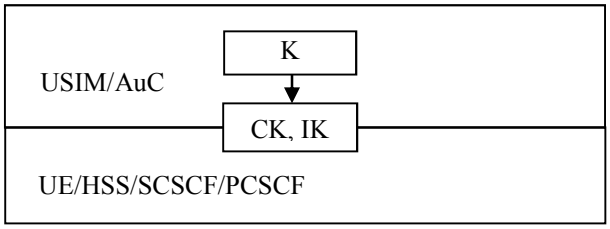

Figure 2-4 IMS key hierarchy

As shown in Fig. 2-4, the IMS uses the two level key hierarchy to protect traffic which is defined in [TS 33.203 2009] and is a subset of the LTE five level key hierarchy. In the IMS, the HSS generates CK and IK by using the pre-shared key K and distributes CK and IK to the S-CSCF and P-CSCF. The traffic between the terminal and the P-CSCF is confidentiality and integrity protected by CK and IK.

2.5.3 EPS AKA authentication

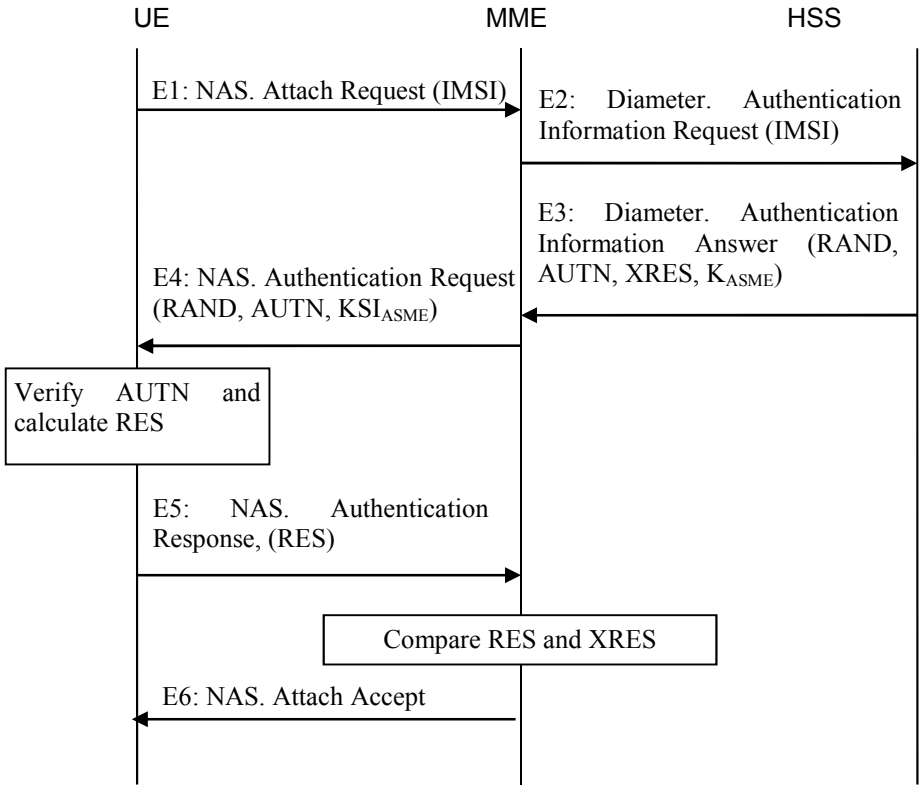

Figure 2-5 LTE EPS AKA authentication procedure

According to the 3GPP TS 33.401, the purpose of EPS AKA is to provide mutual authentication between the terminal and the network and to agree on a key K_{ASME}. The triggering of the EPS AKA procedure occurs for several reasons such as a tracking area update. In most cases, the EPS AKA is performed in the attach procedure when the user terminal is switched on and requesting access to the packet services.

The procedure of EPS AKA is shown in Fig. 2-5 which is based on [TS 23.401 2010][TS 33.102 2009][TS 33.401 2010];

- E1: In order to register with the LTE network to receive services, the UE initiates an Attach Request to the MME with the IMSI number through the NAS protocol;
- E2: If there aren't valid AV in the MME, the MME will send an Authentication Information Request (AIR) to HSS to fetch the AV by using the diameter protocol [TS 29.272 2010];

- E3: Upon receipt of the request, the HSS uses the IMSI number to fetch the user's profile and generates AV (RAND, AUTN, XRES, K_{ASME}); then the HSS sends the AV back to the MME by using the Authentication Information Answer (AIA) command
- E4: Upon receipt of the AV from the HSS, the MME initiates the authentication procedure by sending an Authentication Request to the UE which contains parameters including RAND, AUTN and KSI_{ASME}. The Key Set Identifier (KSI) is used to identify K_{ASME}.
- E5: The UE checks the received AUTN number first. If the AUTN can be accepted, the UE considers that the MME has passed the authentication. After the successful AUTN verification, the UE will calculate the RES number by using RAND and AUTN, and then sends the RES number back to the MME by using the Authentication Response message;
- E6: The MME checks whether the RES from UE match the RES from the HSS, and if they match, the MME considers that the AKA exchange is successfully completed and sends back the Attach Accept message to the UE.

2.5.4 IMS AKA authentication

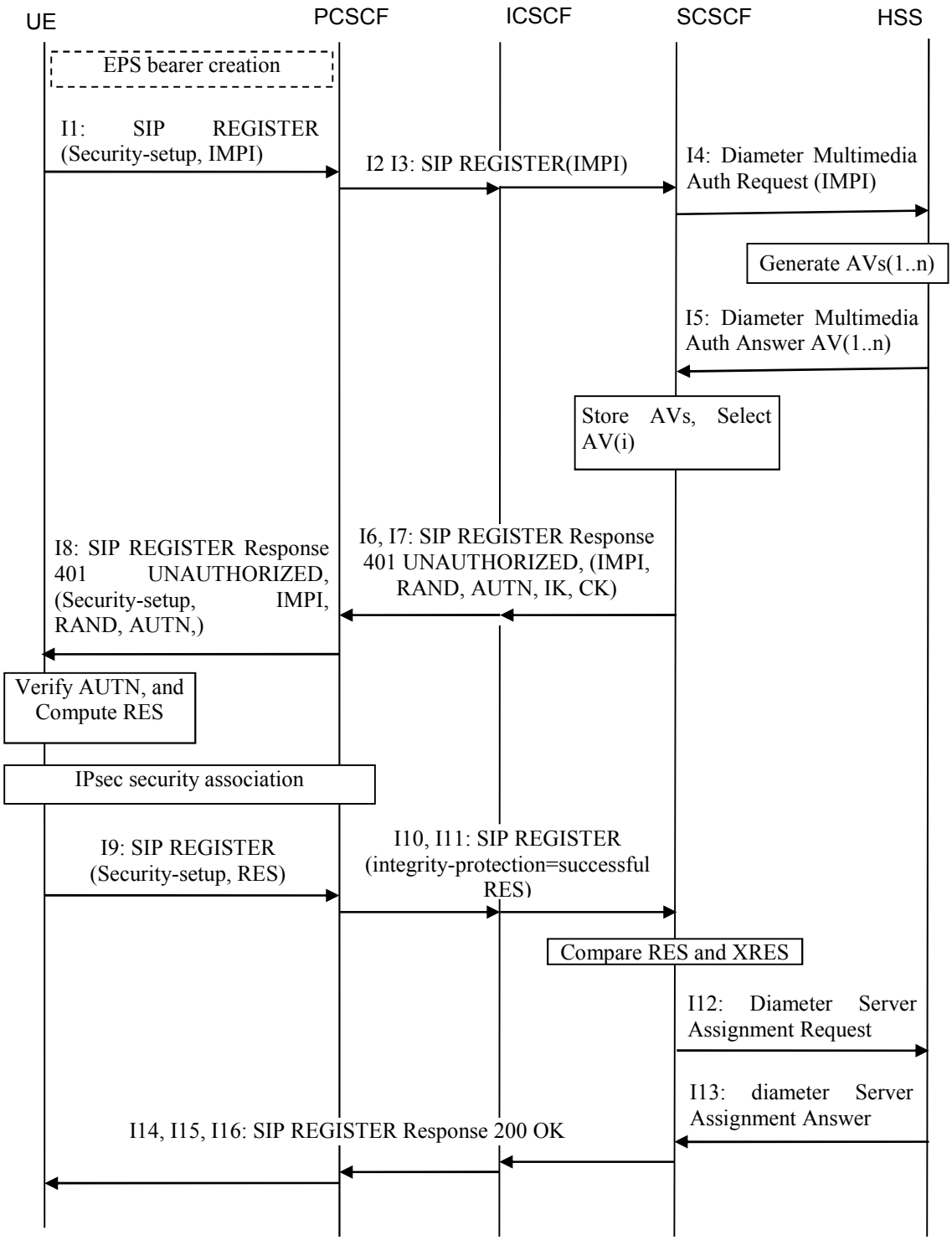

Figure 2-6 IMS AKA authentication procedure

After a successful LTE network layer authentication and the creation of the EPS bearer, the terminal is eligible to access the packet network. In order to access the services, the user has to pass the IMS layer authentication and the procedures are shown in Fig. 2-6 [TS 23.228 2010] [TS 33.102 2009] [TS 33.203 2010].

- I1: After the successful creation of the EPS bearer, the UE sends a REGISTER request to the P-CSCF by using the SIP protocol [TS 24.229 2010]. The SIP REGISTER request has the parameters of IMPI = impi and security-setup line in which the security–setup line is used to create Security Associations (SAs) with the P-CSCF.
- I2, I3: The P-CSCF stores the security-setup parameters; removes the security-setup line from the packet and forwards it to the I-CSCF. The I-CSCF fetches the user and the S-CSCF information from the HSS, locates the address of the next hop S-CSCF and forwards the packet to the S-CSCF.
- I4: After receiving the SIP REGISTER request, the S-CSCF checks whether it has a previously stored AV. If it hasn't, the S-CSCF sends a Multimedia Auth Request to the HSS with IMPI=impi by using the diameter protocol [TS 29.228 2010] [TS29.229 2010];
- I5: The HSS uses the IMPI number to fetch the user's information and generate AVs (RAND, XRES, CK, IK, AUTN); then it sends the AVs back to the S-CSCF;
- I6, I7: The S-CSCF stores the AVs, selects the next AV(i) and sends the AV back to the P-CSCF through the I-CSCF;
- I8: The P-CSCF keeps the received AV, selects parameters for creating the SAs and sends the packet with the parameters of Security-setup line, IMPI, RAND, and AUTN to the UE;
- I9: After receiving the SIP REGISTER response, the UE checks the validity of the AUTN. If this is a valid AUTN, the UE would generate the response number RES, calculate the CK and the IK, create the IPSec security association to protect the following SIP messages, then sends SIP REGISTER message to the P-CSCF with the parameters Security-setup and RES in the security channel;

- I10, I11: After the received packet passed the validity checking of the P-CSCF, the P-CSCF would send the SIP REGISTER packet to the S-CSCF through the I-CSCF with RES and integrity-protection=successful
- I12: The S-CSCF compares the received RES from the UE with the XRES received from the HSS. If they match, the S-CSCF considers the AKA is successfully completed and sends Server Assignment Request to the HSS;
- I13, I14, I15, and I16: After receiving the response from the HSS, the S-CSCF sends SIP REGISTER response 200 OK to the UE through the I-CSCF and the P-CSCF.

2.5.5 Two-pass AKA authentication

The two-pass authentication process is a consequence of the integration of different systems. In the IMS system, each subscriber has two identities, one is for the access layer (e.g, IMSI for 4G mobile network), and one (IMPI number) is for the IP packet layer. The IMS two-pass authentication is used to authenticate the two subscriber identities. In the 4G LTE network, the first authentication step is the LTE layer authentication which enables the user to access the LTE and packet data network by checking the user's IMSI number; and the second authentication step is the IMS service layer authentication which authenticates the user by checking the user's IMPI number to enable the user to access the multimedia service layer.

IMS was first specified in 2003 for 3G network when the data services and voice services were separated. Two identifiers (IMSI and IMPI) were required for different services in that age, IMSI is for the radio access layer and the voice services and IMPI is for the IP packet layer and the multimedia services. In order to keep the two systems fully independent it is necessary to carry out IMS AKA authentication in the 3G mobile network, although the AKA protocol in the two authentication steps are identical. If the system does the 3G network layer authentication and bypasses the IMS service layer authentication it would cause fraudulent IMS usage which is analysed by Lin [Lin 2005].

However, the 4G LTE network is an all IP-based network, and IMS is the only candidate for the core network, since all of the services including the voce services and data services are all based on IP network, the authentication process of using two subscriber identities to do two layers authentication is becoming not so necessary. The

management of two user identities and running two identical authentication protocols increase the management cost and reduce the system performance. Under such background, this research work is trying to provide a novel one-pass authentication protocol without complementary of the security.

2.5.6 Early IMS authentication

In order to have an easier and quicker introduction of the IMS, the early IMS authentication defined in [TR 33.978 2008] could avoid double execution of the AKA protocol by IP address binding with the IMPI, IMPU (IMS Public User identity), and IMSI. The early IMS authentication could protect the IMS against the most significant security threat; however, it suffers from several limitations:

- Early IMS authentication lacks service flexibility, only one contact IP address can be associated with one IMPI;
- Early IMS authentication compromised security since it doesn't support mutual authentication and it does not include IPSec protection at the IMS level.

2.6 Related work

Since authentication is very important to telecommunication networks, there has been considerable research carried out in this area. However the common limitation is either a lack of security or the need for extra modifications of existing systems. Furthermore, 4G network research is current and ongoing. In this section, two related research outcomes are analysed.

2.6.1 Huang and Li

Huang and Li [Huang 2009] proposed a one-pass IMS AKA to replace the 3GPP defined 3G UMTS two-pass AKA authentication which could save 45% of the authentication signalling and 76.5% of the storage space without compromising security.

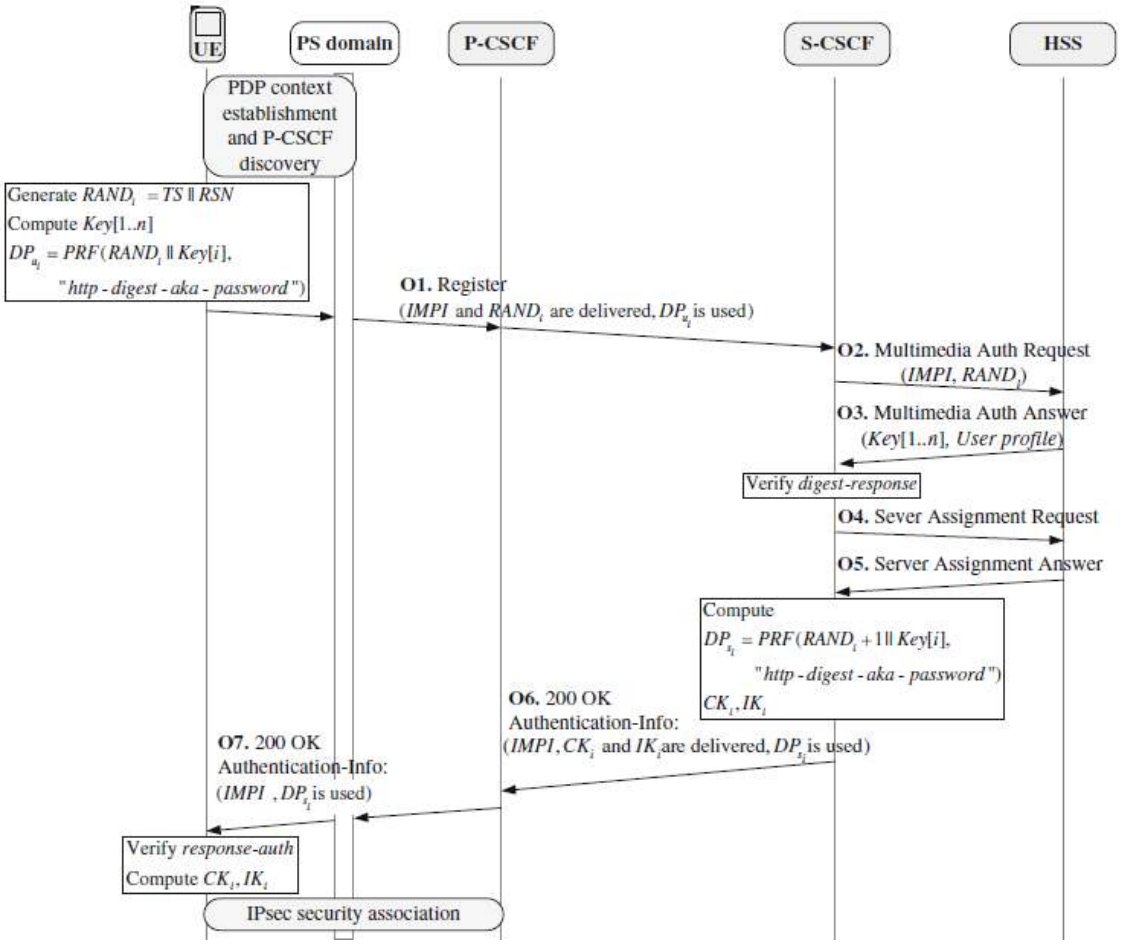

Figure 2-7 One pass IMS AKA [Huang 2009]

As shown in Fig. 2-7, the proposed one pass IMS AKA authentication procedures are described as following:

- Step 1. In the ith one-pass IMS AKA procedure, the UE generates a timestamp TS, random number RANDi, IMS key, digest password DPui, and digest-response. Then it sends SIP request to the P-CSCF with IMPI, RANDi, and digest-response.
- Step 2. Upon receipt of the Step 1 message, the S-CSCF checks the validity of the received RANDi number. Then it sends a request to the HSS to request the IMS key.
- Step 3. HSS uses RANDi to derive the IMS key and sends the key back to the S-CSCF.

- Step 4. Upon receipt of the Step 3 message, the S-CSCF keeps the IMS key for the UE and derives digest password DP'ui. Then the S-CSCF verifies the digest-response retrieved from Step 1 using DP'ui. If the result is positive, it means that UE is a legal user. Finally the S-CSCF sends a SAR message over Cx reference to HSS.
- Step 5. HSS stores the name of S-CSCF and sends a SAA message over Cx to S-CSCF.
- Step 6, and 7. The S-CSCF derives DPsi, CKi, IKi, and response-auth; and sends a SIP 200 OK message to the P-CSCF with IMPI, CKi, IKi and the response-auth. The P-CSCF stores CKi and IKi , and sends the SIP OK message with IMPI and the response-auth to the UE.
- Step 8. The UE checks the validity of the received response-auth firstly, if the result is positive, the S-CSCF is legal. Then the UE derives CKi and IKi which are used for IPSec security association between UE and P-CSCF.

For this proposed authentication mechanism, the terminal is responsible for generating timestamp TS, random number RAND, IMS key, digest password, and digest response, distributing these AV to the server, and checking the response-auth after receiving REGISTER response. The S-CSCF has to check the random number, generate DP'ui, DPsi, CK, and IK, and verify digest response. From the comparison shown in Table 2-2, the proposed mechanism introduces extensive modification and increases the system complexity.

Table 2-2 Huang and Li and 3GPP defined IMS AKA authentication procedure comparison

One pass IMS AKA in [Huang 2009]	3G UMTS two-pass AKA
UE generates a timestamp TS, random number RANDi, IMS key, digest password DPui, and digest-response	-
SIP Register (UE->P-CSCF, P-CSCF ->I-CSCF, I-CSCF->S-CSCF)	SIP Register (UE->P-CSCF, P-CSCF ->I-CSCF, I-CSCF->S-CSCF)
Diameter Multimedia Auth Request (S-CSCF->HSS)	Diameter Multimedia Auth Request (S-CSCF->HSS)

Calculation IMS key by HSS	Calculation AVs by HSS
Diameter Multimedia Auth Answer (HSS->S-CSCF)	Diameter Multimedia Auth Answer (HSS->S-CSCF)
Storing IMS key by S-CSCF	Storing AVs by S-CSCF
Deriving digest password DP'ui, and checking digest-response	-
Diameter Server Assignment Request / Answer (S-CSCF->HSS, HSS->S-CSCF)	-
S-CSCF derives DPsi, CKi, IKi, and response-auth	-
SIP response 401 UNAUTHORIZED (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)	SIP response 401 UNAUTHORIZED (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)
Checking response-auth, and deriving CKi and IKi	Verification of AUTN and calculation of RES number by UE
Creating IPSec SAs	Creating IPSec SAs
-	SIP Register (UE->P-CSCF, P-CSCF ->I-CSCF, I-CSCF->S-CSCF)
-	Compare RES and XRES by S-CSCF
-	Diameter Server Assignment Request / Answer (S-CSCF->HSS, HSS->S-CSCF)
-	SIP REGISTER response 200OK (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)

2.6.1 Ntantogian

Ntantogian [Ntantogian 2010] proposed a generic mechanism for efficient authentication in Beyond 3rd Generation (B3G) networks that attempted to reduce the execution of the 3GPP defined multi-pass authentication steps by using a security binding mechanism. The proposed mechanism authenticates a user in the second and third step of multi-pass authentication by using the user's authentication credentials from

the initial step. As shown in Fig. 2-8, the proposed IMS service layer authentication used an (IMSI, IMPI) pair to authenticate the user without security protection between the Packet Data Gateway (PDG) and the P-CSCF which suffers the following vulnerabilities.

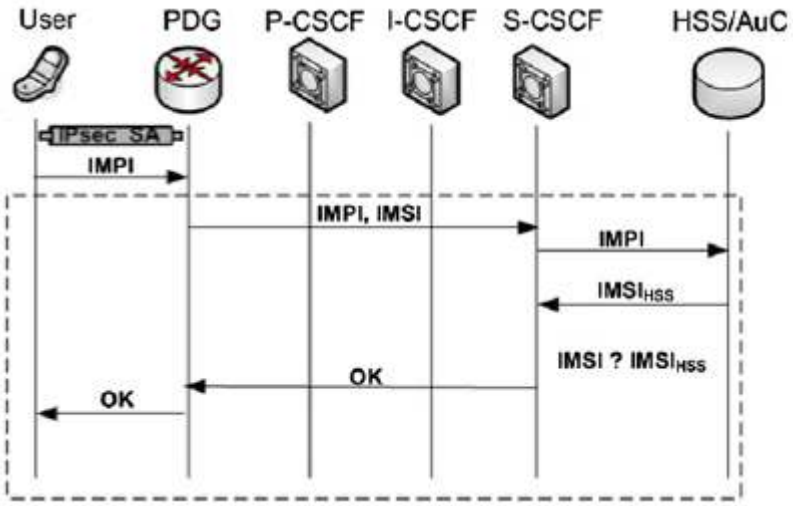

Figure 2-8 Proposed IMS AKA for 3G-WLAN in [Ntantogian 2010]

- Lack of mutual authentication: In the proposed IMS layer authentication, the terminal doesn't support the authentication of the server. Therefore, an adversary may pretend to be the P-CSCF/I-CSCF/S-CSCF/HSS servers to get user related confidential information.
- Lack of confidential and integrity protection: The proposed IMS layer authentication doesn't support the confidential and integrity protection of the messages transmitted between the PDG and the P-CSCF which makes it very vulnerable to the eavesdropping or the middle-man attack.
- Vulnerable of the replay attack: it is assumed there is an adversary sits between the PDG and the P-CSCF and captures packets. Firstly, it could initiate a replay attack by sending the captured packet SIP Register request with (IMPI, IMSI) pair to the server. Since the S-CSCF only checks whether the IMSI received from the terminal equals to the IMSI received from the HSS, with the captured correct (IMPI, IMSI) pair, the adversary will be successful to pass the authentication and get access to the services. Secondly, the adversary could pretend to be the IMS servers to send the captured SIP 200 OK response to the terminal. Without the right security protection, the terminal will create a trust session with the faked

server and send its confidential information to it. Therefore, the proposed IMS layer authentication is very vulnerable to the replay attack.

- Vulnerable to DOS attack: It is assumed that an adversary sends SIP Register request with an IMPI number which belongs to a valid user to the P-CSCF. The P-CSCF would forward this packet to the I-CSCF, the S-CSCF, and the HSS. Therefore, the adversary could flood the core network and jeopardise the whole system.

3 Objectives

The objective of this research is to develop improved IP multimedia subsystem authentication for the 4G LTE network with increased performance, security and less energy consumption. An OPNET Modeller network simulation would be developed and used in this research to validate the improved one-pass authentication system and to permit analysis of the authentication steps, security and performance.

The aim of this thesis is to present research outcomes including a description of the mobile network evolution, a description and analysis of the 3GPP defined 4G LTE authentication protocol, a description of the proposed IAKA authentication protocol, the performance, security, energy consumption analysis of the proposed IAKA authentication protocol, and an in-depth comparison between the 3GPP defined 4G LTE authentication protocol and the proposed IAKA authentication protocol. Furthermore, the research includes the development of a LTE-IMS system model and the simulation of the IMS layer authentication procedure which would be used to do the performance analysis.

The research focus includes:

- Investigation of the mobile network evolution
- Investigation of LTE and IMS system architecture, interfaces, and procedures
- Modelling 3GPP 4G LTE EPS layer authentication procedures
- Modelling 3GPP 4G LTE IMS layer authentication procedures
- Identifying security requirements of LTE and IMS authentication
- Utilising the OPNET Modeller v16.0 network simulator to develop a network and scenarios
- Developing LTE-IMS system model by using OPNET Modeller v16.0
- Developing two scenarios, one is for the 3GPP authentication protocol, and the other is for the IAKA authentication protocol
- Developing a SIP registration procedure which could support packet forwarding between different CSCF servers
- Developing the Diameter authentication protocol between the S-CSCF and the HSS

- Developing the Diameter service (HSS) which could support the Diameter protocol and IMS AKA protocol
- Developing IMS AKA authentication algorithm f1, f2, f3, f4, f5 to generate MAC, RES, CK, IK, and AK [TS 35.206 2009].
- Developing IPSec to protect messages transmitted between the UE and the P-CSCF confidentially and integrity
- Developing a Key Derivation Function (KDF) to derive keys used in the authentication process
- Collecting authentication delay under different system loads and different configurations
- Performance analysis and comparison
- Developing system model for energy consumption analysis
- Energy consumption comparison
- Security analysis of the mutual authentication, confidential and integrity protection, and how to use the proposed authentication approach to avoid malicious attacks

3.1 Research Limitations

This research provided an improved one pass AKA authentication protocol with analysis of performance, security and energy consumption. The data used in the performance analysis was collected from OPNET Modeller LTE-IMS simulation, and the data used in the energy consumption analysis is from previous research work. A necessary number of assumptions were made to ensure that reasonable results could be generated within the research time-frame. Challenges were faced during the network model and simulation scenario development which can be summarized in the following points:

- OPNET Modeller v16.0 doesn't have an IMS model;
- OPNET Modeller v16.0 doesn't support the SIP registration procedure;
- OPNET Modeller v16.0 doesn't support SIP packet forwarding between different SIP servers;

- OPNET Modeller v16.0 doesn't support Diameter protocol;
- OPNET Modeller v16.0 doesn't have a model which could act as a HSS server;
- OPNET Modeller v16.0 doesn't support IPSec protocol for the SIP messages;
- OPNET Modeller v16.0 doesn't support AKA protocol;
- OPNET Modeller v16.0 doesn't support KDF functions;

OPNET Modeller v16.0 was used to simulate the LTE IMS network and the data collected and analysed is anticipated to be slightly different from results collected using a real system. OPNET Modeller v16.0 is one of the leading simulation software applications available today and for this reason the limitations associated with its use are considered acceptable.

The limitations associated with the tools to be used during the research were overcome, the missing nodes, protocols and other elements were developed during the research program and the research focused on the architectures, nodes and protocol behaviours that reflected the LTE and IMS networks.

3.2 Assumption

A necessary number of assumptions were made to keep the simulation complexity manageable, while still meeting the research goals. This section describes assumptions made in modelling both the LTE and IMS data networks, as well as the interworking of these two technologies.

- The network models provided and used within the OPNET Modeller simulation application were considered to be reasonable and suitable for the research. References to other research using OPNET Modeller and the network models have been provided.
- The system processing delay and system load provided and used were considered to be acceptable and suitable for the research.

- The energy consumption system model provided and used in analysing the terminal energy consumption was considered to be acceptable and suitable for the research.

4 Improved one pass authentication protocol

This chapter includes a description of the improved one pass authentication protocol and the steps taken during the development of the simulation models and scenarios.

4.1 The proposed IAKA authentication protocol

4.1.1 Outline

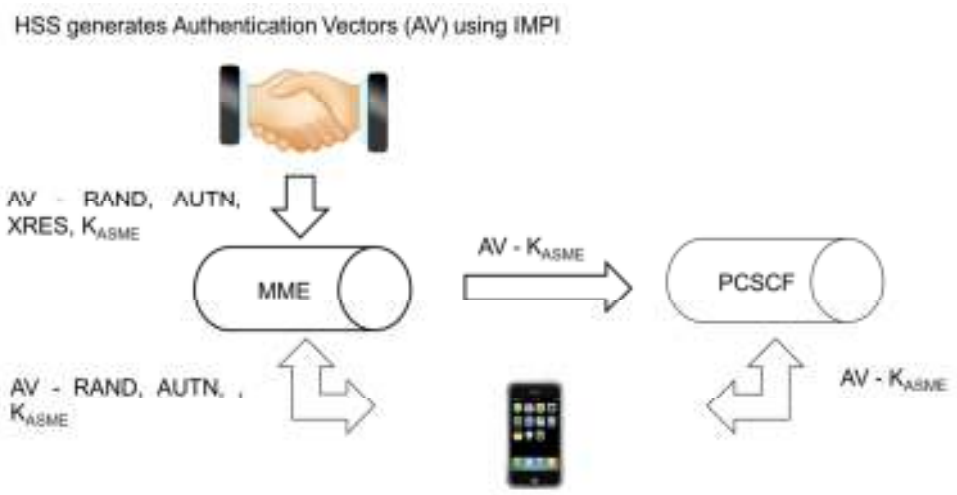

Figure 4-1 The principle of the improved IAKA authentication protocol

Two-pass authentication occurs when it is necessary to provide access to two systems and to keep them relatively independent. 4G LTE is an all-IP based network and the data, voice, and video services are provided via its IP core IMS network. It is considered very redundant to manage two subscriber identities and to run two authentications to access the voice/data services. Only IMSI or IMPI for one subscriber is required in the 4G LTE system to do authentication. Since IMPI is based in IP packet layer and across different access networks, it could be used in both the network and service layers to do registration, authorization, administration and accounting. As shown in Fig. 4-1, this research work proposes IAKA, an improved authentication protocol, which provides a secure binding of the network and service layers authentication by using the IMPI number which avoids the double execution of the AKA authentication protocol. Furthermore, in order to enhance the security and improve the compatibility between LTE and IMS, a 4 layer key hierarchy is proposed. As the results show, the

proposed IAKA could increase the system's performance, save the terminal's energy consumption significantly with enhanced security.

4.1.2 The proposed IMS key hierarchy

The proposed IMS four layer key hierarchy is shown in Fig. 4-2. Compared to the original two layers key hierarchy shown in Fig. 2-4, the intermediate key K_{ASME} is used to derive $K_{PCSCFenc}$ and $K_{PCSCFint}$ to protect the traffic between the UE and the P-CSCF with a particular encryption or integrity algorithm.

$$K_{PCSCFenc} = KDF\ (K_{ASME}\ ,\ s) \quad (1)$$

$$K_{PCSCFint} = KDF\ (K_{ASME}\ ,\ s) \quad (2)$$

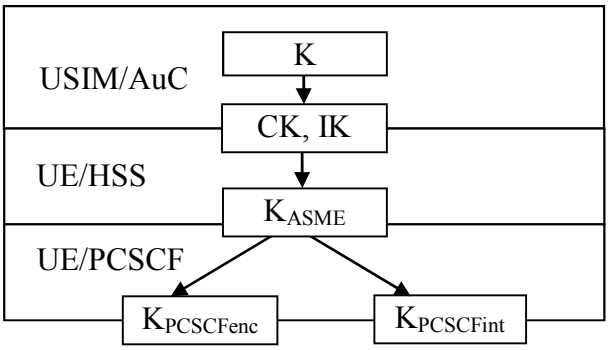

Figure 4-2 The proposed IMS key hierarchy

Where KDF is the key derivation function and s is the input string to the KDF. P0 is a part of the input strings which is used to support the key derivation of $K_{PCSCFenc}$ and $K_{PCSCFint}$ as shown in Table 4-1.

Table 4-1 Algorithm type distinguishers

Algorithm distinguisher	Value
PCSCF_enc_alg	0x06
PCSCF_int_alg	0x07

4.1.3 Improved LTE EPS AKA authentication

When the mobile terminal is powered on, it invokes the Attach procedure to access the LTE network. The EPS AKA is triggered by the Attach procedure to provide mutual authentication and agree on key K_{ASME}.

The 4G LTE EPS AKA authentication protocol utilizes the IMSI number as the identity used for authentication which introduced more authentication overhead. IMPI is a private number of IM systems assigned by home network which is used for registration, authorization, administration, and accounting purposes. Since 4G LTE network is an all-IP network, the IMPI number can be used in the LTE network layer as an identifier to fetch the subscriber's authentication credentials. The procedure is shown in Fig. 4-3.

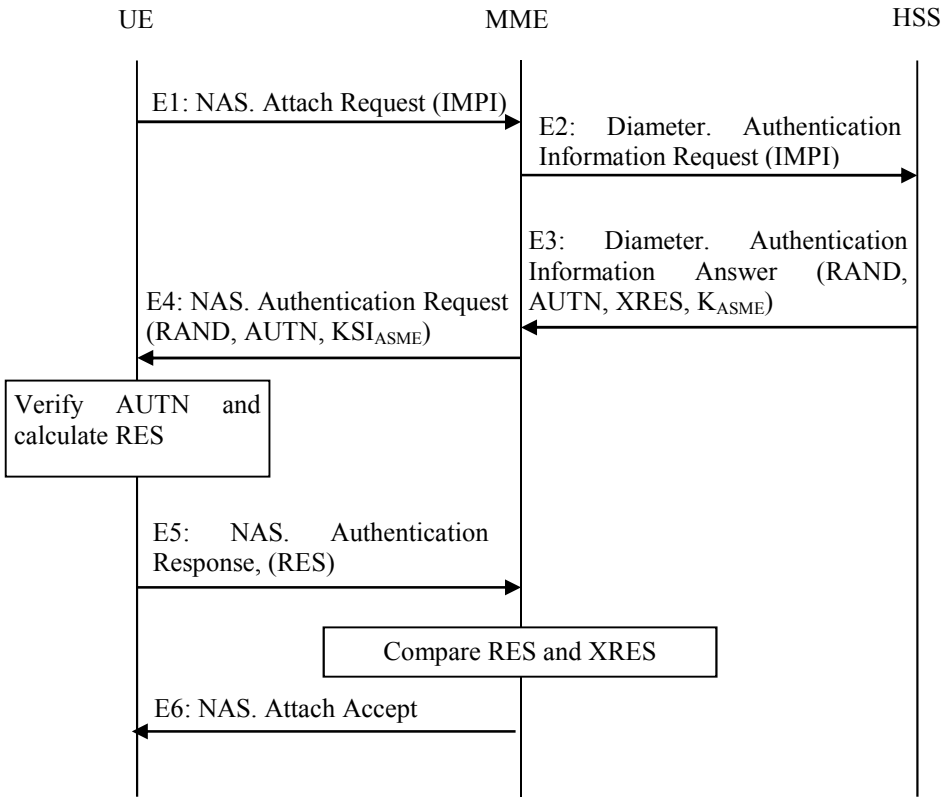

Figure 4-3 Improved LTE EPS AKA authentication procedure

- IE1: In order to access the LTE network, the UE sends an Attach Request to the MME with the IMPI number using the NAS protocol.

- IE2: If there isn't a valid AV in the MME, the MME sends an Authentication Information Request (AIR) to the HSS to fetch the AV by using the Diameter protocol.
- IE3: The HSS uses the IMPI number to fetch the user's profile; generate the AV (RAND, AUTN, XRES, K_{ASME}); and send the AV back to the MME.
- IE4: After receiving the AV from the HSS, the MME sends an Authentication Request (RAND, AUTN, KSI_{ASME}) to the UE to start the authentication procedure. The Key Set Identifier (KSI) is used to identify K_{ASME}.
- IE5: After receiving the Authentication Request, The UE checks the authentication code AUTN first. If this is a valid AUTN number, the terminal considers the network entity to be trusted and continues working with it. Then, the UE calculates the RES number and sends RES back to the MME.
- IE6: The MME checks whether the UE RES matches the HSS XRES, and if they match the MME considers that the authentication and key agreement exchange is successfully completed and sends a message back to the UE to accept the Attach request.

After a successful IAKA EPS authentication, the UE and the MME have completed the authentication steps and have the same K_{ASME} which is used to derive more keys for different security protection purposes.

4.1.4 The improved IMS AKA authentication

After a successful EPS AKA authentication and activation of the EPS bearer, the UE initiates a REGISTER request to login to the IMS network. The procedure is shown in Fig. 4-4 in which step II1 and II4 are used to negotiate the SA parameters and II2 and II3 are used to synchronize the AV. After II4, all of the SIP messages are integrity and confidentiality protected by using the IPSec Encapsulating Security Payload (ESP) protocol [RFC2406].

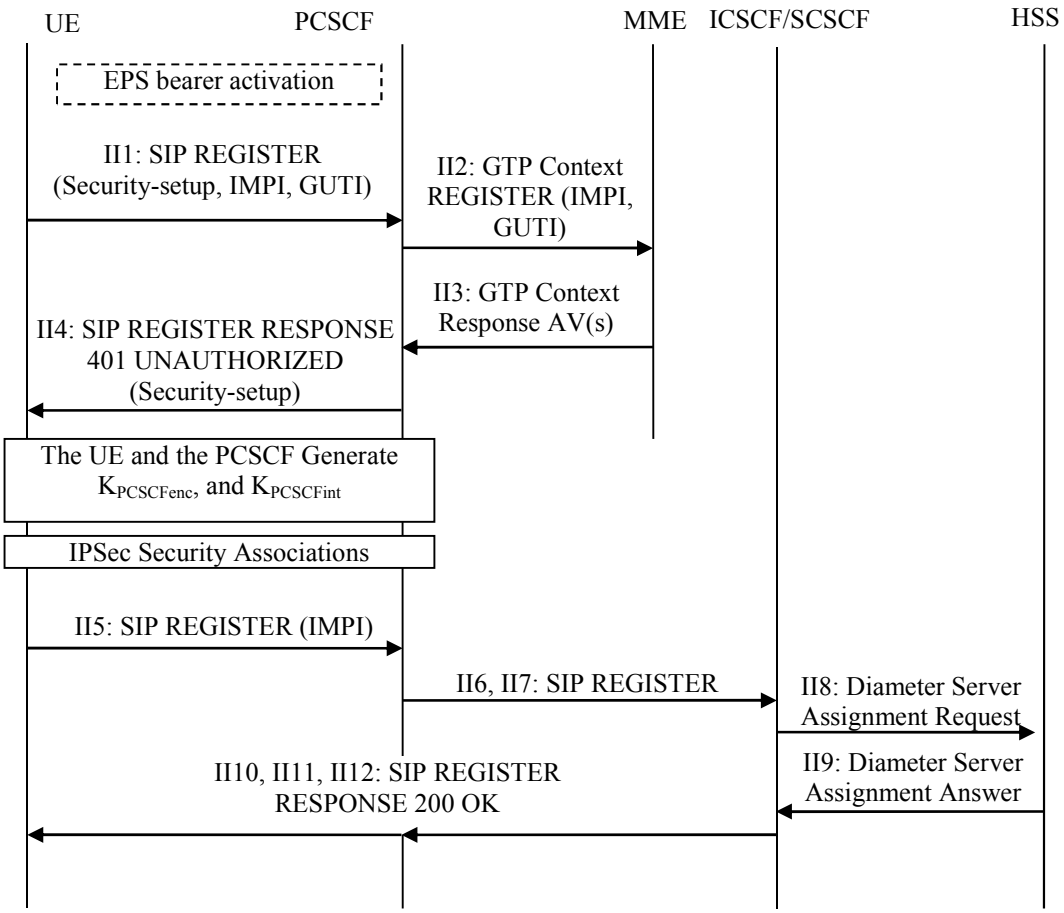

Figure 4-4 the improved IMS AKA authentication procedure

- II1: After activation of the EPS bearer, the UE sends a SIP REGISTER request to the P-CSCF to negotiate the parameters to build SAs with the IMPI number, Global Unique Temporary Identity (GUTI) number and security-setup line. The security-setup line includes the Security Parameters Index (SPI) values, the protected ports selected by the UE and the integrity and encryption algorithm list that the UE supports.
- II2: Upon receipt of II1, the P-CSCF derives the MME address from the GUTI and sends a Context Request to the MME to fetch the AV with the IMPI and GUTI. The protocol between the P-CSCF and the MME is the General packet radio services Tunnelling Protocol (GTP) [TS 29.274] and the security is defined in [TS33.210 2010].
- II3: The MME obtains the user's AV by using the GUTI number and sends the AV back to the P-CSCF.;

- II4: Upon receipt of the AV from the MME, the P-CSCF chooses the SPIs, port numbers, supported integrity and encryption algorithms which are used to construct the security-setup line; and sends the security-setup line back to the UE by using SIP REGISTER response 401 Unauthorised. Meanwhile, the P-CSCF derives the $K_{PCSCFenc}$ and the $K_{PCSCFint}$ for further message protection.;
- II5: Upon receipt II4, the UE derives the $K_{PCSCFenc}$ and the $K_{PCSCFint}$ and builds the SAs by using $K_{PCSCFenc}$, $K_{PCSCFint}$, SPIs, ports and the integrity and encryption algorithms. The UE sends the SIP REGISTER request to the P-CSCF through the IPSec SA;
- II6: The P-CSCF checks the received SIP packet by decryption and calculation of the integrity code. If there is a valid integrity code, the P-CSCF would forward the packet to the I-CSCF;
- II7: The I-CSCF fetches the user and the S-CSCF information from the HSS, locates the next hop S-CSCF address, and sends the packet to the S-CSCF;
- II8: The S-CSCF sends a Server Assignment Request to the HSS with IMPI=impi using the Diameter protocol.;
- II9: The HSS fetches the use's profile and assigns its status by using the IMPI number, and then the HSS sends the Server Assignment Answer back to the S-CSCF;
- II10 and II11: Upon receipt the II9, the S-CSCF responds the I-CSCF by using SIP 200 OK, and the I-CSCF forwards it to the P-CSCF.
- II12: The P-CSCF protects the message by IPSec and sends it to the UE. The UE checks the validity of the packet. If this is a valid packet, the UE considers that it is working with a legal network entity and continues communicating with it.

Table 4-2 IMS LAYER AUTHENTICATION COMPARISON

IAKA	**3GPP**
SIP Register (UE->P-CSCF)	SIP Register (UE->P-CSCF, P-CSCF ->I-CSCF, I-CSCF->S-CSCF)
-	Diameter Multimedia Auth Request (S-CSCF->HSS)
Obtaining AV from MME	-

(P-CSCF->MME, MME->P-CSCF)	
-	Calculation of AVs by HSS
-	Diameter Multimedia Auth Answer (HSS->S-CSCF)
-	Storing of AVs by S-CSCF
SIP response 401 UNAUTHORIZED (P-CSCF->UE)	SIP response 401 UNAUTHORIZED (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)
-	Verification of AUTN and calculation of RES number by UE
Creating IPSec SAs	Creating IPSec SAs
SIP Register (UE->P-CSCF, P-CSCF->I-CSCF, I-CSCF->S-CSCF)	SIP Register (UE->P-CSCF, P-CSCF ->I-CSCF, I-CSCF->S-CSCF)
-	Compare RES and XRES by S-CSCF
Diameter Server Assignment Request / Answer (S-CSCF->HSS, HSS->S-CSCF)	Diameter Server Assignment Request / Answer (S-CSCF->HSS, HSS->S-CSCF)
SIP REGISTER response 200OK (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)	SIP REGISTER response 200OK (S-CSCF->I-CSCF, I-CSCF->P-CSCF, P-CSCF->UE)

4.2 System Model

4.2.1 OPNET Modeller

OPNET is the abbreviation of Optimized Network Engineering which is a provider of the leading software tools for analysing and designing communication networks, devices, protocols, and applications [OPNET 2011]. OPNET Modeller can be used to simulate and design networks, develop protocols and communication technologies, and analyse and compare end to end behaviour. Furthermore, OPNET Modeller supports a development environment to enable modelling of network technologies including UMTS, LTE and SIP.

The key reasons for choosing OPNET Modeller for this research include:

- The fastest discrete event simulation engine among leading industry solutions.
- Supporting LTE, UMTS, SIP and the other related models with source code.

- OPNET Modeller supports flexible customization by using C/C++ programming language which allows modelling of all communication protocols, algorithms and transmission technologies.
- OPNET Modeller supports hierarchical models (Project, Node, and Process) which parallels to the structure of actual communication networks and makes it easy to build simulation network.
- OPNET Modeller carries out simulations using industry specifications and network scenarios that closely match real networks.
- OPNET Modeller supports graphical editors that permit the models to be entered and interacted with.

As shown in Fig. 4-5, OPNET hierarchical Modelling domains are divided into 4 layers: Network Domain, Node Domain, Process Domain, and External System Domain.

- Network Domain: Network Domain is used to define the network topology which includes the connection entities (nodes), the links that connect these nodes, and the geography information.
- Node Domain: The Node Domain provides the common entities which are used to construct the network such as a SIP Server, a router and so on. A node is created using building blocks called Modules which can be a transmitter, a receiver, a processor, a queue, or an external system. Packet Stream may be used to connect the modules to transmit information. The building block Module is edited in the process domain.
- Process Domain: The process domain uses finite state machines to define the behaviour of processes and it is programmed by using C/C++.
- External System Domain: OPNET Modeller provides an interface to integrate the Modules developed by the other tools into an OPNET network.

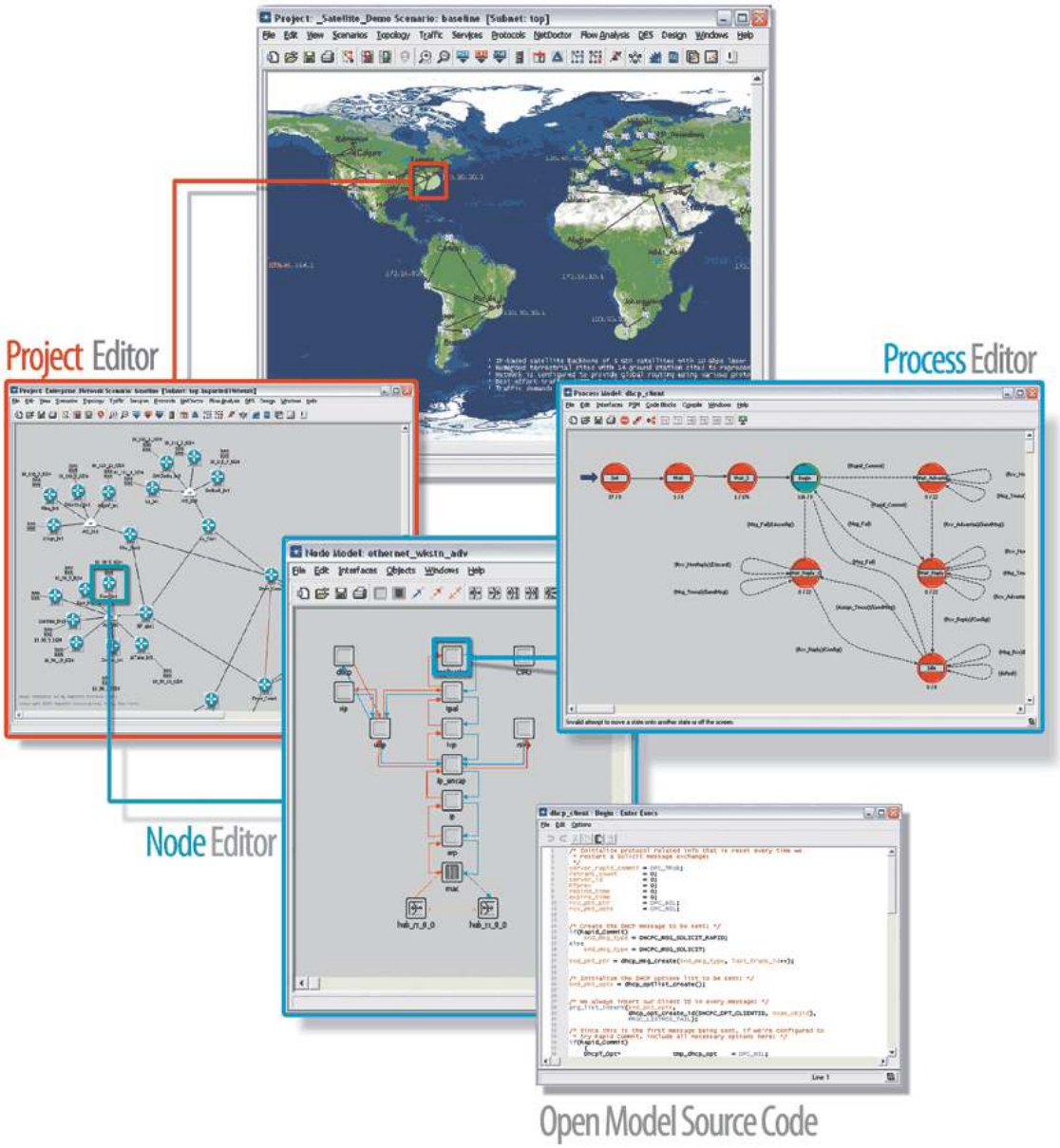

Figure 4-5: OPNET hierarchical GUI editors

4.2.2 Model Methodology

The following methodology was used to design and develop the LTE-IMS models used to carry out the research.

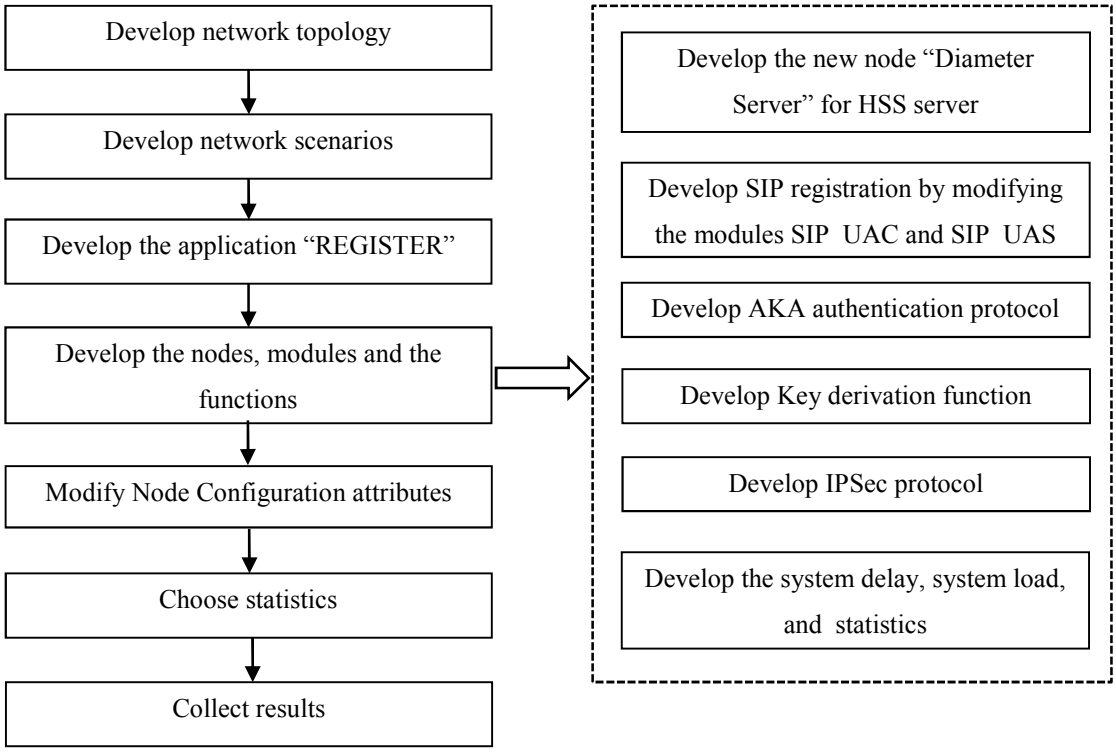

Figure 4-6 OPNET Modeller development methodology

4.2.3 Network Topology

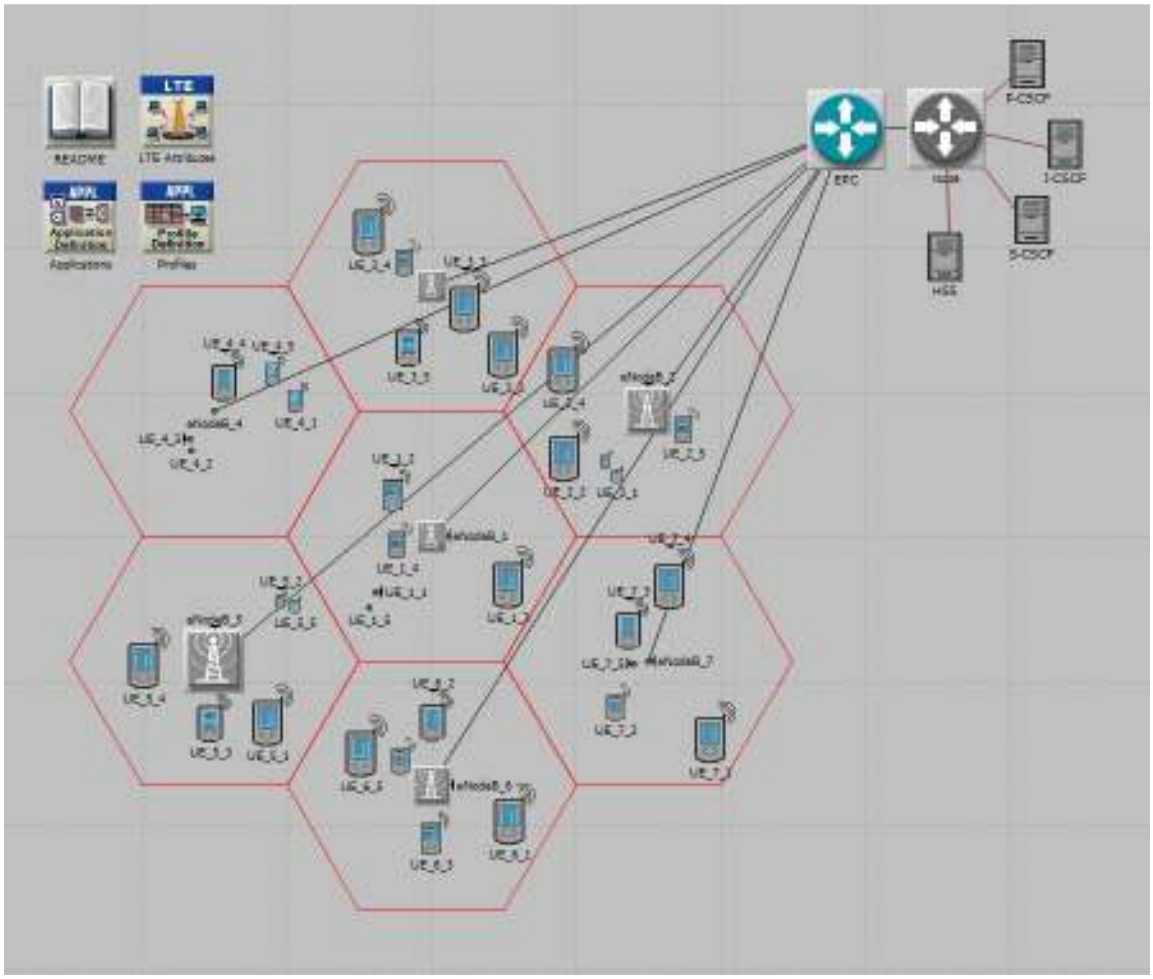

Figure 4-7 Simulation network topology

In order to do a comprehensive analysis of the proposed authentication protocol, a LTE-IMS model was developed using OPNET Modeller. The LTE-IMS model was used to simulate different authentication approaches; and the network topology is shown in Fig. 4-7.

The LTE system is made up of 7 cells; each cell has 5 mobile terminals and 1 eNB; the 5 terminals are connected to the eNB in the same cell and all of the eNB are connected to the EPC. Since the EPC is the integration of the MME, SGW and PGW, it takes the responsibility to connect to the IMS system.

The IMS system is made up of one P-CSCF server, one I-CSCF server, one S-CSCF server and one HSS server. The three CSCF servers are for the call session control and the HSS server is for the end user authentication.

OPNET Modeller includes the 3GPP specified LTE model and the model provided meets most of the research requirements.

OPNET Modeller doesn't include a standard IMS model based on 3GPP specifications. And the standard SIP model supported by OPNET Modeller can't be used to create the IMS scenarios because of the following limitations:

- There is only one SIP proxy which can be put between two user terminals
- It doesn't support SIP registration
- It doesn't support multi domain and roaming scenarios
- It doesn't consider the intermediaries' process delays
- It doesn't consider all the messages that take part in a dialog among SIP intermediaries in the IMS
- It doesn't allow the control of intermediaries' process delays
- It doesn't support AKA authentication protocol

The OPNET Modeller contributed SIP-IMS model [Vazquez 2005] has the following features and was therefore selected as a suitable model that would be modified and upgraded during the research program to permit the IAKA to be modelled adequately:

- It supports IMS servers: P-CSCF, S-CSCF, and I-CSCF;
- It supports multi domain and roaming scenarios;
- It supports process delay.

An OPNET Modeller HSS model was not found which supports the Diameter protocol and the corresponding user authentication and user authorization. Therefore, a HSS model was developed using the standard ethernet_server model as the starting point.

The node models used in the OPNET Modeller simulation scenarios are summarized in Table 4-3.

Table 4-3 The Node Model and Link Model used in the simulation

Name	Model
UE_X_X	lte_wkstn_adv
eNodeB_X	lte_enodeb_atm4_ethernet4_slip4_adv
EPC	lte_epc_atm8_ethernet8_slip8_adv
route	ethernet4_slip8_gtwy
P-CSCF	Contributed sip_proxy_server
I-CSCF	
S-CSCF	
HSS	Developed Diameter protocol and authentication protocol based on the model ethernet_server
LTE Attributes	lte_attr_definer_adv
Applications	Application Config
Profiles	Profile Config

4.2.4 Network Scenarios

Two scenarios were developed using the LTE-IMS model; one is for the legacy 3GPP 4G LTE two-pass authentication protocol, and another is the proposed one-pass authentication protocol. The two scenarios were used to compare the IMS service layer performance using the two authentication approaches. The LTE EPS layer authentication wasn't considered since they they are similar in the two approaches.

In order to identify the different scenarios, a node attribute "Authentication Mode" was defined to show which scenario was active. If the attribute "Authentication Mode" was set to "3GPP", the simulator would run according to the diagram shown in Fig. 2-6; and if the attribute "Authentication Mode" was set to "IAKA", the diagram shown in Fig. 4-4 would be run. Furthermore, a global variable "auth_mode" has to be defined to read the value of the node attribute "Authentication Mode". Table 4-4 shows the definition of the node attribute "Authentication Mode" and the relationship with the global variable "auth_mode".

Table 4-4: Node Attribute “Authentication Mode” definition

Attribute Name	Variable Name	Attribution Value	Description
Authentication Mode	auth_mode	3GPP	Run 3GPP scenario
		IAKA	Run IAKA scenario

The self-defined attribute “Authentication Mode” is under Node “Application Config” and the configuration is shown in Fig. 4-8.

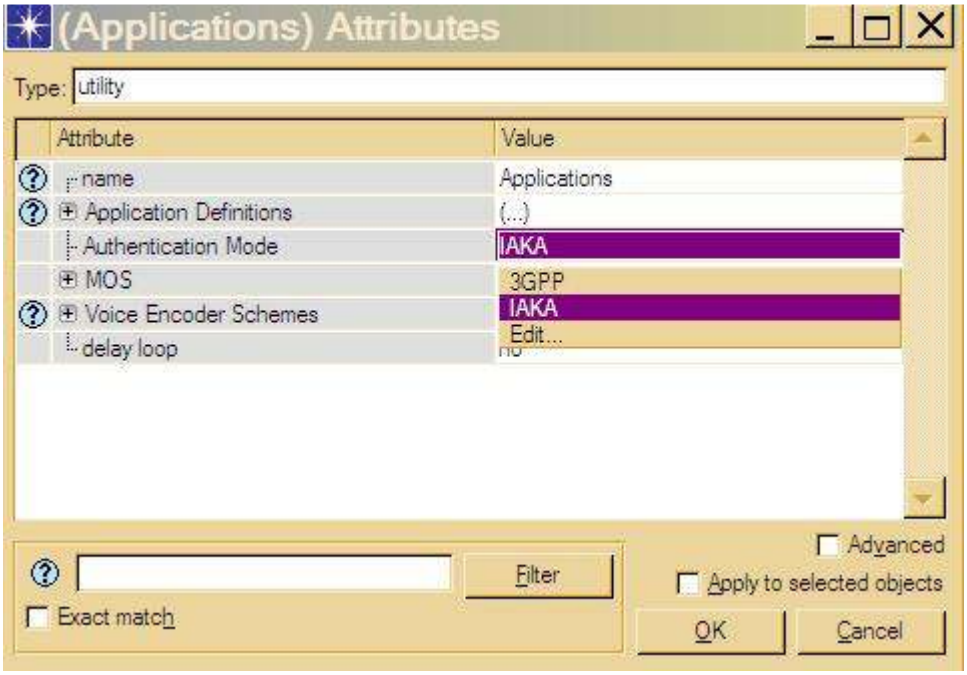

Figure 4-8 Node attribute “Authentication Mode” configuration

In order to define the attribute “Authentication Mode”, we have to open the process model of “Application Config” first, choose the “Model Attributes” and add “Authentication Mode” into this list. The detailed definition is shown in Fig 4-9.

Figure 4-9: Node attribute "Authentication Mode" definition

Once OPNET Modeller is started, the value of the node interface "Authentication Mode" will be read and kept in the global variable "auth_mode" which is used to indicate which authentication procedure should be invoked. The pseudo-code is:

1) Global variable definition

   char auth_mode[128];

2) Read the value of the node interface "Authentication Mode" and store the value into the global variable

   /* Get the authentication mode */

   memset(auth_mode, '\0', sizeof(auth_mode));

   op_ima_obj_attr_get (own_node_objid, "Authentication Mode", auth_mode);

3) Invoke different authentication procedure based on the value of the auth_mode.

   if(strcmp(auth_mode,"3GPP")==0)

   { /*invoke the 3GPP AKA authentication procedure*/ }

else

{ /*invoke the IAKA authentication procedure */ }

4.2.5 Application

A new application “REGISTER” was developed to invoke the SIP Registration procedure repetitively; and the process diagram is shown in Fig. 4-9.

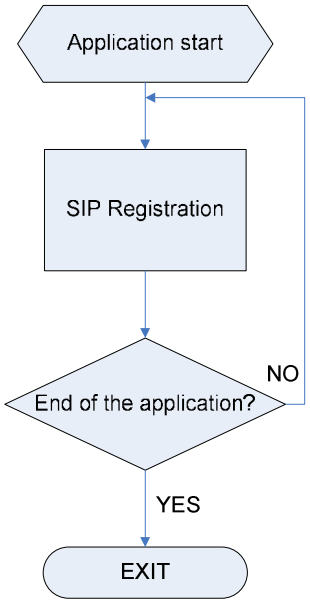

Figure 4-10 Application “REGISTER” process diagram

The application “REGISTER” was developed based on the standard voice application. A new value called “Register only” was added to the voice attributes “Traffic Modelling” which is shown in Table 4-5.

Table 4-5: Attribute “Traffic Modelling” definition

Attribute Name	Data Type	Variable Name	Value	Description
Traffic Modelling	integer	Simulation _mode	0	Control & Traffic Plane: a complete voice service with control signals and voice data traffic
			1	Control Plane Only: a voice service only with control signals (no voice data traffic)
			2	Register Only: a service only do SIP registration repeatly

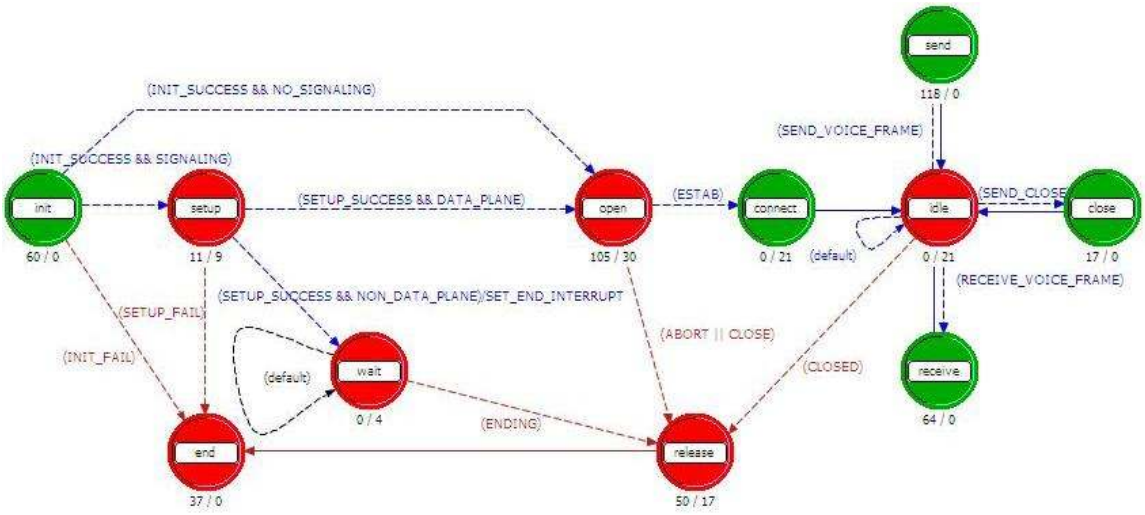

Figure 4-11: Voice application state transition diagram

As shown in Fig. 4-11, the transition states are:

1) Constants in “gna_mgr.h”

/* Indicates the simulation mode for the Voice & Video Application*/

#define GNAC_TRAFFIC_AND_CONTROL 0

#define GNAC_CONTROL_ONLY 1

#define GNAC_REGISTER_ONLY 2

2) States in “gna_voice_calling_mgr.pr.m”

#define DATA_PLANE (simulation_mode == GNAC_TRAFFIC_AND_CONTROL)

#define CONTROL_ONLY (simulation_mode == GNAC_CONTROL_ONLY)

#define REGISTER_ONLY (simulation_mode == GNAC_REGISTER_ONLY)

#define NON_DATA_PLANE (simulation_mode == GNAC_CONTROL_ONLY || simulation_mode == GNAC_REGISTER_ONLY)

In the “setup” state, if the “simulation_mode” is “GNAC_REGISTER_ONLY”, the current process will invoke the function “sip_request_register” to send a SIP

registration request to the process “SIP UAC” to start the authentication procedure. In the “release” state, the voice process also will invoke the function “sip_request_register_finish” to terminate the SIP registration repetition in the process “SIP UAC”. The detailed source code of the function “sip_request_register” and “sip_request_register_finish” is shown in Appendix A.1.

4.2.6 SIP registration

Since the SIP registration procedure is not supported by any of the OPNET Modeller models, it has to be developed based on the contributed SIP models [Vazquez 2005] which are made up of two logical end-points “sip_UAC” and “sip_UAS”. In the SIP client server model, a SIP User Agent Client (UAC) is a logical network model which sends a SIP request, and SIP User Agent Server (UAS) is used to receive a SIP request and send a SIP response. In the SIP registration process developed during this research, the UE takes the role of sip_UAC, and the CSCF servers take the role of sip_UAS.

Packets are used in the simulations to transfer information from one entity to another. The structure of a SIP packet is shown in Fig. 4-12.

Figure 4-12: SIP packet format

The definition of the SIP packet is shown as following:

1) Definition of the packet attribute “type”

typedef enum

{

SIPC_Packet_Type_Request,

```
        SIPC_Packet_Type_ACK,
        SIPC_Packet_Type_Invalid
}SIPT_Packet_Type;
```

2) Definition of the packet attribute “msg”

```
#define  SIPC_CALL_INVITE                100
#define  SIPC_CALL_BYE                   101
#define  SIPC_CALL_CONNECT_SUCCESS       200
#define  SIPC_CALL_CONNECT_FAIL          400
#define  SIPC_CALL_DISCONNECT_SUCCESS    201
#define  SIPC_CALL_DISCONNECT_FAIL       401
#define  SIPC_CALL_SESSION_PROGRESS      103
#define  SIPC_CALL_PRACK                 104
#define  SIPC_CALL_OK                    105
#define  SIPC_CALL_UPDATE                106
#define  SIPC_CALL_RINGING               107
#define  SIPC_CALL_ACK                   108
#define  SIPC_CALL_REGISTER              109
#define  SIPC_CALL_REGISTER_RES          110
#define  SIPC_CALL_REG_UNAUTHORISED      111
#define  SIPC_CALL_REG_OK                112
#define  SIPC_CALL_REG_REPFINISH         113
#define  SIPC_CALL_REGISTER_NEXT         114
```

3) Definition of the packet attribute “call_info”

```
typedef struct
{
   SIPT_Call_Status       call_status;
```

```
   SIPT_Call_Info*        sip_call_info_ptr;
}SIPT_Call_Info_Shell;


typedef struct
{
        Int                     call_id;
        int                     network_delay;
        OmsT_Qm_Tos             tos;
        char*                   call_initiator_addr;
        char*                   invitee_addr;
        double                  call_init_time;
        double                  UAC_call_connect_time;
        double                  UAS_call_connect_time;
        Boolean                 via_icscf;
        char*                   IMPI;
        char*                   RAND;
        char*                   AUTN;
        char*                   XRES;
        char*                    CK;
        char*                   IK;
        char*                   RES;
        char*                   IPSec_encrypted;
        char*                   IPSec_ICV;
} SIPT_Call_Info;
```

4.2.6.1 sip_UAC

The sip_UAC SIP registration diagram is shown in Fig. 4-13 and was developed in the process module "sip_UAC". The detailed state transition diagram of the "sip_UAC" module is shown in Fig. 4-14.

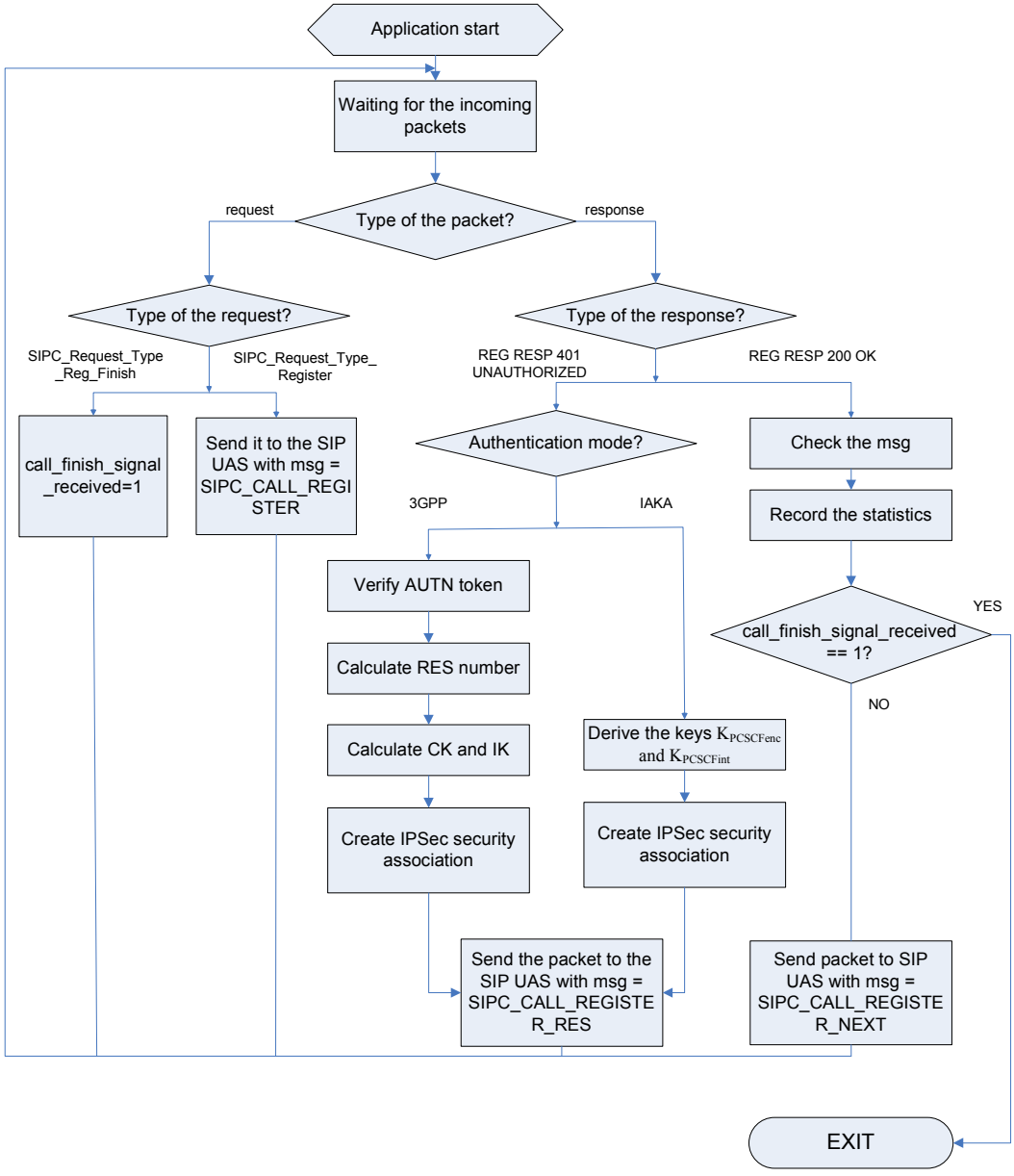

Figure 4-13: SIP registration diagram of SIP UAC

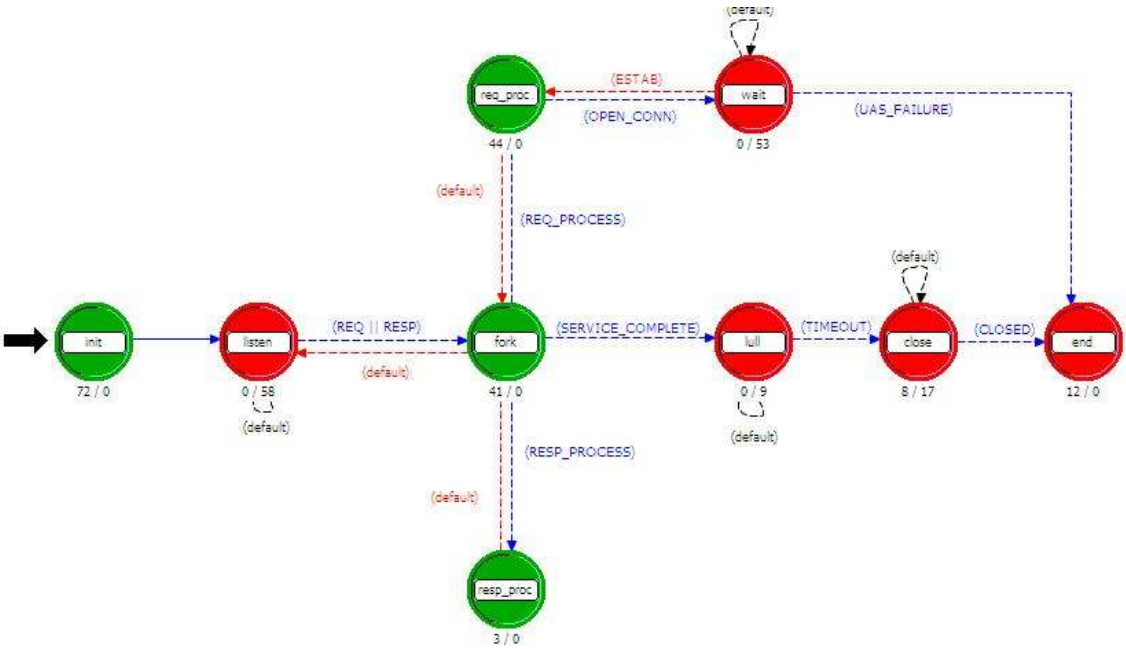

Figure 4-14: sip_UAC state transition diagram

4.2.6.2 sip_UAS

sip_UAS is the processing model in the P-CSCF, I-CSCF, and S-CSCF to receive SIP related packets, process them and send them to the next hop. Since S-CSCF is also connected to the HSS server to do authentication and authorization, the sip_UAS also needs to support the Diameter protocol. Therefore the sip_UAS is designed as shown in Fig. 4-15; and the state transition diagram is shown in Fig. 4-18.

As shown in Fig. 4-15, once the process is started, the sip_UAS process begins to listen to the packets on its sip_UAS port 506 and DIAMETER port 3868. If a SIP packet arrives, the sip_UAS would check the packet type and do the processing corresponding to the type which is described in detail in Fig. 4-17. Then the sip_UAS would decide what is the next hop for the SIP packet; if the next hop is the user terminal or the CSCF servers, the sip_UAS would use a SIP session to send the packet; if the next hop is the HSS, the sip_UAS creates a DIAMETER packet, and sends it to the HSS via the DIAMETER connection. If a Diameter packet arrives, the sip_UAS checks the packet type and translates it to a corresponding SIP packet. The detailed procedure is shown in Fig. 4-17.

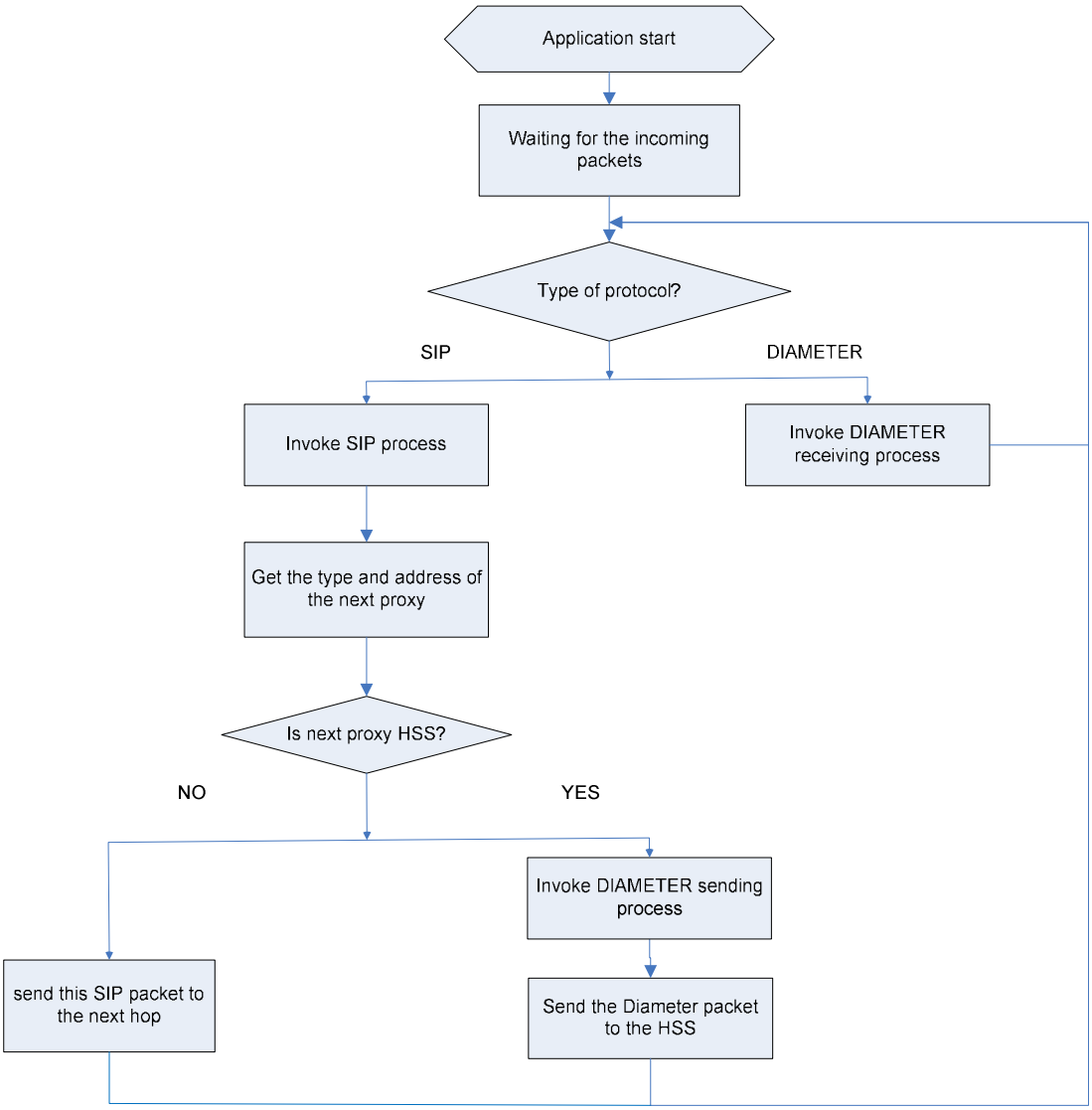

Figure 4-15: sip_UAS SIP registration diagram

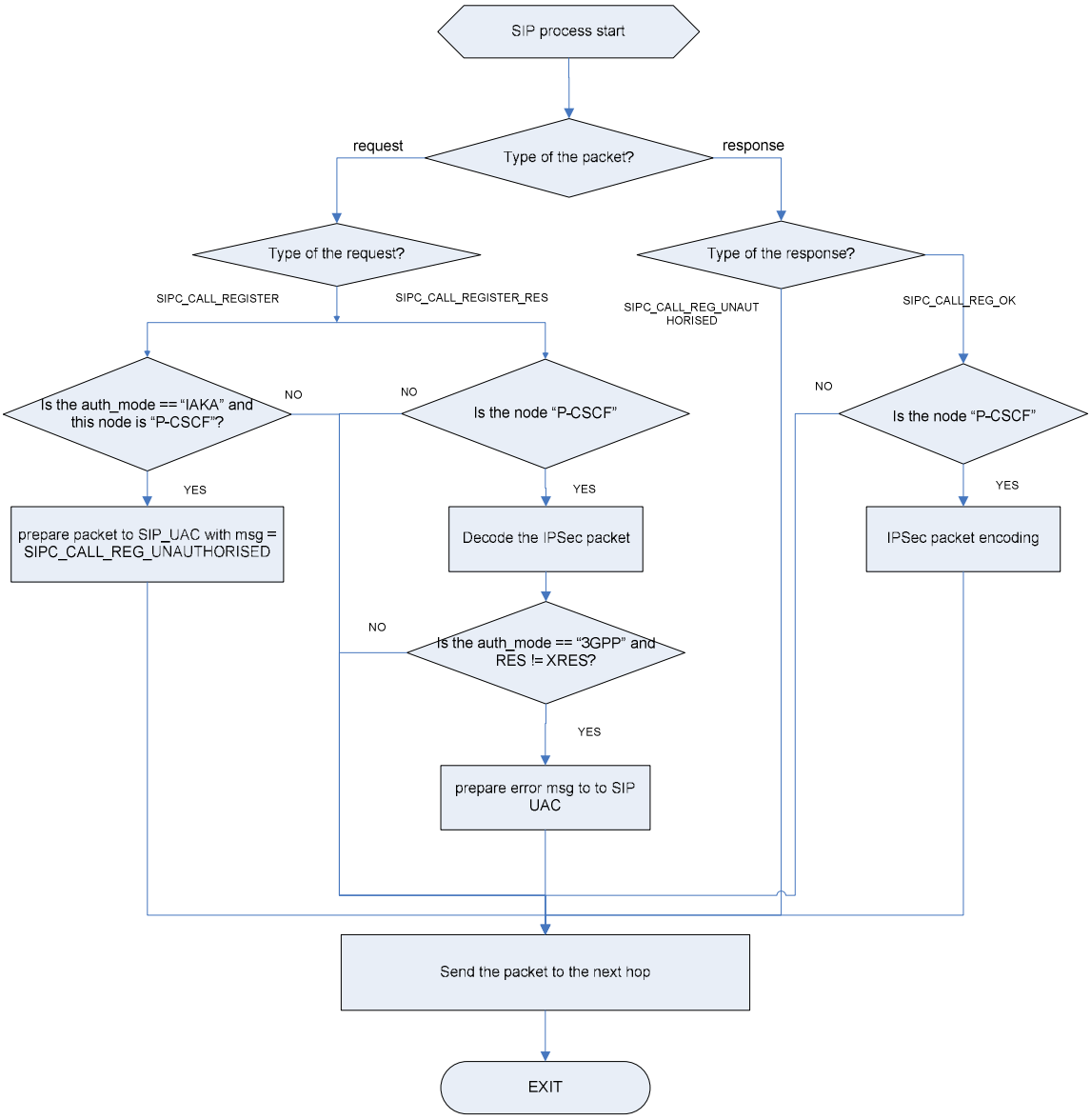

Figure 4-16: SIP packet process

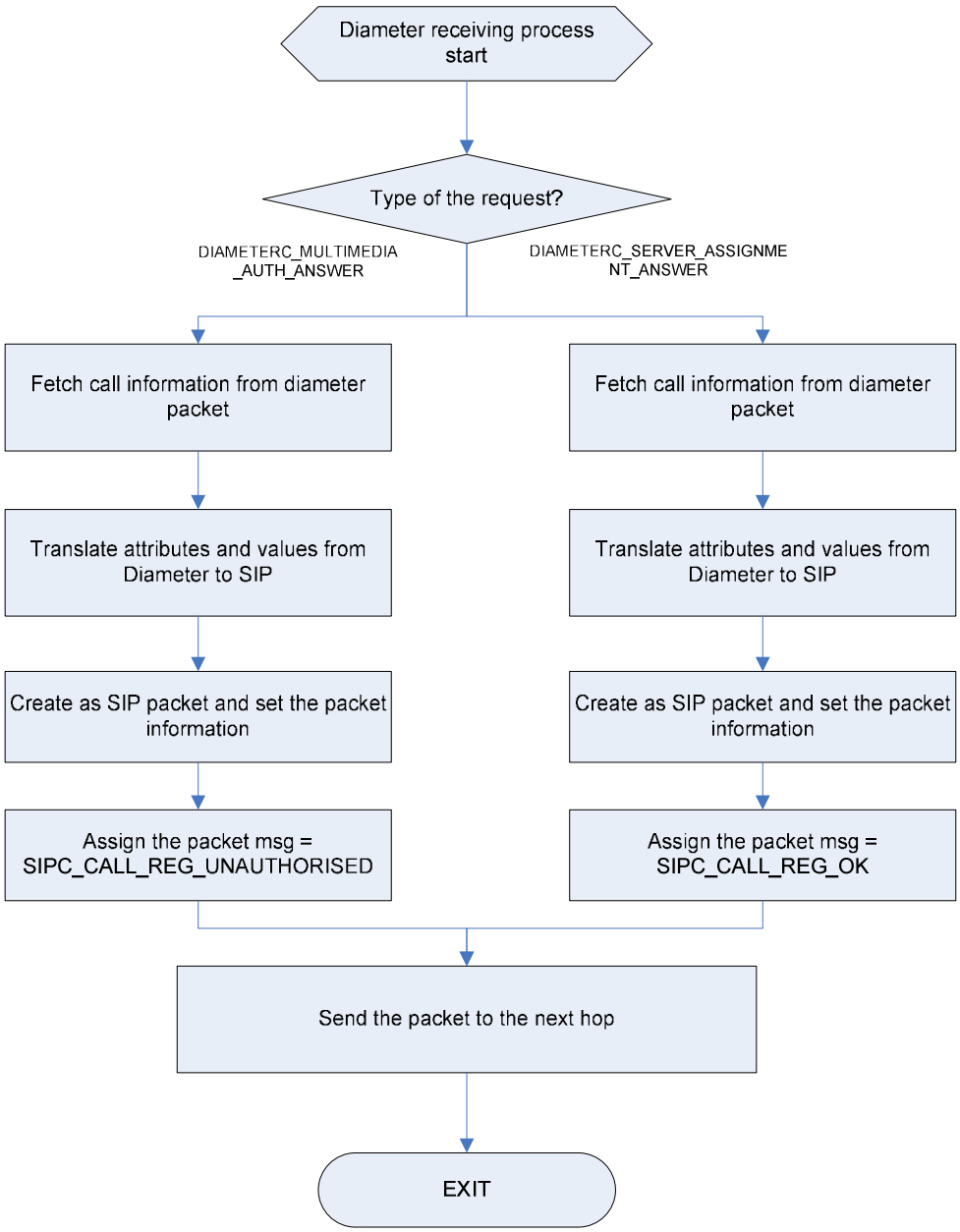

Figure 4-17: Diameter receiving process

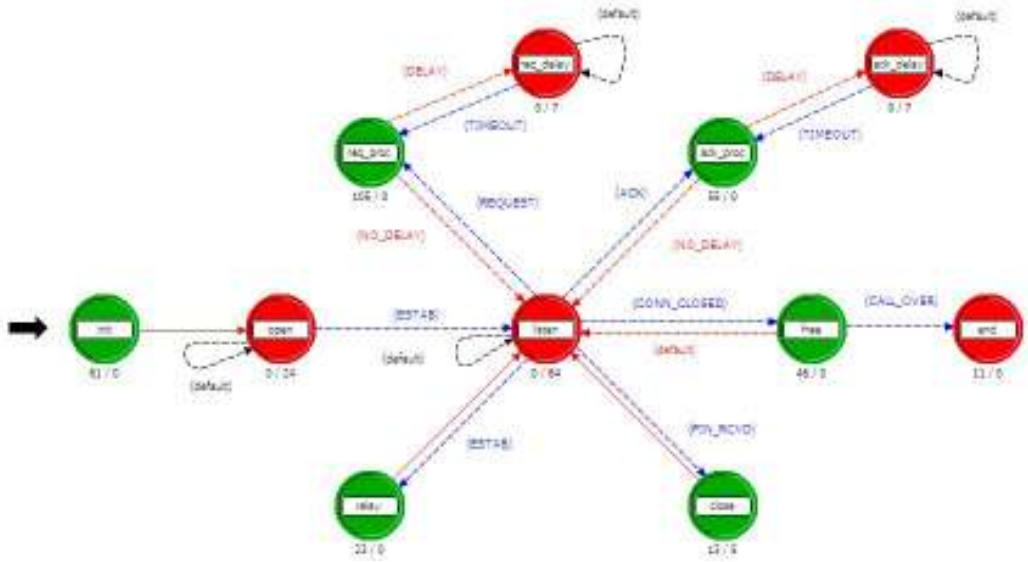

Figure 4-18: sip_UAS state transition diagram

4.2.7 Diameter server

Neither the standard OPNET Modeller models nor the contributed OPNET Modeller models support Diameter and a new Diameter server model was developed during the research program to simulate the HSS server.

The Diameter server was developed starting with the standard ethernet_server_adv model and the Diameter node model is shown in Fig. 4-19. The Diameter server model receives Diameter packets, carries out authentication, and sends a Diameter answer packet to the S-CSCF. The Diameter server functions are developed in the models diameter_UAS_mgr and diameter_UAS.

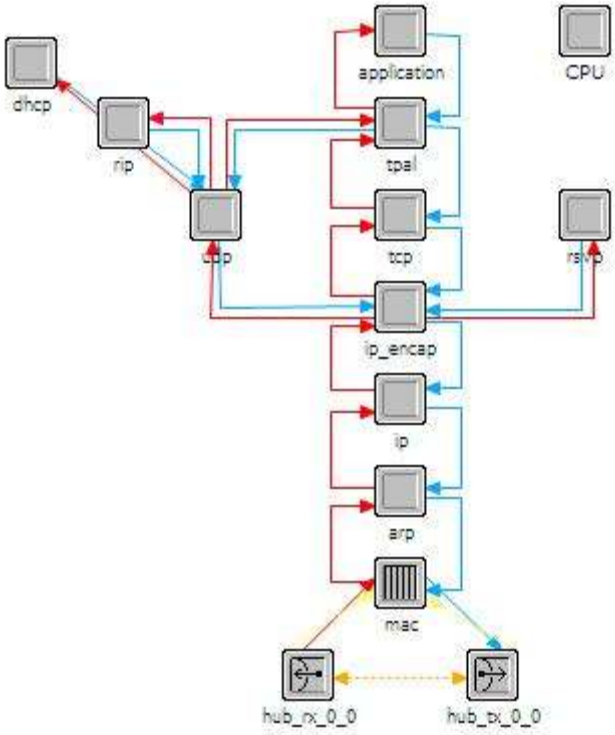

Figure 4-19: ethernet_server_adv node model

4.2.7.1 Diameter packets

Figure 4-20: Diameter packet format

The Diameter packet definition includes:

1) Definition of the packet attribute “type”

```
typedef enum
{
   DIAMETERC_Packet_Type_Request,
   DIAMETERC_Packet_Type_ANSWER,
   DIAMETERC_Packet_Type_Invalid
}DIAMETERT_Packet_Type;
```

2) Definition of the packet attribute “msg”

```
#define  DIAMETERC_MULTIMEDIA_AUTH_REQUEST
                        30301 /* REQUEST OF AUTHENTICATION      */

#define  DIAMETERC_MULTIMEDIA_AUTH_ANSWER
                        30302 /* ANSWER OF AUTHENTICATION      */

#define  DIAMETERC_SERVER_ASSIGNMENT_REQUEST
                    30101 /* REQUEST OF SERVER ASSIGNMENT*/

#define  DIAMETERC_SERVER_ASSIGNMENT_ANSWER
                        30102 /* ANSWER OF SERVER ASSIGNMENT*/
```

3) Definition of the packet attribute “call_info”

```
/*The structure DIAMETERT_Call_Info stores information for identifying and
managing a call and its resources */

typedef struct

{

        int             call_id;

        int             network_delay;

        OmsT_Qm_Tos     tos;

        char*           call_initiator_addr;

        char*           invitee_addr;

        double          call_init_time;

        double          UAC_call_connect_time;

        double          UAS_call_connect_time;

        Boolean         via_icscf;

        char*           IMPI;

        char*           RAND;

        char*           AUTN;

        char*           XRES;

        char*           CK;
```

```
char*                   IK;
char*                   RES;
} DIAMETERT_Call_Info;
```

```
/* DIAMETERT_Call_Info_Shell is the shell around the Call Information which
is used to check for validity of Call Handles. */
typedef struct
  {
  DIAMETERT_Call_Status      call_status;
  DIAMETERT_Call_Info*       diameter_call_info_ptr;
  }DIAMETERT_Call_Info_Shell;
```

4.2.7.2 Diameter_UAS_mgr

Diameter_UAS_mgr is used to manage and activate the Diameter_UAS process, the state transition is shown in Fig. 4-21 and the source code is shown in Appendix A.2.

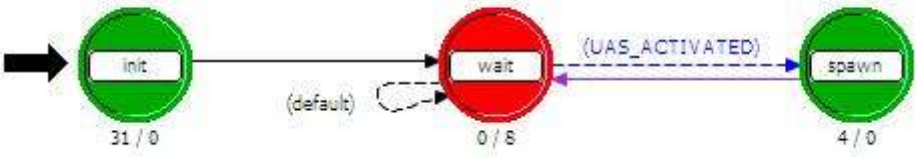

Figure 4-21: State transition diagram of process Diameter_UAS_mgr

4.2.7.3 Diameter_UAS

Diameter_UAS is used to receive and process Diameter request, generate Diameter response, and send it back to the client. The process diagram is shown in Fig. 4-22, the state transition diagram is shown in Fig. 4-23, and the state transition diagram is shown in Appendix A.3.

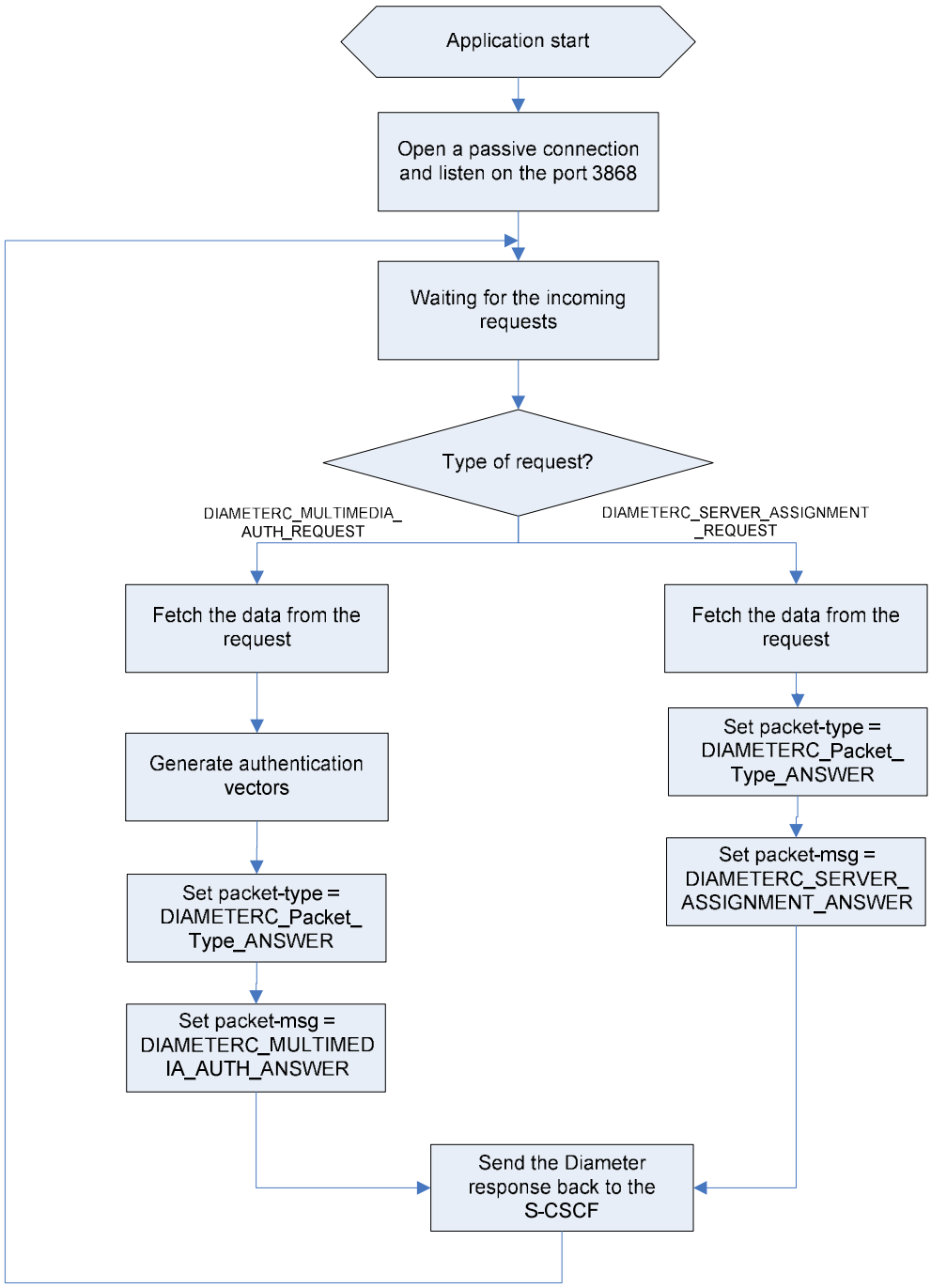

Figure 4-22: Process diagram of Diameter_UAS

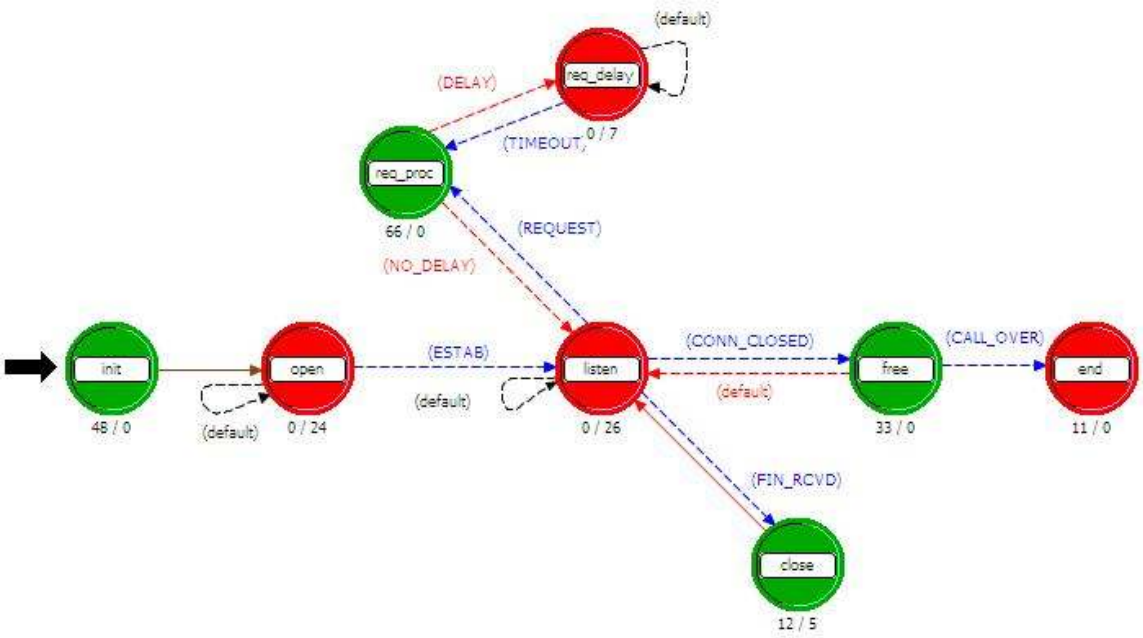

Figure 4-23: Diameter_UAS state transition diagram

4.2.8 AKA authentication protocol

AKA authentication protocol is developed using the algorithm described in 3GPP TS 35.205 and 3GPP TS 35.206.

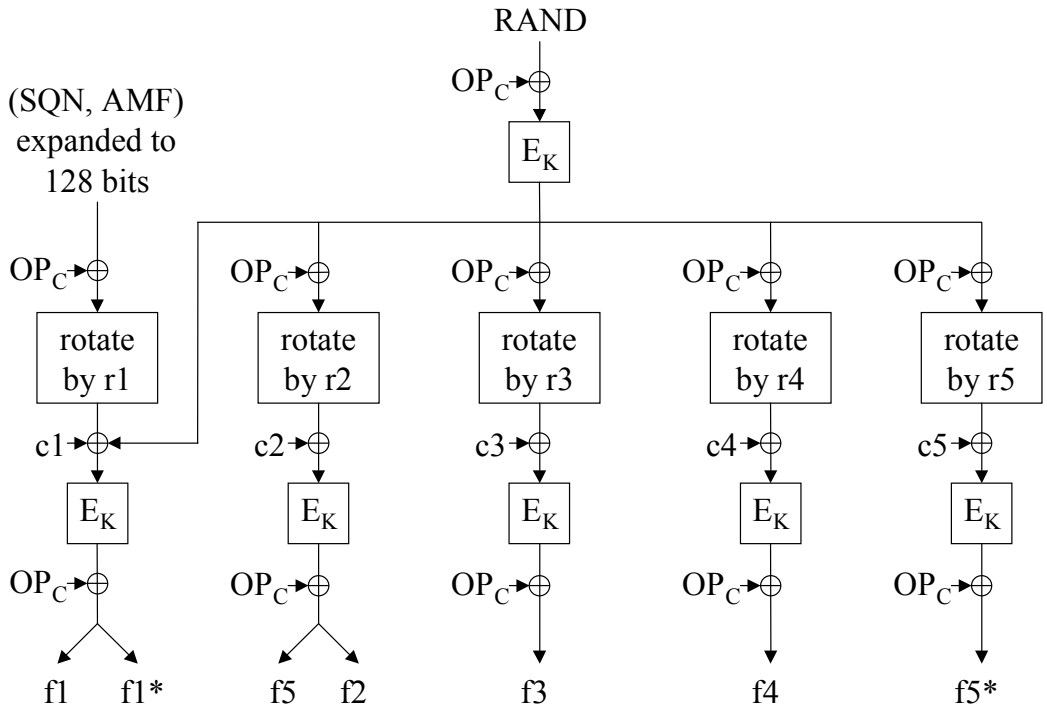

Figure 4-24: f1, f1*, f2, f3, f4, f5, and f5* definition

The items displayed in Fig. 4-24 are described as following:

- RAND is a 128 bits random number;
- K is the key shared by the UE and the HSS;
- OP is a 128 bits operator variant algorithm configuration field;
- OP_C is a 128 bits value derived from OP and K;
- E_k is the kernel encryption algorithm. In this research work, the Advanced Encryption Standard (AES) algorithm is chosen and developed based on [Gladman 2011];
- SQN is a 48 bits sequence number which is fresh in each transaction;
- AMF is a 16 bits authentication management field;
- f1 is the network authentication function which is used to generate Message Authentication Code (MAC);
- f2 is the user authentication function which is used to generate response number (RES);
- f3 is the Cipher Key (CK) derivation function ;
- f4 is the Integrity Key (IK) derivation function;
- f5 is the Anonymity Key (AK) derivation function.

The following are the algorithm framework which is referred to in 3GPP TS 35.206.

- $OP_C = OP \oplus E[OP]_K$.
- TEMP is a 128 bits intermediate value which is computed as
  - $TEMP = E[RAND \oplus OP_C]_K$.
- IN1 is a 128 bit value which is used to construct MAC
  - IN1[0] .. IN1[47] = SQN[0] .. SQN[47]
  - IN1[48] .. IN1[63] = AMF[0] .. AMF[15]
  - IN1[64] .. IN1[111] = SQN[0] .. SQN[47]
  - IN1[112] .. IN1[127] = AMF[0] .. AMF[15]
- c1, c2, c3, c4, and c5 are five 128 bits constants:

- c1[i] = 0 for $0 \leq i \leq 127$
- c2[i] = 0 for $0 \leq i \leq 127$, except that c2[127] = 1
- c3[i] = 0 for $0 \leq i \leq 127$, except that c3[126] = 1
- c4[i] = 0 for $0 \leq i \leq 127$, except that c4[125] = 1
- c5[i] = 0 for $0 \leq i \leq 127$, except that c5[124] = 1

- r1, r2, r3, r4, and r5 are 5 integers:
  - r1 = 64;
  - r2 = 0;
  - r3 = 32;
  - r4 = 64;
  - r5 = 96
- OUT1, OUT2, OUT3, OUT4, and OUT5 are five 128 bits intermediate value:
  - $OUT1 = E[TEMP \oplus rot(IN1 \oplus OP_C, r1) \oplus c1]_K \oplus OP_C$
  - $OUT2 = E[rot(TEMP \oplus OP_C, r2) \oplus c2]_K \oplus OP_C$
  - $OUT3 = E[rot(TEMP \oplus OP_C, r3) \oplus c3]_K \oplus OP_C$
  - $OUT4 = E[rot(TEMP \oplus OP_C, r4) \oplus c4]_K \oplus OP_C$
  - $OUT5 = E[rot(TEMP \oplus OP_C, r5) \oplus c5]_K \oplus OP_C$
- MAC, RES, CK, IK and AK are the outputs of the 5 functions:
  - Output of f1 = MAC-A, where MAC-A[0] .. MAC-A[63] = OUT1[0] .. OUT1[63]
  - Output of f2 = RES, where RES[0] .. RES[63] = OUT2[64] .. OUT2[127]
  - Output of f3 = CK, where CK[0] .. CK[127] = OUT3[0] .. OUT3[127]
  - Output of f4 = IK, where IK[0] .. IK[127] = OUT4[0] .. OUT4[127]
  - Output of f5 = AK, where AK[0] .. AK[47] = OUT2[0] .. OUT2[47]

Based on the algorithm described above, the AKA algorithm was developed using C/C++ in OPNET Modeller, Appendix A.4.

4.2.9 Key derivation of $K_{PCSCFenc}$ and $K_{PCSCFint}$

$K_{PCSCFenc}$ and $K_{PCSCFint}$ are the two keys derived from K_{ASME} to protect the traffic between the UE and the P-CSCF. According to [TS 33.220 2010] and [TS 33.401 2010], the key derivation function (KDF) is defined as following:

Derived key = HMAC-SHA-256(key, S) in which

- Key = K_{ASME}
- S= FC || P0 || L0 || P1 || P0
- FC = 0x15
- P0 = algorithm type distinguisher
- L0 = length of algorithm type distinguisher
- P1 = algorithm identity
- L1 = length of algorithm identity

The key derivation function HMAC-SHA-256 was developed using [Eastlake 2011].

4.2.10 IPSEC

As defined in [TS 33.203 2009], the SIP signalling messages are confidentially and integrity protected by IPSec ESP protocol [RFC 2406]. The encryption algorithm used in the simulator is AES, and the integrity algorithm used is HMAC-SHA-1-96 [Eastlake 2011].

4.2.11 Other system processes

Compared to the authentication procedure defined in Sections 2.5.4 and 4.1.4, the following activities were not implemented and were simulated as a network delay:

- The request processing on the P-CSCF, I-CSCF, S-CSCF and HSS which should be supported in the two approaches and is currently simulated as $DELAY_{processing}$.
- The I-CSCF locating the next hop S-CSCF address by checking the information from the HSS which should be supported in the two approaches and is currently simulated as $DELAY_{ICSCFHSS}$.

- The database operations on the HSS such as searching the user's profile by using IMPI number should be supported in the two approaches and is currently simulated as $DELAY_{HSS}$.
- The P-CSCF obtaining the AVs from the MME should be supported in the IAKA authentication protocol and is currently simulated as $DELAY_{AVretrieval}$.

Two configuration types were utilized for the assumed delays including a fixed network delay configuration and a varying network delay configuration.

The fixed network delay configuration was used to assign a fixed value to the system processing delay to observe the system performance. Referring to [Foster 2002], it was assumed $DELAY_{processing}$ = 0.025s, $DELAY_{HSS}$ = 0.055s, $DELAY_{ICSCFHSS}$ = 0.055s; and $DELAY_{AVretrieval}$ = 0.025s.

For the varying network delay configuration an average network delay was used. It was assumed that $DELAY_{processing} = DELAY_{HSS} = DELAY_{ICSCFHSS} = DELAY_{AVretrival}$ = averageDelay. The average delay started from 1ms and the terminals carried out SIP registration procedures repeatedly and each procedure has a 1ms increment.

A node attribute "delay loop" is defined under the "Application Config" node to identify different configuration modes. If the attribute "delay loop" is set to "no", the OPNET Modeller will use the fixed network delay configuration mode to simulate the system processing. If the attribute "delay loop" is set to "yes", the OPNET Modeller will use the varying network delay configuration mode to simulate the system processing. Furthermore, a global variable "delay_loop" has to be defined to read the value of the node attribute "delay loop". Table 4-6 shows the definition of the node attribute "delay loop" and the relationship with the global variable "delay_loop".

Table 4-6: The definition of Node Attributes "delay loop"

Attribute Name	Variable Name	Attribution Value	Description
delay loop	delay_loop	no	fixed network delay configuration
		yes	varying network delay configuration

The self-defined attribute “delay loop” is under Node “Application Config” and the configuration is shown in Fig. 4-25.

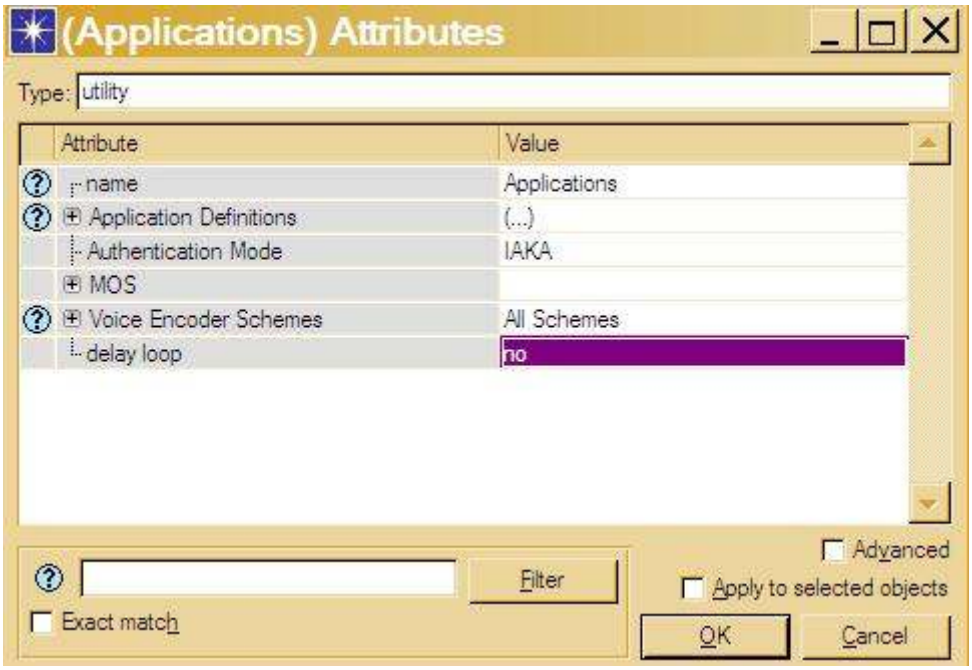

Figure 4-25 Configuration of Node Attribute “delay loop”

Under the fixed delay authentication mode, for sip_proxy_server (P-CSCF, I-CSCF, and S-CSCF), the delays are configured under “SIP Proxy Server Parameters” as shown in Fig. 4-26. For the HSS model, the delays are configured under “Diameter Proxy Server Parameters” as shown in Fig. 4-27.

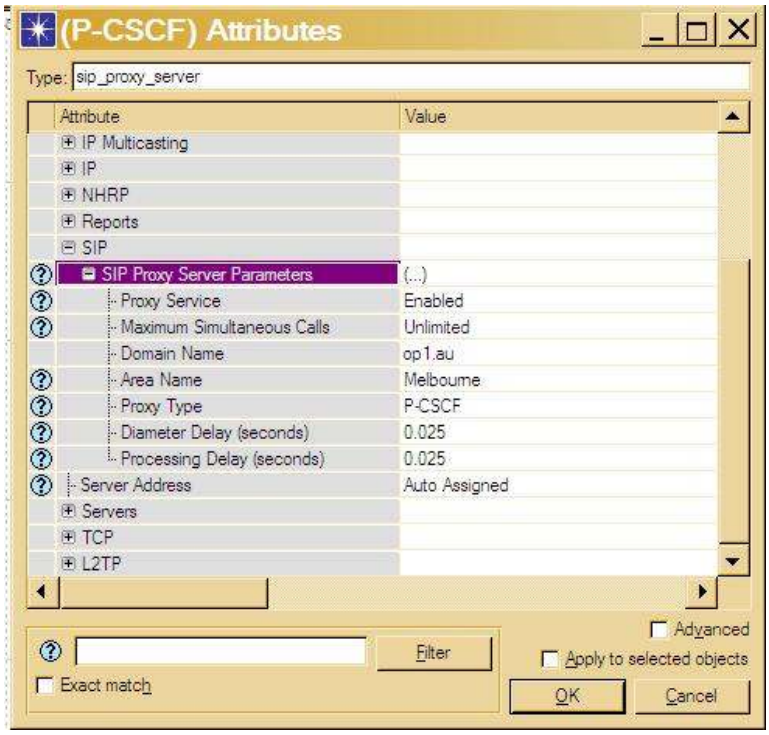

Figure 4-26 Configuration of system processing delay of sip_proxy_server

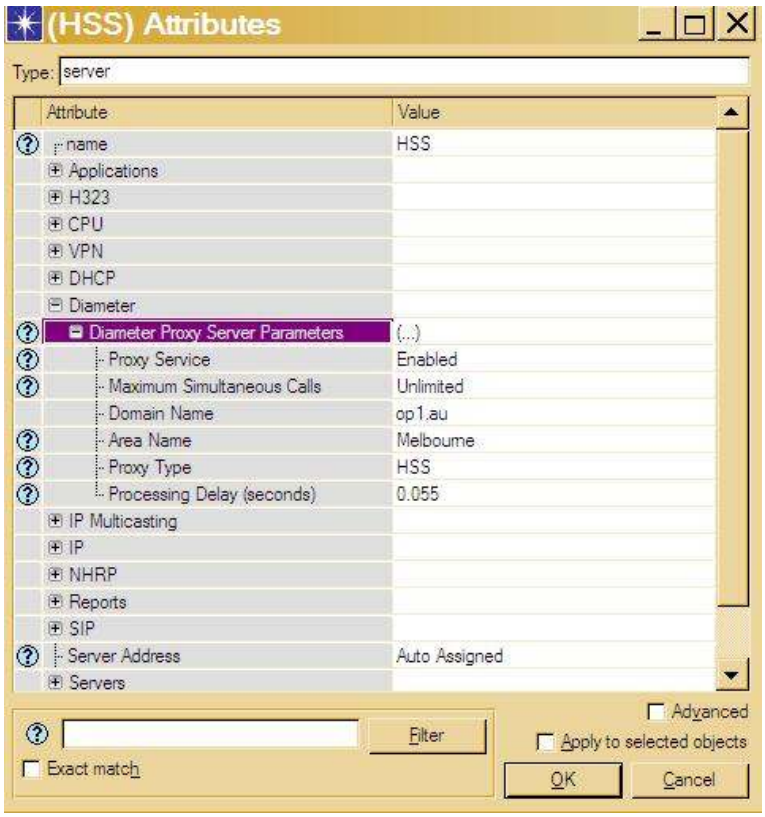

Figure 4-27 Configuration of system processing delay of HSS model

4.2.12 System Load

Simultaneous Rate Per Terminal (SPT) is used to describe the number of SIP registration procedures each terminal initiated simultaneously and the effect on system load. Four different system loads were chosen with SPT = 1, SPT = 20, SPT = 30, and SPT = 40.

The SPT is configured under node “Profile Config” by setting the application “REGISTER” to run concurrently which is shown in Fig. 4-28.

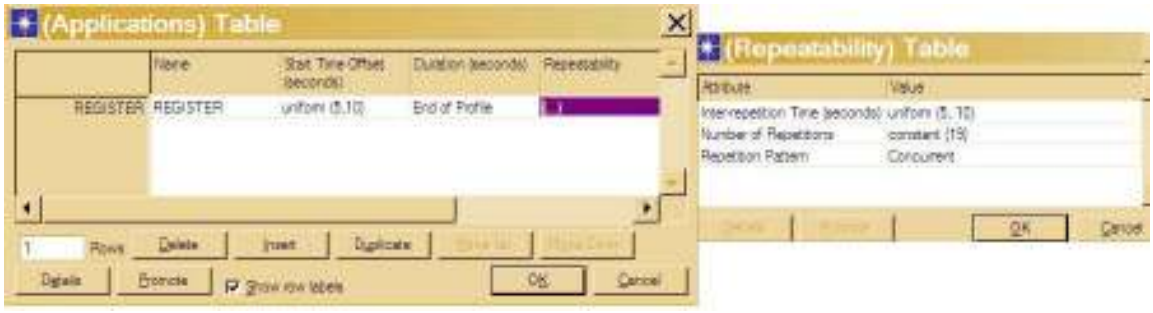

Figure 4-28 Configuration of SPT

4.2.13 Statistics

Authentication delay is used to measure the performance of different authentication approaches which is the time taken between sending the first SIP Register request and receiving the SIP Register response 200 OK, shown in Fig. 4-29.

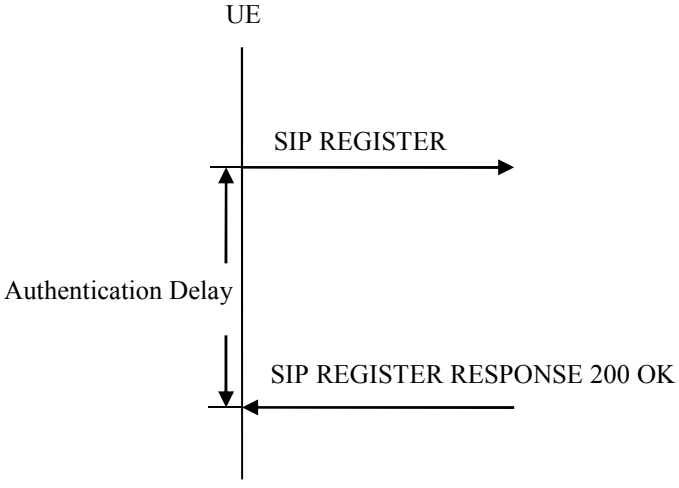

Figure 4-29 Authentication delay

4.2.14 Node Configuration

In the LTE-IMS model, in order to make the results achieved comparative to those expected on a real network, the IMS signals are transmitted in the correct EPS bearer. According to the definition in [TS 23.203 2011], the IMS signalling has highest priority and the QOS Class Identifier (QCI) is 5 which is shown in Table 4-7.

Table 4-7: Standardized QCI characteristics [TS 23.203 2011]

QCI	Resource Type	Priority	Packet Delay Budget (NOTE 1)	Packet Error Loss Rate (NOTE 2)	Example Services
1	GBR	2	100 ms	10^{-2}	Conversational Voice
2		4	150 ms	10^{-3}	Conversational Video (Live Streaming)
3		3	50 ms	10^{-3}	Real Time Gaming
4		5	300 ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
5	Non-GBR	1	100 ms	10^{-6}	IMS Signalling
6		6	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7		7	100 ms	10^{-3}	Voice, Video (Live Streaming) Interactive Gaming
8		8	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file
9		9			sharing, progressive video, etc.)

To include the 3GPP definition in the LTE-IMS model, firstly, the Type of Service (ToS) of the application is defined to Excellent Effort (3); secondly, an EPS bearer called “platinum” is created under the node “LTE Attributes” which is shown in Fig. 4-29; and finally the UE has to be configured to support the EPS bearer “platinum” and the Traffic Flow Template (TFT) is used to map the Excellent Effort traffic in this EPS bearer. The configuration of the UE EPS bearer is shown in Fig. 4-30.

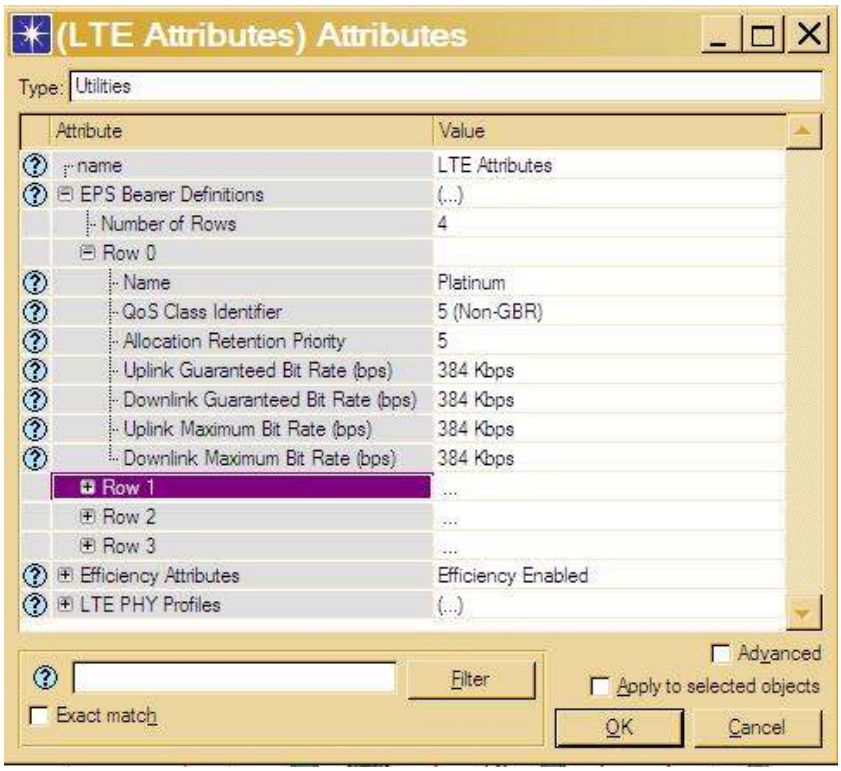

Figure 4-30 EPS bearer “platinum” configuration

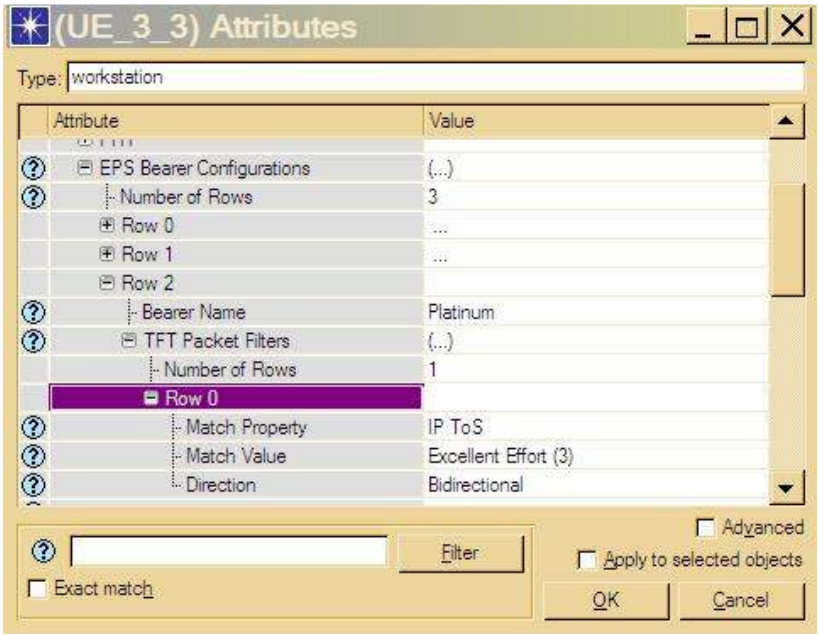

Figure 4-31 UE EPS bearer configuration

4.2.15 Run Simulation

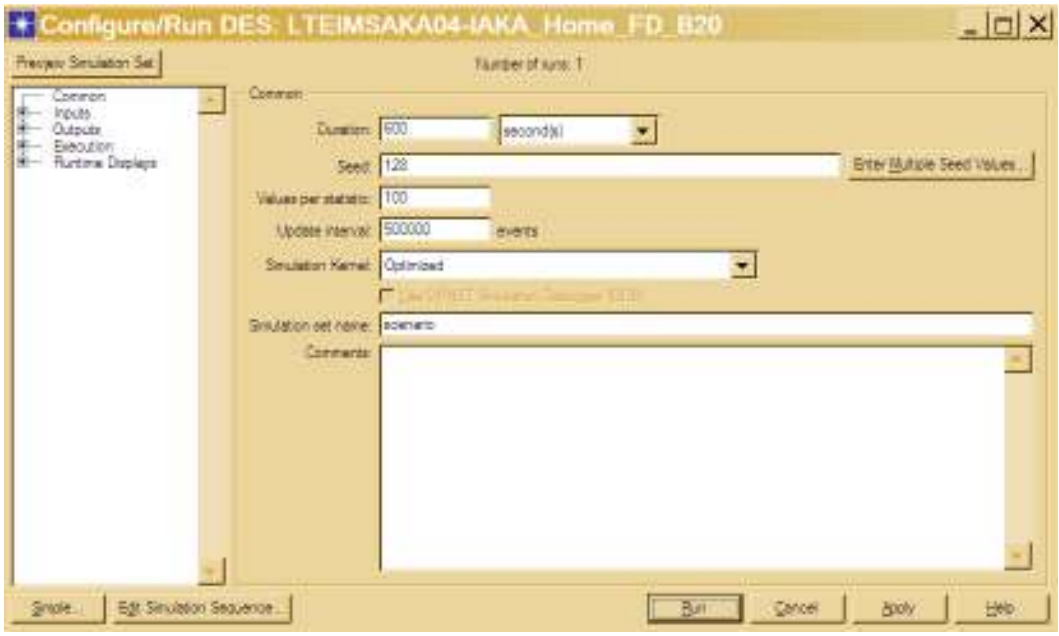

Figure 4-32 Simulation runtime configuration

The simulation is run under the environment which is configured as shown in Fig. 4-31. The simulation duration was ten minutes.

4.2.16 Simulation Results

The simulation results were collected and a detailed analysis and discussion of provided in Section 5.

4.3 Summary

This section has provided the research work carried out and a description of the simulation environment and models developed to support the study network.

5 Analysis

The proposed one-pass authentication protocol was simulated and an analysis was carried out by comparing it with the 4G LTE two-pass authentication protocol and the proposed one-pass authentication protocol.

5.1 Performance Analysis

In order to analyse the performance of the proposed one-pass authentication, a LTE-IMS OPNET Modeller scenario was developed to compare IAKA with the 4G LTE 3GPP defined two-pass authentication protocol. As described in Section 4.2, the performance was compared under different network delay configurations and different system loads.

5.1.1 Fixed Network Delay Configuration

5.1.1.1 System Load SPT 1

When SPT = 1, no background system load is assumed. Under such conditions, the authentication delay is very stable. The average IMS layer authentication delay of the 4G LTE two-pass AKA protocol is 0.4s and the average delay of the proposed IAKA is 0.267s which is shown in Fig. 5-1. Therefore, compared to the legacy 4G LTE two-pass AKA authentication protocol, the proposed IAKA could save 33.34% of the IMS layer authentication delay which is shown in Fig. 5-2.

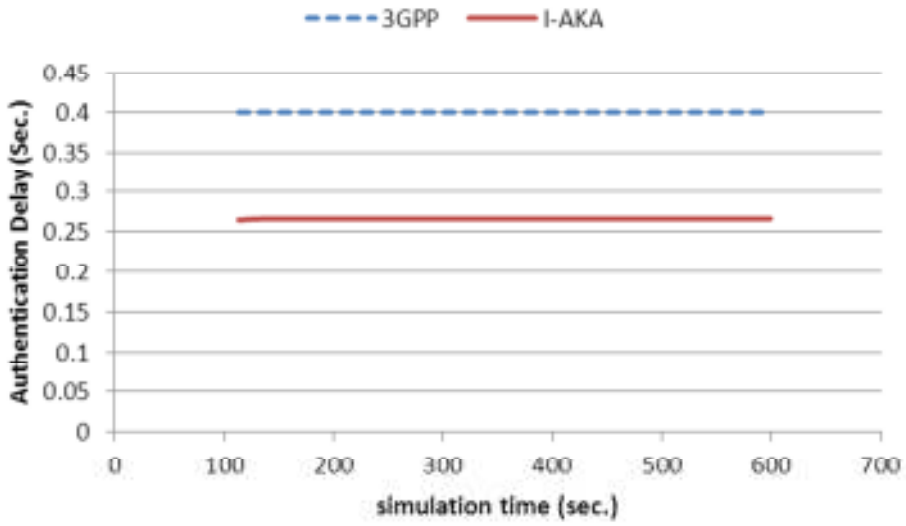

Figure 5-1 Authentication Delay (SPT = 1)

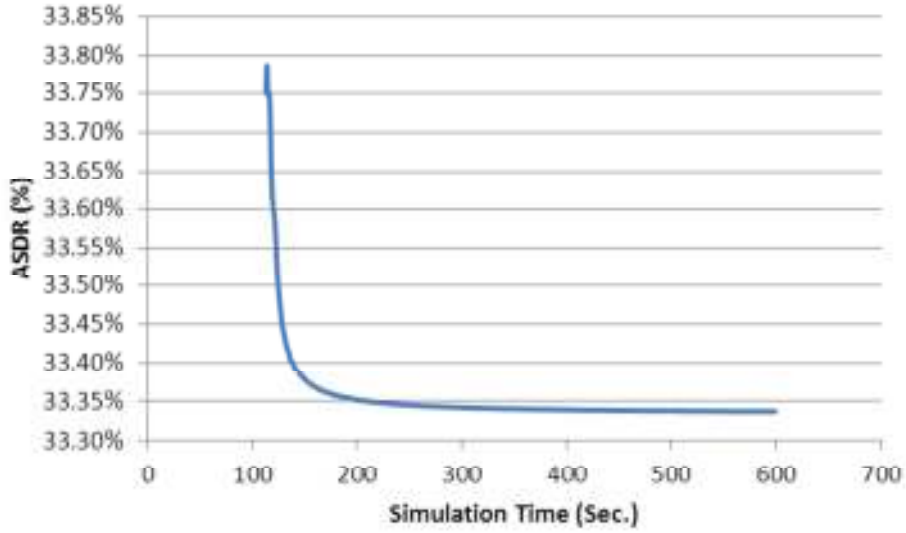

Figure 5-2 Authentication Delay Saving Rate (SPT = 1)

5.1.1.2 System Load SPT 20

When SPT = 20, each terminal generates 20 requests simultaneously. Under this scenario, the authentication delay begins to increase with increasing system load as shown in Fig. 5-3 and Fig. 5-4.

From Fig 5-3 and Fig. 5-4, it can be seen that there is no traffic between 0~100 seconds simulation time because the application starts at simulation time between 100 to 110 seconds (uniform (100,110)).

After the first 100 seconds the application (about the simulation time 100 to 199 seconds) is still in the initialization phase and the results are unreliable.

From simulation time 200 seconds, the authentication delay and the ADSR begin to increase with the increasing system load. From Table 5-1, it is shown that at simulation time 200~299 seconds, the authentication delay of the IAKA approach is 0.30s and the 4G LTE approach is 0.43s while the IAKA mode processes 7682 requests / responses per second and the 4G LTE mode processes 5264 requests/requests per second. At simulation time 500~599 seconds, the authentication delay of the IAKA approach is 0.33s and the 3GPP approach is 0.48s while the IAKA mode processes 8400

requests / responses per second and the 3GPP mode processes 5664 requests / responses per second.

The IAKA authentication mode processes more requests and responses because the terminal initiates the same requests simultaneously for the two authentication steps and repeatedly generates another request once it finishes one authentication procedure. Since the authentication delay of the IAKA mode is shorter than the 4G LTE mode, the UE of the IAKA mode could initiate more authentication request than the 4G LTE mode, and the IAKA mode would have higher a system load than the 3GPP mode.

Table 5-1 Average simulation results (SPT = 20)

Simulation time (second)	Authentication Delay (second)		ASDR (%)	System load of eNB/EPC (requests/ responses per second)	
	4G LTE	IAKA		4G LTE	IAKA
100~199	0.40	0.27	32.72%	2176	3220
200~299	0.43	0.30	30.60%	5264	7682
300~399	0.46	0.32	31.63%	5671	8400
400~499	0.47	0.32	31.95%	5669	8400
500~599	0.48	0.33	32.12%	5664	8400

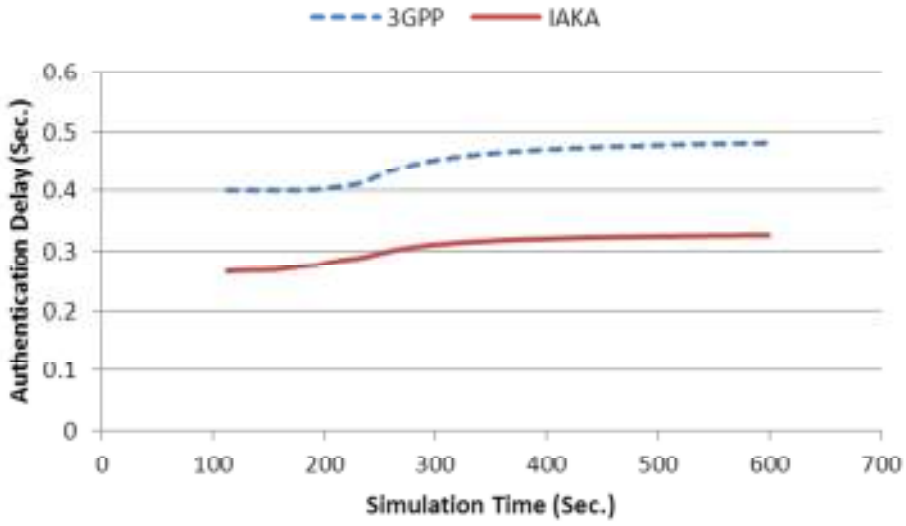

Figure 5-3 Authentication Delay (SPT = 20)

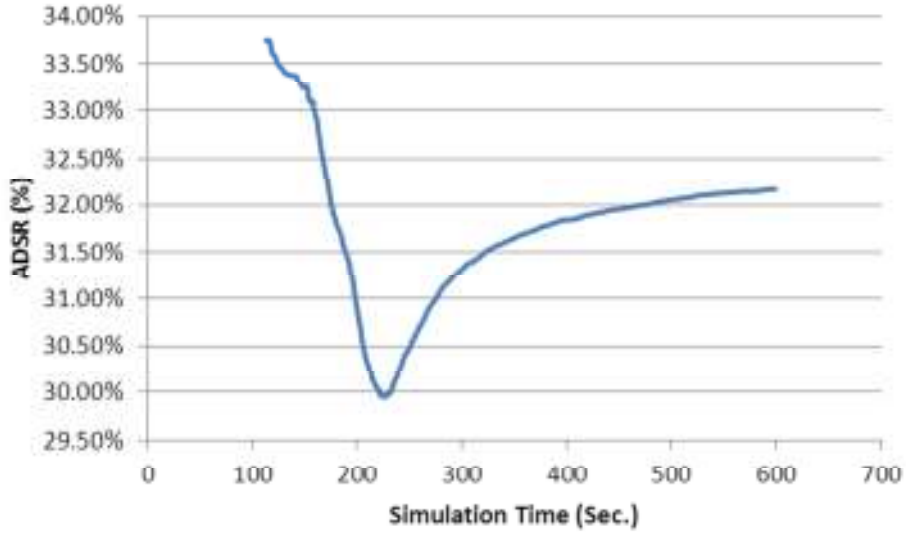

Figure 5-4 Authentication Delay Saving Rate (SPT = 20)

5.1.1.3 System Load SPT 30

When SPT = 30, each terminal generates 30 requests simultaneously. Under this scenario, the authentication delay begins to increase with increasing system load as shown in Fig. 5-5 and Fig. 5-6.

As described in Section 5.1.1.3, there is no traffic between 0~100 seconds simulation time because the application starts between 100 to 110 seconds. The results for 100 to 199 seconds are unreliable because the system is in an initialization phase. Also, the IAKA mode system is under higher system load because it processes more requests and responses than the 4G LTE mode system.

From simulation time 200 seconds, the authentication delay and the ADSR begin to increase with increasing system load. From Table 5-2, it is shown that at simulation time 200~299 seconds, the authentication delay of the IAKA approach is 0.30s and the 4G LTE approach is 0.43s while the IAKA mode system processes 7952 requests / responses per second and the 4G LTE mode system processes 5193 requests / responses per second. At simulation time 500~599 seconds, the authentication delay of the IAKA approach is 0.38s and the 4G LTE approach is 0.58s while the IAKA mode system processes 10850 requests / responses per second and the 3GPP mode system processes 7006 requests / responses per second.

Table 5-2 Average simulation results (SPT = 30)

Simulation time (second)	Authentication Delay (second)		ASDR (%)	System load of eNB/EPC (requests / responses per second)	
	4G LTE	IAKA		4G LTE	IAKA
100~199	0.40	0.27	32.72%	2176	3220
200~299	0.43	0.30	31.31%	5193	7952
300~399	0.52	0.34	34.21%	6856	10504
400~499	0.56	0.37	34.67%	7001	10850
500~599	0.58	0.38	34.89%	7006	10850

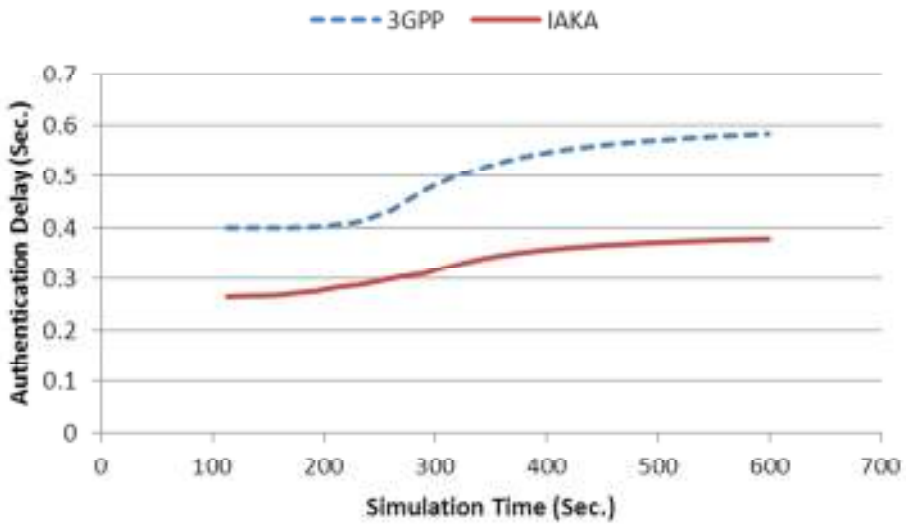

Figure 5-5 Authentication Delay (SPT = 30)

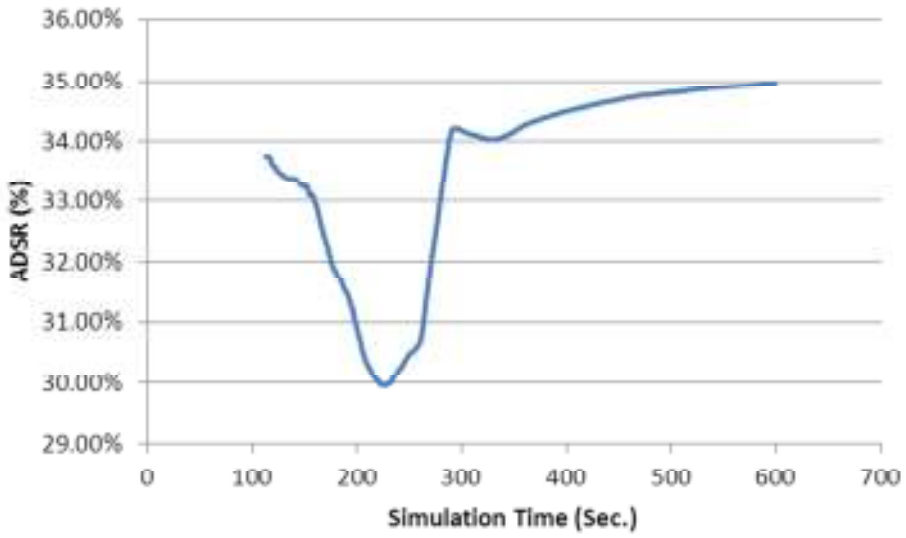

Figure 5-6 Authentication Delay Saving Rate (SPT = 30)

5.1.1.4 System Load SPT 40

When SPT = 40, each terminal generates 40 requests simultaneously. Under this scenario, the authentication delay begins to increase with increasing system load as shown in Fig. 5-7 and Fig. 5-8.

Also as described in Section 5.1.1.3, there is no traffic between 0~100 seconds simulation time because the application starts between 100 to 110 seconds. The results of 100 to 199 seconds are unreliable because the system is in an initialization phase. Furthermore, the IAKA mode system is under higher system load because it processes more requests and responses.

From simulation time 200 seconds, the authentication delay and the ADSR begin to increase with the increasing of the system load. From Table 5-3, it is shown that at simulation time 200~299 seconds, the authentication delay of the IAKA approach is 0.30s and the 4G LTE approach is 0.43s while the IAKA mode system processes 7952 requests / responses per second and the 3GPP mode system processes 5193 requests / responses per second. At simulation time 500~599 seconds, the authentication delay of the IAKA approach is 0.41s and the 4G LTE approach is 0.66s while the IAKA mode system processes 12201 requests / responses per second and the 3GPP mode system processes 7168 requests / responses per second.

Table 5-3 Average simulation results (SPT = 40)

Simulation time (second)	**Authentication Delay (second)**		**ASDR (%)**	**System load of eNB/EPC (requests/ responses per second)**	
	4G LTE	**IAKA**		**4G LTE**	**IAKA**
100~199	0.40	0.27	32.72%	2176	3220
200~299	0.43	0.30	31.31%	5193	7952
300~399	0.52	0.34	34.48%	7036	10972
400~499	0.61	0.38	36.72%	7200	12221
500~599	0.66	0.41	38.07%	7168	12201

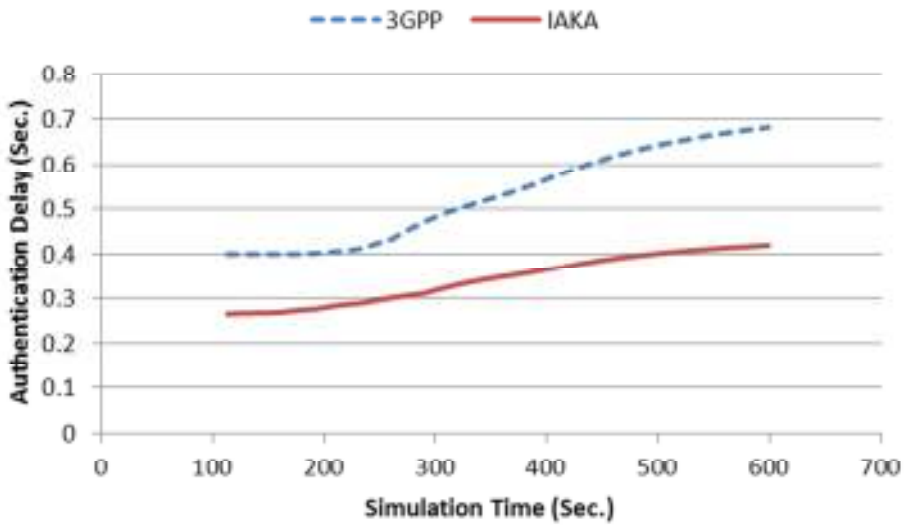

Figure 5-7 Authentication Delay (SPT = 40)

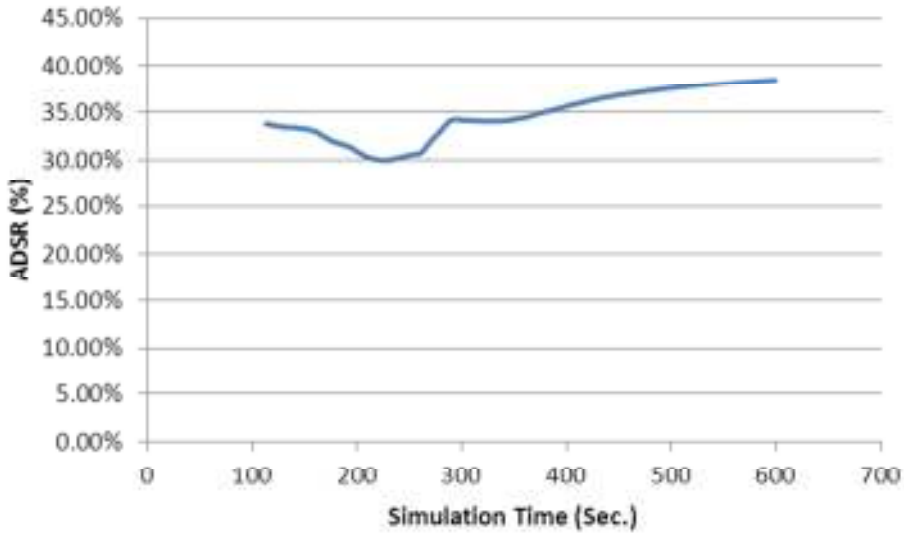

Figure 5-8 Authentication Delay Saving Rate (SPT = 40)

5.1.1.5 Summary

Based on the results shown above, it is identified that:

- In non-loaded condition, the average IMS layer authentication delay of the 4G LTE two-pass authentication protocol is 0.4s, the average delay of the proposed

IAKA is 0.267s. The proposed IAKA algorithm could save 33.34% of the IMS layer authentication delay.

- The authentication delay increased with increased system load as expected from Fig. 5-9. The effect of the average authentication delay under different system loads during the simulation period 500 to 599 seconds is shown in Table 5-4.

Table 5-4 Average simulation results during simulation period (500 second~599 second)

SPT	Authentication Delay (second)		ASDR (%)	System load of eNB/EPC (requests/ responses per second)	
	4G LTE	IAKA		4G LTE	IAKA
20	0.4798	0.3257	32.12%	5664	8400
30	0.5775	0.376	34.89%	7006	10850
40	0.6641	0.4112	38.08%	7168	12201

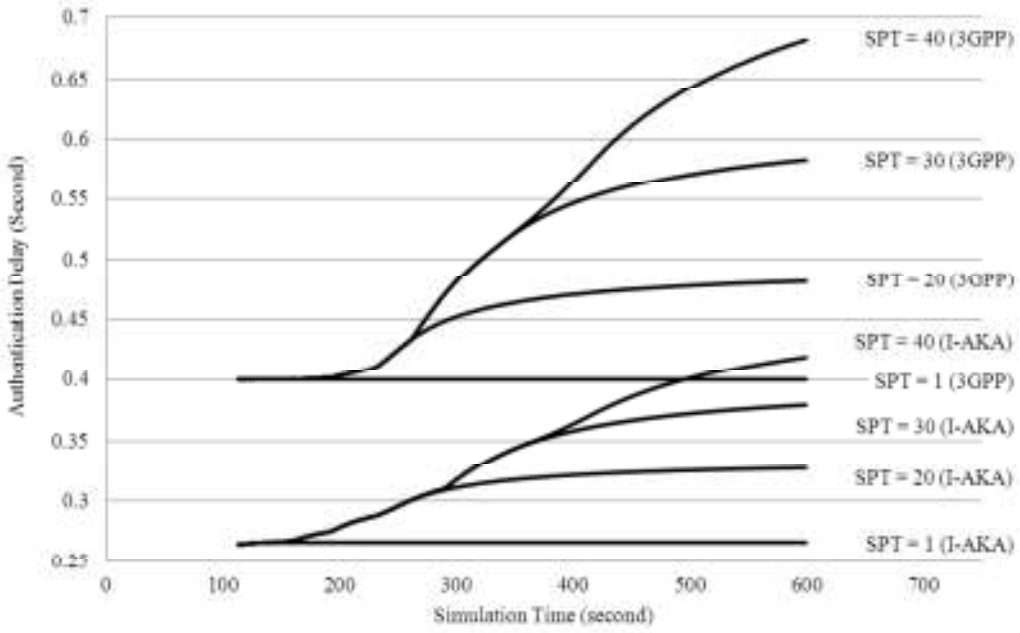

Figure 5-9 IMS layer Authentication Delay under fixed network delay configuration

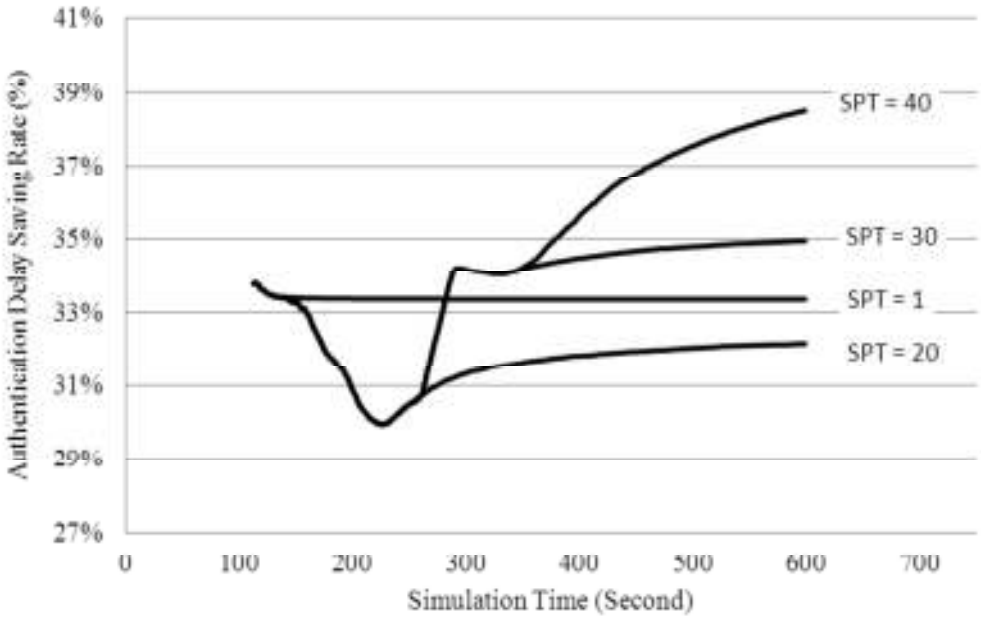

Figure 5-10 IMS layer Authentication Delay Saving Rate under fixed network delay configuration

- In the loaded network condition, as system load increases the ADSR increases as expected from Fig. 5-10. As shown in Table 5-4, the average ASDR is 38.08% when SPT = 40 which is much higher than the ASDR for SPT = 20 and SPT = 30.
- Under low system load (SPT = 20), the IMS authentication delay of the IAKA protocol increased faster than the 3GPP AKA protocol. The main reason for this outcome was found at the eNB / EPC which in the IAKA scenario processed more requests / responses during the same period which is shown in Table 5-4. The wireless network is more sensitive than the wired network in low system load conditions. Therefore, the ASDR under low system load (SPT =20) is lower than the ASDR without system load (SPT = 1).
- Under a high system load, the IMS authentication delay of the 3GPP protocol increased faster than the IAKA protocol. Although the wireless network IAKA approach processed more packets, the authentication steps saved in the proposed IAKA authentication protocol occurred in the wired network. Under certain system loads, the wired network is very sensitive to the packets processed. Therefore, the wired part of the authentication delay of the 3GPP approach increased faster than the proposed IAKA authentication approach and the ADSR

of SPT =30 and SPT = 40 is higher than the ADSR without system load (SPT = 1).

5.1.2 Varying Network Delay Configuration

For the varying network delay configuration the average network delay was used. It was assumed that $DELAY_{processing} = DELAY_{HSS} = DELAY_{ICSCFHSS} = DELAY_{AVretrival}$ = averageDelay. The average delay started from 1ms and the terminals carried out SIP registration procedures repeatedly and each procedure has a 1ms increment. In the non-loaded condition (SPT = 1), the statistics for the authentication delay and ADSR were collected from the UE and the results are shown in Fig. 5-11 and Fig. 5-12. Fig. 5-11 and Fig. 5-12 show that the authentication delay and the ADSR increase with increasing network delay. Under the non-loaded network environment, with increasing the network delay, the proposed IAKA authentication protocol could save up to 33% of the authentication delay.

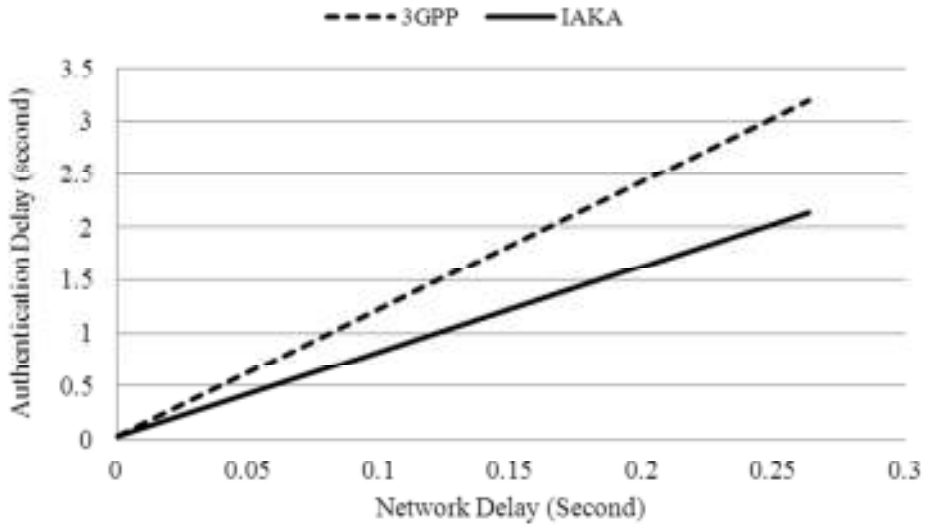

Figure 5-11 Authentication Delay with varying network delay configuration

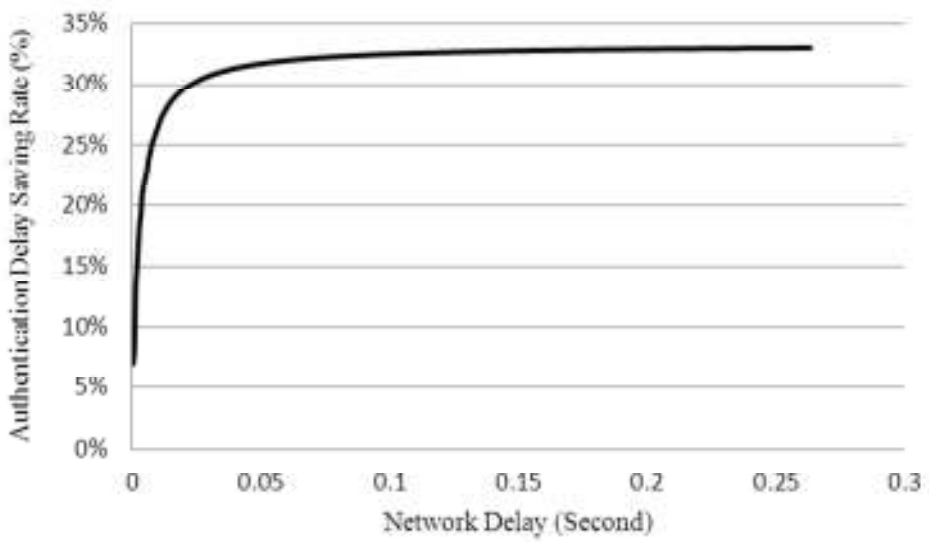

Figure 5-12 ADSR with varying network delay configuration

5.2 Security Analysis

This section describes the proposed IAKA authentication protocol security mechanism, and shows that the proposed IAKA protocol carries out authentication correctly, supports mutual authentication, confidential and integrity message protection, and prevents possible malicious attacks.

5.2.1 Authentication Accuracy

This section describes the proposed IAKA authentication process in the network and service layers. The UE authenticates with network entities in the network layer and it is assumed there is a mobile terminal with a pre-shared key K_{ue} and a network entity with the key K_{hss}. The network entity initiates a NAS Authentication Request to the UE with the random number RAND and authentication code AUTN as shown in (3) and (4).

$$AUTN = SQN \oplus AK \,||\, AMF \,||\, MAC \tag{3}$$

$$MAC = f1_K(SQN||RAND||AMF) \tag{4}$$

Where f1 is the message authentication function and K is the key used. After receiving the Authentication Request with the RAND and the AUTN, the terminal computes the AK and the SQN first, then calculates XMAC by using (4). The terminal checks whether the received MAC calculated by the server is equal to the XMAC

calculated by the terminal. If XMAC == MAC, the network entity is accepted and continues to communicate with the terminal. For a server with an invalid key identified by the IMPI number $K_{ue} \neq K_{hss}$, the calculated MAC won't be equal to the XMAC calculated by the terminal, and the terminal rejects the server connection and ceases the server connection and communication process.

After receiving the RES from the UE, the MME checks the RES number. If the RES is equal to the XRES received from the HSS, the terminal authentication is accepted and the terminal can access the network. XRES and RES are calculated by using (5).

$$XRES = f2_K (RAND) \quad (5)$$

Where f2 is a message authentication function and K is the key. If it is an invalid terminal without a valid key so that $K_{ue} \neq K_{hss}$, the RES wouldn't be equal to XRES and the MME would reject the terminal attach attempt. Therefore, only the terminal with the correct key identified by the same IMPI number can pass authentication and successfully login to the LTE network.

In the IMS service layer, the P-CSCF performs authentication for the network by checking the Integrity Checking Value (ICV), described in (6), received in SIP Register Request using the IPSec ESP protocol.

$$ICV = fint_K (M) \quad (6)$$

Where fint is the function used to generate the integrity code, M is the ESP packet minus the ICV and K is the key $K_{PCSCFint}$. A valid UE, after a successful network layer authentication, can derive an intermediate key K_{ASME} which is equal to the K_{ASME} stored in the P-CSCF and identified by the IMPI number. Using (2), both the terminal and the P-CSCF derive $K_{PCSCFint}$ by using the same K_{ASME}, hence, the $K_{PCSCFintUE} = K_{PCSCFintHSS}$. The P-CSCF, after receiving the ESP packet, retrieves the ICV from the ESP packet and calculates XICV by using (6), and if the terminal and the P-CSCF use the same key and the M wasn't changed during transmission, the ICV would be equal to XICV. Without a valid pre-shared key to generate the correct K_{ASME} and $K_{PCSCFint}$, ICV ≠ XICV, and the P-CSCF rejects the register request.

Finally, the UE authenticates the network in the IMS service layer by checking the ICV received in the SIP Register Response encapsulated by the IPSec ESP protocol packet. Using (6), when the UE and the P-CSCF have the same key and the input string M, the received ICV will be equal to the calculated XICV. Any server without a correct key K_{ASME} can't generate a valid ICV to pass authentication at the terminal, and the communication session is rejected.

5.2.2 Mutual authentication

As described, the network and service layers authentication using the proposed IAKA protocol support mutual authentication. Only the terminals and the network entities with the correct keys and a valid IMPI number authenticate successfully and build a trust relationship with each other. Without the valid pre-shared key identified by the IMPI number, adversaries cannot pretend to be a legal terminal and access the network and the service resources. Also, adversaries cannot pretend to be a valid network entity to get the user's private information.

5.2.3 Confidential and integrity protection

This section describes the confidential and integrity protection of the proposed IAKA IMS service layer authentication. The network layer is not discussed in this section because the confidential and integrity protection of the IAKA network layer is the same as the 3GPP definition.

The proposed IAKA IMS layer authentication protocol creates a pair of IPSec security associations to protect the information transmitted between the terminal and the P-CSCF. Before transmission, the sender encrypts the data by using the symmetric encryption algorithm and the key $K_{PCSCFenc}$. Only a receiver with the same key $K_{PCSCFenc}$ can decrypt the message and obtain the correct plain text.

The messages between the terminal and the P-CSCF are also integrity protected in the IPSec SAs. The receiver checks the message's integrity by using the ICV number. If the received ICV doesn't equal to the calculated XICV, the received message is incomplete and will be discarded.

5.2.4 Possible attacks

This section analysed how the proposed IAKA prevented the possible replay attack and DOS attack.

A replay attack can't be achieved in the network and service layers because in the network layer, if an adversary was eavesdropping on the conversation between the terminal and the MME and captured the NAS Authentication Request and NAS Authentication Response (RES), it might initiate a replay attack by sending the Authentication Request (RAND, AUTN and KSI_{ASME}) packet to the terminal; however, the adversary couldn't pass a SQN number check which is a part of the AUTN. As defined in [TS 33.102 2009], the SQN should be different in the new session and the terminal should check the SQN number validity first. If the received SQN is the same as the previously accepted SQN number, the terminal would send an error response to the fake server. The adversary might do a replay attack by sending the Authentication Response packet to the server, however, a RES number check would identify this as a fake message. Using (5), RES is calculated by using the HSS randomly generated RAND which is different for each session. Therefore in a new session, the captured packet with the old RES number would be identified as a fake message and access to the services would be prevented.

In the service layer, adversaries can't get the user or the server's information by sending the captured packets because the packets are protected by using IPSec ESP which includes a monotonically increasing sequence number. After receiving a packet, the receiver verifies the sequence number first to make sure this is not a duplicated packet; hence, the captured packet with the old sequence number would not be accepted. Therefore, the proposed IAKA authentication protocol can protect the system from the malicious replay attack.

The proposed IAKA protocol has higher security against a DOS attack than the legacy 3GPP defined authentication protocol. With the legacy 3GPP defined authentication protocol the IMS core system is vulnerable to a DOS attack. Assume the adversary floods the P-CSCF/I-CSCF/S-CSCF/HSS by sending SIP REGISTER requests with a valid IMPI number which belongs to a valid subscriber X. The P-CSCF would forward the requests to the core network servers I-CSCF/S-CSCF/HSS. However, for

IAKA authentication, the flood messages can only reach the P-CSCF and the MME, leaving the core network fully protected. If the subscriber X hasn't logged into the LTE network and the P-CSCF is not able to retrieve the subscriber's AV from the MME, the P-CSCF considers this to be an invalid terminal and will reject the Register request. If the subscriber X has already registered with the LTE network, the adversary is required to transmit SIP Register requests in the IPSec security associations, and without the valid key, the messages transmitted to the P-CSCF will not be validated and forwarded to the next hop. Therefore the adversaries cannot jeopardize the core I-CSCF / S-CSCF / HSS servers.

5.3 Energy Consumption Analysis

In this section, the energy cost of the user terminal's authentication related security activities was calculated. It is shown that the IAKA authentication protocol could save 81.82% of the energy consumption in IMS layer authentication and save 39.13% of the terminal's energy consumption in both the network layer and service layer.

For the 3GPP defined 4G LTE and proposed IAKA authentication protocols, the terminal's security activities include the execution of the EPS AKA and the IMS AKA authentication protocol. Therefore, $E = E_{EPS\text{-}AKA} + E_{IMS\text{-}AKA}$ where E denotes the energy consumption.

In the 3GPP defined authentication protocol, the terminal's energy cost to execute the EPS AKA authentication protocol includes (a) checking the AUTN number which is made up of generating AK and MAC (b) generating the RES number, and (c) generating CK, IK and deriving K_{ASME}.

$$E_{4G\text{-}EPS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} + E_{KASME} \quad (7)$$

The energy cost to execute the IMS AKA authentication protocol is shown in (4).

$$E_{4G\text{-}IMS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} \quad (8)$$

For the IMS layer, the IPSec energy consumption wasn't considered because IPSec is mainly used for message transmission and it is the same in the two approaches.

As described in Section 4.2.8, this research work uses the AV generation functions f1, f2, f3, f4, f5 to generate MAC, RES, CK, AK, and choose AES as the kernel encryption algorithm. Also, HMAC-SHA-256 is used as the key derivation function to derive K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$. For the energy consumption calculation, only the energy used in encryption is considered. The security operations of bitwise exclusive-OR and bitwise rotation aren't calculated as they won't affect the results significantly. The energy consumption of AES and HMAC was studied in [Potlapally 2006], and the results showed the energy consumption of the AES algorithm is made up of two phases, the first is in the key setup phase $E_{KEY-SETUP}$ which is 7.87 μJ and the second is in the encryption/decryption phase $E_{ENC/DEC}$. In the encryption/decryption phase, the energy consumption per byte (EPB) is 1.21 μJ and for the HMAC-SHA-256, the EPB is 1.16 μJ. Therefore the energy can be calculated as shown in (9) and (10).

$$E_{AES}(n) = 7.87\ \mu J + 1.21\ \mu J * n \quad (9)$$

$$E_{HMAC}(n) = 1.16\ \mu J * n \quad (10)$$

Where n is the length of the input string in bytes. Although the MAC, RES, CK, IK, and AK have different lengths, they were all generated by encrypting 16 byte data blocks three times. Therefore, referring to (9), $E_{AK} = E_{MAC} = E_{RES} = E_{CK} = E_{IK} = 81.69$ μJ. For K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$, they were derived by using 32 byte data blocks so the energy consumption is 37.12 μJ by using (10). For the 3GPP defined 4G LTE authentication protocol, referring to (7) and (8), (a) $E_{4G-EPS-AKA} = 445.57$ μJ, and (b) $E_{4G-IMS-AKA} = 408.45$ μJ.

For the improved IAKA authentication protocol, the LTE network layer energy consumption is same as it in the 3GPP approach which is 445.57 μJ ($E_{IAKA-EPS-AKA} = 445.57$ μJ). The energy cost of the IMS service layer includes key derivation of $K_{PCSCFint}$ and $K_{PCSCFenc}$ by using HMAC-SHA-256 which is 74.24 μJ ($_{EIAKA-IMS-AKA} = 74.24$ μJ).

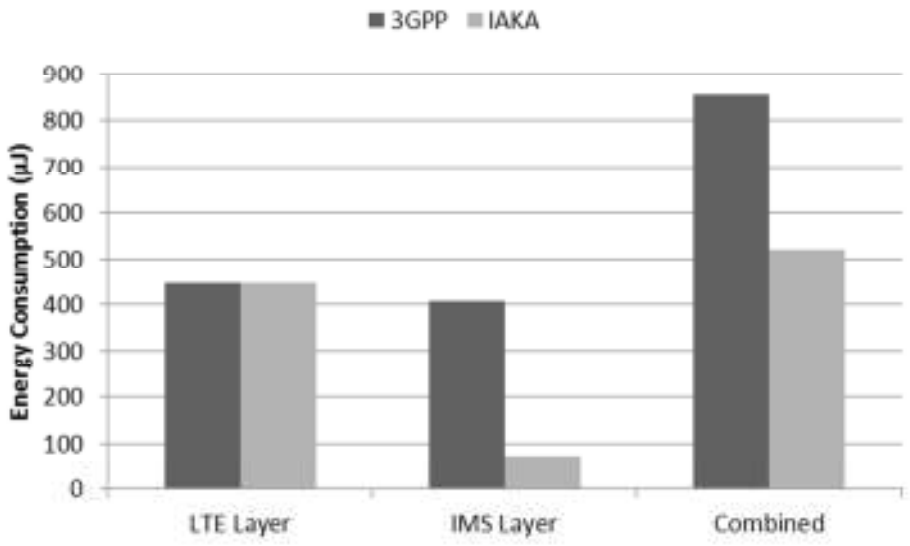

Figure 5-13 Energy Consumption Comparison

$$ESR_{I\text{-}IMS\ AKA} = (E_{4G\text{-}IMS\text{-}AKA} - E_{I\text{-}IMS\text{-}AKA}) / E_{4G\text{-}IMS\text{-}AKA} * 100\% \quad (11)$$

$$ESR_{IAKA} = (E_{4G} - E_{IAKA}) / E_{3GPP} * 100\% \quad (12)$$

By using (11) and (12), $ESR_{I\text{-}IMS\ AKA} = 81.82\%$ and $ESR_{IAKA} = 39.13\%$. This means that by using the improved authentication protocol, the terminal could save 81.82% of IMS layer security related activity energy consumption; and save 39.13% of the security activity energy consumption in both the network layer authentication and the IMS service layer authentication by using the improved authentication protocol

6 Conclusion

The scope and objectives of this research were successfully completed. The research provided an improved one-pass IMS authentication protocol for the LTE system with higher performance, higher security and lower energy consumption. A comprehensive investigation of the current 3GPP defined 4G LTE two-pass AKA authentication protocol and the related research work were carried out in this research; and the proposed authentication protocol was simulated to prove its achievability. Furthermore the performance analysis, security performance and energy consumption were provided by comparing the proposed authentication protocol and the 4G LTE authentication protocol by using the system model identified in this research.

An important outcome of the literature review was identifying previous one pagss authentication approaches lacked security or required significant system modification. Furthermore, most of the previous research work was focused on the 3G network or the network layer authentication.

The initial research results were presented at the International Conference on Information Networking (ICOIN 2011) conference, which was held in Kuala Lumpur, Malaysia in January 2011 and the conference paper [Gu 2011a] is provided in Appendix B. The research results about the energy consumption and security were presented at Australian Telecommunication Network and Application Conference (ATNAC 2011) in Melbourne, Australia in November 2011 and the conference paper [Gu 2011b] is shown in Appendix C. The research results presented in this thesis have been included in a journal paper [Gu 2011c] and the revised version was submitted to the IEEE Transactions on Wireless Communications at 7 Dec. 2011, Appendix D.

The research was focused on 4G LTE network and IMS authentication, and the research outcome was an improved one-pass authentication protocol that improves 4G system performance, security, and reduces UE energy consumption. The results provided in this thesis include:

- A literature review of the mobile network which starts from 1G through to 4G and includes a review of the 4G LTE network and IMS network.

- The introduction and analysis of the 4G LTE IMS two-pass IAKA authentication protocol, the security mechanism and why this approach is important;
- An introduction and analysis of the related research work and identifies their limitations;
- Proposed the IAKA one-pass authentication protocol with security key hierarchy, LTE EPS AKA authentication procedure and the improved IMS AKA authentication procedure;
- A comprehensive system model was designed and a LTE-IMS model was developed using OPNET Modeller v16. The models were used to compare the system performance between different authentication approaches;
- The proposed IAKA authentication proposed was developed successfully and models, nodes and protocols were developed using C/C++ within OPNET Modeller v16 and integrating with the OPNET provided LTE-IMS models;
- A comprehensive comparison of the system performance, security, and energy consumption between the proposed authentication protocol and the current standardized 4G LTE authentication protocol;
- It was found that the proposed IAKA one-pass authentication protocol simplifies the 4G LTE / IMS authentication procedures significantly;
- On system performance, by comparing the IAKA one-pass authentication protocol with the current standardized 4G LTE authentication protocol, the following conclusions are made:
    - For the system's packet processing delay configured to a fixed value
        - In non-loaded environment, the proposed IAKA could save up to 33.34% of the authentication delay;
        - The authentication delay increased with increasing system load

        - In the loaded system environment, the proposed IAKA could reduce authentication delay with higher system load
        - In the loaded system environment, the proposed IAKA could reduce the authentication delay by 38%.
    - If the system's packet processing delay is configured to a varying value which starts from 1ms with 1ms increments for each procedure, the proposed IAKA one-pass authentication protocol could reduce the authentication delay by 33%.

- On security, it was proved that the proposed IAKA one-pass authentication protocol could not only authenticate the end user correctly but also support mutual authentication, confidential and integrity message protection and prevents possible malicious attack. Furthermore, the proposed IAKA provides stronger system protection against the DOS attack.
- On energy consumption, it was found that the proposed IAKA one-pass authentication protocol could save up to 81.82% in the IMS service layer authentication and save up to 39.13% in the combined two layer authentication processes.

7 Future Work

A number of future research topics were identified during this research. Future research may include further investigation of the improved one-pass authentication protocol, and the following is a suggested list of possible research directions.

- Current research mainly focuses on the home users that the end user and the HSS server are in the same network domain. The authentication for the roaming users including a visited end user from different area of the same carrier, a visited end user from a different carrier with different network environment consideration may be further studied.

- The next generation network is a highly integrated network. WLAN network, mobile network, fibre network and the other networks can be seamlessly integrated under one core network to provide the end user the most flexible services. The current multi-pass authentication protocol will be a big burden and there will be high requirements for a generic authentication protocol for the highly integrated next generation network. The prospective research work my include the design and develop a generic authentication protocol for mobile network, wireless network, fibre network and the other networks; simulation of the proposed authentication protocol to prove its achievability, and a comprehensive analysis of the proposed authentication protocol with the other authentication approaches.

References:

[3GPP 2011] http://www.3gpp.org, Accessed Oct 04 2011

[Copeland 2009] Rebecca Copeland, *Converging NGN Wireline and Mobile 3G Network with IMS*, Taylor & Francis Group, U.S.A, 2009

[Eastlake 2011] Donald Eastlake, Tony Hansen, "US Secure Hash Algorithms (SHA and SHA based HMAC and HKDF)", draft-eastlake-sha2b-07, 15 Feb 2011

[Foster 2002] G. Foster, M.I. Pous, D. Pesch, A. Sesmun, and V. Kenneally, "Performance Estimation of Efficient UMTS Packet Voice Call Control," Proc. IEEE Vehicular Technology Conf., vol. 3, pp. 1447-1451, Sep 2002

[Gladman 2011] Brian Gladman, Advanced Encryption Standard Source code, http://www.pudn.com/downloads36/sourcecode/crypt/detail113322.html, Accessed Mar 22 2011

[Gu 2011a] Lili Gu and Mark A Gregory, "Improved One-Pass IP Multimedia Subsystem Authentication for UMTS", International Conference on Information Networking, Kuala Lumpur Malaysia, Jan 2011

[Gu 2011b] Lili Gu and Mark A Gregory, "A Green and Secure Authentication for the 4Th Generation Mobile Network", Australasian Telecommunication Networks And Applications Conference, Melbourne Australia, Nov. 2011, accepted

[Gu 2011c] Lili Gu and Mark A Gregory, "Optimized Authentication and Key Agreement Protocol for 4G Long Term Evolution and IP Multimedia Subsystem", IEEE Transactions on Wireless Communications, Jun. 2011, submitted

[Huang 2007] Chung-Ming Huang and Jian-Wei Li, "Efficient and Provably Secure IP Multimedia Subsystem Authentication

for UMTS", The Computer Journal, Vol 50, No.6, pp739–757, 2007

[Huang 2009] Chung-Ming Huang and Jian-Wei Li, "Reducing Signaling Traffic for the Authentication and Key Agreement Procedure in an IP Multimedia Subsystem", Wireless Personal Communications, Vol51, pp95-107, 2009

[ITU-R M.2134] ITU-R M.2134, "Requirements related to technical performance for IMT-Advanced radio interface(s)", 2008

[Khilifi 2008] Hechmi Khilifi, Jean-Charles Gregoire, IMS Application Servers Roles, Requirements, and Implementation Technologies, IEEE Computer Society, May/June 2008

[Lescuyer 2008] Pierre Lescuyer and Thierry Lucidarme, Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS, John Wiley &Sons, Ltd, 2008

[Lin 2005] Y.B. Lin, M.F. Chang, M.T. Hsu, L.Y. Wu, "Onepass GPRS and IMS Authentication Procedure for UMTS", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 6, pp: 1233-1239, Jun 2005.

[Ntantogian 2009] Christoforos Ntantogian and Christos Xenakis, "One-pass EAP-AKA Authentication in 3G-WLAN Integrated Networks", Wireless Personal Communications, Vol48, pp569-584, 2009

[Ntantogian 2010] Christoforos Ntantogian, Christos Xenakis, and Ioannis Stavrakakis, "A generic mechanism for efficient authentication in B3G networks", Computers and Security, Vol29, pp460-475, 2010

[OPNET 2011] OPNET Technologies, Inc., Opnet Modeler - ver. 16.0, http://www.opnet.com, Accessed Mar 22 2011

[RAPPAPORT 2002] THEODORE S. Rappaport, *Wireless Communications Principles and practice Second Edition*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2002

[Potlapally 2006] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "A Study of the Energy Consumption Characteristics of

Cryptographic Algorithms and Security Protocols," IEEE Trans. on Mobile Computing, vol. 5, no 2, pp. 128-143, Mar-Apr 2006

[RFC 2406] S. Kent , R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC2406, Nov 1998

[RFC 3552] E. Rescorla, B. Korver, "Guidelines for Writing RFC Text on Security Considerations", RFC3552, Jul 2003

[TR 33.978 2008] 3GPP TR 33.978 V8.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security aspects of early IP Multimedia Subsystem (IMS) (Release 8)", Dec 2008

[TS 23.203 2011] 3GPP TS 23.203 V11.1.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 11)", Mar. 2011

[TS 23.228 2010] 3GPP TS 23.228 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 10)", Mar 2010

[TS 23.234 2009] 3GPP TS 23.234 V9.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP system to Wireless Local Area Network (WLAN) interworking; System description (Release 9)", Dec 2009

[TS 23.401 2010] 3GPP TS 23.401 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 10)", Jun 2010

[TS 24.301 2010] 3GPP TS 24.301 V9.3.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for

Evolved Packet System (EPS); Stage 3 (Release 9)", Jun 2010

[TS 24.229 2010] 3GPP TS 24.229 V9.3.1 "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 9)", Mar 2010

[TS 29.228 2010] 3GPP TS 29.228 V9.2.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signalling flows and message contents (Release 9)", Jun 2010

[TS 29.229 2010] 3GPP TS 29.229 V9.2.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Cx and Dx interfaces based on the Diameter protocol;Protocol details (Release 9)", Jun 2010

[TS 29.272 2010] 3GPP TS 29.272 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Evolved Packet System (EPS); Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter protocol (Release 10)", Sep 2010

[TS 29.274 2010] 3GPP TS 29.274 V10.1.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 (Release 10)", Dec 2010

[TS 33.102 2009] 3GPP TS 33.102 V9.1.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 9)", Dec 2009

[TS 33.105 2009] 3GPP TS 33.105 V9.0.0, “3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;3G Security; Cryptographic algorithm requirements (Release 9)”, Dec 2009

[TS 33.203 2009] 3GPP TS 33.203 V9.3.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 9)”, Dec 2009

[TS 33.210 2010] 3GPP TS 33.210 V10.0.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Network Domain Security (NDS); IP network layer security (Release 10)”, Oct 2010

[TS 33.220 2010] 3GPP TS 33.220 V9.3.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 9)”, Jun 2010

[TS 33.401 2010] 3GPP TS 33.401 V9.4.0 “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 9)”, Jun 2010

[TS 35.205 2009] 3GPP TS 35.205 V9.0.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General (Release 9)”, Dec 2009

[TS 35.206 2009] 3GPP TS 35.206 V9.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1,

f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 9)", Dec 2009

[Vazquez 2005] AH Enrique Vazquez, Jose Ignacio Fernandez, "SIP-IMS Model for OPNET Modeler," OPNET University Program Contributed Models, 2005

[Veltri 2006] L. Veltri, S. Salsano, G. Martiniello, “Wireless LAN-3G integration: unified mechanisms for secure authentication based on SIP”, International Conference on Communications, (ICC). Istanbul, Turkey; Jun 2006

Appendix A

Source Code in Opnet

This section presents the source code developed in OPNET to create the LTE and IMS system to do the authentication simulation.

A.1: Source Code for application SIP REGISTER

A.1.1: Function “sip_request_register”

```
/*send request of register to SIP UAC*/
SIPT_Call_Info_Shell* sip_request_register (Prohandle register_CP_prohndl, Prohandle sip_UAC_mgr_prohndl, const char* invitee_addr, OmsT_Qm_Tos tos)
{
	static int 			call_id = 0;
	Ici* 				sip_req_ici_ptr = OPC_NIL;
	Prohandle* 			register_CP_prohndl_ptr = OPC_NIL;
	SIPT_Call_Info* 		sip_call_info_ptr = OPC_NIL;
	SIPT_Call_Info_Shell* 		sip_call_info_shell_ptr = OPC_NIL;
	Objid 				call_initiator_node_objid;
	Objid 				call_initiator_tpal_module_id;

	/* This function sends an REGISTER request to the specified UAC */
	FIN (sip_request_register (inviting_CP_prohndl, sip_UAC_mgr_prohndl));

	/* Check if the specified UAC mgr exists */
	if (op_pro_valid (sip_UAC_mgr_prohndl) == OPC_FALSE)
	{
		/* Issue an error message and return NIL */
		sip_service_unsupported_log_write ();
		FRET (OPC_NIL);
	}

	/* Allocate memory to store the call information */
	sip_call_info_ptr = (SIPT_Call_Info*) op_prg_mem_alloc (sizeof (SIPT_Call_Info));
	/* Assign a unique ID to the call */
	sip_call_info_ptr->call_id = call_id++;
	/*set network delay = 0 ms*/
	sip_call_info_ptr->network_delay = -1;
	/* Assign the invitee addr to the call */
	sip_call_info_ptr->invitee_addr = (char*) invitee_addr;
	/* Reset the via field */
	sip_call_info_ptr->via_icscf = OPC_FALSE;
	/* Assign the type of service to the call */
	sip_call_info_ptr->tos = tos;
	/* Get the node objid */
	call_initiator_node_objid = op_topo_parent (op_id_self ());
	/* Get the tpal id of the node initiating the call */
	call_initiator_tpal_module_id = sip_tpal_module_id_get (call_initiator_node_objid);
	/* Get the TPAL address */
	sip_call_info_ptr->call_initiator_addr = sip_tpal_addr_get (call_initiator_tpal_module_id);
```

```
	/* Time stamp the init time of this call request */
	sip_call_info_ptr->call_init_time = op_sim_time ();
	/* Create the handle to be returned */
	sip_call_info_shell_ptr = (SIPT_Call_Info_Shell*) op_prg_mem_alloc (sizeof
(SIPT_Call_Info_Shell));

	/* Assign the call information to the shell */
	sip_call_info_shell_ptr->sip_call_info_ptr = sip_call_info_ptr;

	/* Set the status of the call to indicate that the call has just now been initiated */
	sip_call_info_shell_ptr->call_status = SIPC_Call_Initiated;

	/* Create an ICI to communicate with the UAC */
	sip_req_ici_ptr = op_ici_create ("sip_req");

	/* Allocate memory to pass the caller's prohandle */
	/* It will be deallocated by the UAC */
	register_CP_prohndl_ptr = (Prohandle*) op_prg_mem_alloc (sizeof (Prohandle));
	*register_CP_prohndl_ptr = register_CP_prohndl;

	/* Set the fields of the ICI */
	op_ici_attr_set (sip_req_ici_ptr, "appl_prohandle_ptr", register_CP_prohndl_ptr);
	op_ici_attr_set (sip_req_ici_ptr, "request_type", SIPC_Request_Type_Register);
	op_ici_attr_set (sip_req_ici_ptr, "call_info_ptr", sip_call_info_shell_ptr);

	/* Install the ICI */
	op_ici_install (sip_req_ici_ptr);

	/* Schedule an interrupt to the UAC with the INVITE request */
	op_intrpt_schedule_process (sip_UAC_mgr_prohndl, op_sim_time (), SIPC_REQUEST);

	FRET (sip_call_info_shell_ptr);
}/* End sip_request_register () */
```

A.1.2: Function “sip_request_register_finish”

```
/*The function “sip_request_register_finish” is invoked by the voice process to terminate SIP registration
repletion in SIP UAC.*/

Void sip_request_register_finish (Prohandle leaving_CP_prohndl, Prohandle sip_UAC_mgr_prohndl,
SIPT_Call_Info_Shell* sip_call_info_shell_ptr)
{
	Ici*		sip_req_ici_ptr = OPC_NIL;
	Prohandle*	leaving_CP_prohndl_ptr = OPC_NIL;

	FIN (sip_request_register_finish (inviting_CP_prohndl, sip_UAC_mgr_prohndl,
	sip_call_info_shell_ptr));

	/* Create an ICI to communicate with the UAC */
	sip_req_ici_ptr = op_ici_create ("SIP_req");
	/* Allocate memory to pass the caller's prohandle */
	/* It will be deallocated by the UAC */
	leaving_CP_prohndl_ptr = (Prohandle*) op_prg_mem_alloc (sizeof (Prohandle));
	*leaving_CP_prohndl_ptr = leaving_CP_prohndl;
	/* Set the fields of the ICI */
	op_ici_attr_set (sip_req_ici_ptr, "appl_prohandle_ptr", leaving_CP_prohndl_ptr);
```

```
	/* Set the fields of the ICI */
	op_ici_attr_set (sip_req_ici_ptr, "request_type", SIPC_Request_Type_Reg_Finish);
	/* Set the fields of the ICI */
	op_ici_attr_set (sip_req_ici_ptr, "call_info_ptr", sip_call_info_shell_ptr);
	/* Install the ICI */
	op_ici_install (sip_req_ici_ptr);
	/* Schedule an interrupt to the UAC with the BYE request */
	op_intrpt_schedule_process (sip_UAC_mgr_prohndl, op_sim_time (), SIPC_REQUEST);
	FOUT;
}/* End sip_request_register_finish () */
```

A.2: Source Code for DIAMETER_UAS_mgr model

```
static void diameter_UAS_spawn_passive ()
{
	Prohandle		diameter_UAS_prohndl;
	char			msg [128];

	/* This function spawns a UAS process which can LISTEN to incoming requests */
	FIN (diameter_UAS_spawn_passive ());

	/* Spawn a diameter UAS */
	diameter_UAS_prohndl = op_pro_create ("diameter_UAS", UAS_ptc_mem_ptr);
	if (ltrace_diameter_UAS_mgr_active)
	{
		sprintf(msg, "UAS_mgr (PID %d) has spawned UAS (PID %d)", my_pro_id, op_pro_id
		(diameter_UAS_prohndl));
		op_prg_odb_print_major(msg, OPC_NIL);
	}

	/* Invoke the diameter UAS so that it can initialize itself */
	op_pro_invoke (diameter_UAS_prohndl, OPC_NIL);

	FOUT;
}/* End diameter_UAS_spawn_passive ()*/

static void diameter_UAS_mgr_attrs_parse ()
{
	Objid		my_node_objid;
	Objid		comp_attr_objid;
	Objid		row_objid;
	char		temp_str[256];

	/* This function parses the diameter attributes set on this node */
	FIN (diameter_UAS_mgr_attrs_parse ());

	/* Get the objid of the node on which this UAC mgr runs*/
	my_node_objid = op_topo_parent (op_id_self ());

	/* Get the diameter Parameters attribute */
	op_ima_obj_attr_get (my_node_objid, "Diameter Proxy Server Parameters", &comp_attr_objid);

	/* Get the first (and only) row */
	row_objid = op_topo_child (comp_attr_objid, OPC_OBJTYPE_GENERIC, 0);
```

```
	/* Get the max number of simultaneous calls this node can handle */
	op_ima_obj_attr_get (row_objid, "Maximum Simultaneous Calls", &open_lines_count);

	/* Check if the Maximum Simultaneous Calls is set to Unlimited */
	if (open_lines_count == -1)
	{
		/* Set it to the largest possible integer */
		open_lines_count = OPC_INT_INFINITY;
	}

	/* Get domain name */
	op_ima_obj_attr_get (row_objid, "Domain Name", temp_str);
	domain_name = (char*) op_prg_mem_alloc (strlen (temp_str) + 1);
	strcpy (domain_name, temp_str);

	/* Get area name */
	 op_ima_obj_attr_get (row_objid, "Area Name", temp_str);
	area_name = (char*) op_prg_mem_alloc (strlen (temp_str) + 1);
	strcpy (area_name, temp_str);

	/* Get proxy type */
	op_ima_obj_attr_get (row_objid, "Proxy Type", temp_str);
	proxy_type = (char*) op_prg_mem_alloc (strlen (temp_str) + 1);
	strcpy (proxy_type, temp_str);

	/* Get processing delay */
	op_ima_obj_attr_get (row_objid, "Processing Delay", &processing_delay);

	FOUT;
}/* End diameter_UAS_mgr_attrs_parse ()*/

static void diameter_UAS_mgr_stats_register ()
{

	/* This function register all the stats for the UAS */
	FIN (diameter_UAS_mgr_stats_register ());

	/* Register the Call Setup Requests stat */
	UAS_call_setup_requests_stathandle = op_stat_reg ("diameter UAS.Call Setup Requests",
	OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

	/* Register the Calls Connected stat */
	UAS_calls_connected_stathandle = op_stat_reg ("diameter UAS.Calls Connected",
	OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

	/* Register the Calls Rejected stat */
	UAS_calls_rejected_stathandle = op_stat_reg ("diameter UAS.Calls Rejected",
	OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

	/* Register the Active Calls stat */
	UAS_active_calls_stathandle = op_stat_reg ("diameter UAS.Active Calls",
	OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

	/* Register the Active Calls stat */
	UAS_call_duration_stathandle = op_stat_reg ("diameter UAS.Call Duration (sec)",
	OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
```

```
	/* Initialize the active calls stat */
	op_stat_write (UAS_active_calls_stathandle, 0);

	FOUT;
} /* End diameter_UAS_mgr_stats_register ()*/

static void diameter_UAS_mgr_ptc_mem_create ()
{
	/* This function creates the parent-to-child memory */
	/* which is passed between the UAS_mgr and the UAS  */
	FIN (diameter_UAS_mgr_ptc_mem_create ());

	/* Allocate memory for the UAC parent-to-child memory */
	UAS_ptc_mem_ptr = (DIAMETERT_UAS_PtC_Mem*) op_prg_mem_alloc
	(sizeof(DIAMETERT_UAS_PtC_Mem));

	/* Assign the TPAL address of the node on which this process runs */
	UAS_ptc_mem_ptr->strm_to_tpal_ptr = &strm_to_tpal;

	/* Assign the open lines counter */
	UAS_ptc_mem_ptr->open_lines_count_ptr = &open_lines_count;

	/* Assign the domain name */
	UAS_ptc_mem_ptr->domain_name_ptr = domain_name;

	/* Assign the area name */
	UAS_ptc_mem_ptr->area_name_ptr = area_name;

	/* Assign the processing delay */
	UAS_ptc_mem_ptr->processing_delay_ptr = &processing_delay;

	/* Assign the proxy type */
	UAS_ptc_mem_ptr->proxy_type_ptr = proxy_type;

	/* Assign the active calls counter */
	UAS_ptc_mem_ptr->active_calls_count_ptr = &active_calls_count;

	/* Assign the stathandles */
	UAS_ptc_mem_ptr->call_setup_requests_stathandle = UAS_call_setup_requests_stathandle;
	UAS_ptc_mem_ptr->calls_connected_stathandle =
	UAS_calls_connected_stathandle;
	UAS_ptc_mem_ptr->calls_rejected_stathandle = UAS_calls_rejected_stathandle;
	UAS_ptc_mem_ptr->active_calls_stathandle = UAS_active_calls_stathandle;
	UAS_ptc_mem_ptr->call_duration_stathandle = UAS_call_duration_stathandle;

	FOUT;
}/* End diameter_UAS_mgr_ptc_mem_create ()*/

static void diameter_UAS_mgr_ltrace_activate ()
{
	/* This function activates the ODB traces */
	FIN (diameter_UAS_mgr_ltrace_activate ());

	/* Set the full trace */
	ltrace_diameter_UAS_mgr_active = op_prg_odb_ltrace_active ("diameter");

	/* Set the error trace */
```

```
	ltrace_diameter_UAS_mgr_errors_active = op_prg_odb_ltrace_active ("diameter_error");

	/* Set the INVITE trace */
	ltrace_diameter_UAS_mgr_invite_active = op_prg_odb_ltrace_active ("diameter_invite");

	/* Set the BYE trace */
	ltrace_diameter_UAS_mgr_bye_active = op_prg_odb_ltrace_active ("diameter_bye");

	/* Set the all REQUESTS trace */
	ltrace_diameter_UAS_mgr_req_active = op_prg_odb_ltrace_active ("diameter_req");

	/* Set the all RESPONSES trace */
	ltrace_diameter_UAS_mgr_resp_active = op_prg_odb_ltrace_active ("diameter_resp");

	FOUT;
}/* End diameter_UAS_mgr_ltrace_activate ()*/
```

A.3: Source Code for DIAMETER_UAS

```
/***** Requests Related Functions *****/
static void diameter_UAS_req_process (Packet* pk_ptr)
{
	char		msg [128];
	int		request_type;

	/* This function updates the message on the packet   */
	/* based on the request type. The message informs    */
	/* the destination UAC of the request of the src UAC */
	FIN (diameter_UAS_req_process (pk_ptr));

	/* Get the request type from the packet */
	op_pk_nfd_get (pk_ptr, "msg", &request_type);

	/* Show a trace */
	if(op_prg_odb_ltrace_active("PACKET"))
	{
		diameter_packet_trace(pk_ptr);
	}

	/* Update the message on the packet based on the request type */
	switch (request_type)
	{
		case (DIAMETERC_MULTIMEDIA_AUTH_REQUEST):
		{
			diameter_UAS_Register_req_process(pk_ptr);
			break;
		}
		case (DIAMETERC_SERVER_ASSIGNMENT_REQUEST):
		{
			diameter_UAS_Register_res_req_process(pk_ptr);
			break;
		}

		default:
		{
			/* Print an error message */
```

```
                        sprintf (msg, "Unknown Request Type received by UAS (PID %d)",
my_pro_id);
                        op_prg_odb_print_major (msg, OPC_NIL);
                        break;
                        }
                }

        FOUT;
        }/* End diameter_UAS_req_process () */


/***** Transport Layer Related Functions *****/
static void diameter_UAS_conn_open_passive (char* protocol, OmsT_Qm_Tos tos)
 {
        Objid                                   my_node_objid;
        Objid                                   my_module_objid;
        DIAMETERT_Session*                      diameter_session_ptr;
        char                                    msg [128];
        Ici*                                    session_ici_ptr;
        DIAMETERT_Session_Ici_Map*              session_ici_map_ptr;


        /* This function opens a passive transport connection */
        /* to which UACs can connect to communicate with UAS  */
        FIN (diameter_UAS_conn_open_passive ());

        if (ltrace_diameter_UAS_active || ltrace_diameter_UAS_conn_active)
        {
                sprintf (msg, "UAS (PID %d) opening a passive connection on port (%d)", my_pro_id,
                DIAMETERC_UAS_LISTENING_PORT);
                op_prg_odb_print_major (msg, OPC_NIL);
        }

        /* Get the objid of the node on which this UAS runs */
        my_module_objid = op_id_self ();
        my_node_objid = op_topo_parent (my_module_objid);

        /* Create the session information */
        diameter_session_ptr = (DIAMETERT_Session *) op_prg_mem_alloc (sizeof
        (DIAMETERT_Session));
        diameter_session_ptr->diameter_prohndl = op_pro_self();
        diameter_session_ptr->session_index = next_sess_index;
        DIAMETERC_UAS_LISTENING_PORT, protocol);

        /* Open a passive TCP connection */
        session_ici_ptr = tpal_session_open_passive_v2 (my_node_objid,
        DIAMETERC_UAS_LISTENING_PORT, protocol, DIAMETER_UAS_SERVICE_NAME,
        diameter_session_ptr, OPC_NIL, tos, APPL_TYPE_SIP);
        /* Create a structure to store this transport session in a list which maps   */
        /* destination addresses to transport sessions.*/
        session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*)
        op_prg_mem_alloc(sizeof(DIAMETERT_Session_Ici_Map));
        session_ici_map_ptr->session_ici_ptr = session_ici_ptr;
        session_ici_map_ptr->session_index = next_sess_index;

        /* Save it in the mappings list */
        op_prg_list_insert (sessions_ici_map_lptr, session_ici_map_ptr, OPC_LISTPOS_TAIL);
```

```
	/* Increment the next session index so that the next session gets a unique identifier */
	next_sess_index++;

	FOUT;
	}/* End diameter_UAS_conn_open_passive () */

static void diameter_UAS_conn_open_active (char* dest_tpal_addr, int dest_port, char* protocol, OmsT_Qm_Tos tos)
{
	Objid                                   my_module_objid;
	Objid                                   my_node_objid;
	DIAMETERT_Session*                      diameter_session_ptr;
	char                                    msg [128];
	Ici*                                    session_ici_ptr;
	DIAMETERT_Session_Ici_Map*              session_ici_map_ptr;

	/* This function opens an active connection to the UAS */
	FIN(diameter_UAS_conn_open_active ());

	if (ltrace_diameter_UAS_active || ltrace_diameter_UAS_conn_active)
	{
		sprintf (msg, "UAS (PID %d) opening an active (%s) connection to dest (%s) on port (%d)",
		my_pro_id, protocol, dest_tpal_addr, dest_port);
		op_prg_odb_print_major (msg, OPC_NIL);
	}

	/* Get the objid of the node on which this UAC exists */
	my_module_objid = op_id_self ();
	my_node_objid = op_topo_parent (my_module_objid);

	/* Create the session information */
	diameter_session_ptr = (DIAMETERT_Session *) op_prg_mem_alloc (sizeof (DIAMETERT_Session));
	diameter_session_ptr->diameter_prohndl = op_pro_self();
	diameter_session_ptr->session_index = next_sess_index;

	/* Open a connection */
	session_ici_ptr = tpal_session_open_active_v2 (my_node_objid, dest_tpal_addr, dest_port, protocol, DIAMETER_UAS_SERVICE_NAME , diameter_session_ptr, OPC_NIL, tos, APPL_TYPE_SIP);

	/* Create a structure to store this transport session in a list which maps   */
	/* destination addresses to transport sessions.                              */
	session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*) op_prg_mem_alloc(sizeof(DIAMETERT_Session_Ici_Map));
	session_ici_map_ptr->session_ici_ptr = session_ici_ptr;
	session_ici_map_ptr->dest_tpal_addr = dest_tpal_addr;
	session_ici_map_ptr->session_index = next_sess_index;

	/* Save it in the mappings list */
	op_prg_list_insert (sessions_ici_map_lptr, session_ici_map_ptr, OPC_LISTPOS_TAIL);

	/* Increment the next session index so that the next session gets a unique identifier */
	next_sess_index++;
```

```
	FOUT;
	}/* End diameter_UAS_conn_open_active ()*/

static void diameter_UAS_conn_close (int    session_index)
{
	Objid                   my_module_objid;
	Objid                   my_node_objid;
	Ici*                    session_ici_ptr;

	/* This function closes the transport connection */
	FIN (diameter_UAS_conn_close (session_index));

	/* Get the session ici - whose position in the sesions list */
	/* will always correspond to the session index          */
	session_ici_ptr = diameter_UAS_session_ici_from_index_get (session_index);

	/* Get the objid of the node on which this UAC runs */
	my_module_objid = op_id_self ();
	my_node_objid = op_topo_parent (my_module_objid);

	/* Request tpal to close the connection */
	op_ici_install(session_ici_ptr);
	tpal_session_close (my_node_objid);

	FOUT;
}/* End diameter_UAS_conn_close () */

/***** Manager Feedback Related Functions *****/
static void diameter_UAS_mgr_inform_open ()
 {

	/* This function informs the diameter_UAS mgr that the      */
	/* active transport connection has been established */
	FIN (diameter_UAS_mgr_inform_open ());

	/* Schedule a forced interrupt for the UAS_mgr to inform that */
	/* an active connection has been established.                 */
	op_intrpt_schedule_process (my_mgr_prohandle, op_sim_time (),
	DIAMETERC_UAS_ACTIVATED);

	/* Set the ODB tag to "active" since this UAS is connected to a UAC */
	op_pro_tag_set (my_prohandle, "active");

	FOUT;
	}/* End diameter_UAS_mgr_inform_open () */

/**** Utility Functions ****/
static void diameter_UAS_ltrace_activate ()
{
	/* This function activates the ODB traces */
	FIN (diameter_UAS_ltrace_activate ());

	/* Set the full trace */
	ltrace_diameter_UAS_active = op_prg_odb_ltrace_active ("diameter");

	/* Set the error trace */
	ltrace_diameter_UAS_errors_active = op_prg_odb_ltrace_active ("diameter_error");
```

```
	/* Set the INVITE trace */
	ltrace_diameter_UAS_invite_active = op_prg_odb_ltrace_active ("diameter_invite");

	/* Set the BYE trace */
	ltrace_diameter_UAS_bye_active = op_prg_odb_ltrace_active ("diameter_bye");

	/* Set the all REQUESTS trace */
	ltrace_diameter_UAS_req_active = op_prg_odb_ltrace_active ("diameter_req");

	/* Set the all RESPONSES trace */
	ltrace_diameter_UAS_resp_active = op_prg_odb_ltrace_active ("diameter_resp");

	/* Set the all RESPONSES trace */
	ltrace_diameter_UAS_conn_active = op_prg_odb_ltrace_active ("diameter_conn");

	FOUT;
}/* End diameter_UAS_ltrace_activate ()*/

static Ici* diameter_UAS_session_ici_from_index_get (int session_index)
{
	DIAMETERT_Session_Ici_Map*	session_ici_map_ptr = OPC_NIL;
	int				num_sessions;
	int				ith_session;

	/* This function searches the sessions_ici_map_lptr to find   */
	/* an existing transport session to the specified destination */
	FIN (diameter_UAS_session_ici_from_index_get (session_index));

	/* Get the number of currently active sessions */
	num_sessions = op_prg_list_size (sessions_ici_map_lptr);

	/* Loop through to find the session which matches the */
	/* destination of interest                            */
	for (ith_session = 0; ith_session < num_sessions; ith_session++)
	{
		/* Get the ith active session */
		session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*) op_prg_list_access
		(sessions_ici_map_lptr, ith_session);

		/* Check if this session has the destination of interest */
		if (session_ici_map_ptr->session_index == session_index)
		FRET (session_ici_map_ptr->session_ici_ptr);
	}

	FRET (OPC_NIL);
}/* End diameter_UAS_session_ici_from_index_get () */

static Ici* diameter_UAS_session_ici_from_addr_get (char*  dest_tpal_addr)
{
	DIAMETERT_Session_Ici_Map*	session_ici_map_ptr = OPC_NIL;
	int				num_sessions;
	int				ith_session;

	/* This function searches the sessions_ici_map_lptr to find   */
	/* an existing transport session to the specified destination */
	FIN (diameter_UAS_session_ici_from_addr_get (dest_tpal_addr));
```

```
	/* Get the number of currently active sessions */
	num_sessions = op_prg_list_size (sessions_ici_map_lptr);

	/* Loop through to find the session which matches the    */
	/* destination of interest - the 0th session is not      */
	/* considered because it it the session to the call host */
	for (ith_session = 1; ith_session < num_sessions; ith_session++)
	{
		/* Get the ith active session */
		session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*) op_prg_list_access
		(sessions_ici_map_lptr, ith_session);

		/* Check if this session has the destination of interest */
		if (strcmp(session_ici_map_ptr->dest_tpal_addr,dest_tpal_addr) == 0)
		FRET (session_ici_map_ptr->session_ici_ptr);
	}
	FRET (OPC_NIL);
}/* End diameter_UAS_session_ici_from_addr_get () */

static void diameter_UAS_session_map_destroy (int  session_index)
{
	DIAMETERT_Session_Ici_Map*	session_ici_map_ptr = OPC_NIL;
	int				num_sessions;
	int				ith_session;

	/* This function searches the sessions_ici_map_lptr to remove */
	/* the transport session with the given session index. This   */
	/* function is called atthe time the session is closed.       */
	FIN (diameter_UAS_session_map_destroy (dest_tpal_addr));

	/* Get the number of currently active sessions */
	num_sessions = op_prg_list_size (sessions_ici_map_lptr);

	/* Loop through to find the session which matches the */
	/* destination of interest                            */
	for (ith_session = 0; ith_session < num_sessions; ith_session++)
	{
		/* Get the ith active session */
		session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*) op_prg_list_access
		(sessions_ici_map_lptr, ith_session);

		/* Check if this session has the destination of interest */
		if (session_index == session_ici_map_ptr->session_index)
		{
			/* Remove the session from the list */
			session_ici_map_ptr = (DIAMETERT_Session_Ici_Map*) op_prg_list_remove
			(sessions_ici_map_lptr, ith_session);

			/* Free its memory */
			op_prg_mem_free (session_ici_map_ptr);

			FOUT;
			}
		}
	FOUT;
}/* End diameter_UAS_session_map_destroy () */
```

```
static void diameter_UAS_mem_free ()
{
	/* This function frees all the memory associated with  */
	/* this process and is called just before this process */
	/* destroys itself                                     */
	FIN (diameter_UAS_mem_free ());

	/* Free the list of packets waiting to be relayed   */
	/* This list should be empty at this time           */
	if (op_prg_list_size (relay_packets_lptr) > 0)
		op_prg_odb_print_major ("DIAMETER UAS is being destroyed while there are pending
RELAY PACKETS", OPC_NIL);
	else
		op_prg_mem_free (relay_packets_lptr);

	/* Free the list of transport sessions to diffeernt UACs   */
	/* This list should be empty at this time                  */
	if (op_prg_list_size (sessions_ici_map_lptr) > 0)
		op_prg_odb_print_major ("DIAMETER UAS is being destroyed while there are
ACTIVE sessions", OPC_NIL);
	else
		op_prg_mem_free (sessions_ici_map_lptr);


	FOUT;
}/* End diameter_UAS_mem_free () */

static void diameter_get_proxy(const char* proxy_type, const char* domain, const char* area, char *addr)
{
	int		count;
	int		i;
	int		number;
	int		this_one;
	int		matches;


	/* This function returns the address */
	/* of a particular diameter proxy server */
	FIN (diameter_get_proxy ());

	/* Multiple elements could satisfy the search */
	/* Randomly choose an element to use */
	number = diameter_count_proxy(proxy_type, domain, area, addr);
	this_one = (int)op_dist_uniform(number);

	/* Loop through every node */
	count = op_topo_object_count (OPC_OBJTYPE_NODE_FIX);
	for (i = 0, matches = 0; i < count; i++)
	{
		Objid		node;

		node = op_topo_object (OPC_OBJTYPE_NODE_FIX, i);
		if (op_ima_obj_attr_exists (node, "Server Address"))
		{
			char		temp_string [128];
			Objid		diameter_UAS;
```

```
                        Objid                   diameter_UAS_row;
                        char                    node_domain[128];
                        char                    node_area[128];
                        char                    node_proxy_type[128];

                        /* Get the address and check if it is the right proxy */
                        op_ima_obj_attr_get (node, "Server Address", temp_string);
                        strcpy(addr, temp_string);

                        /* Get the attribute family */
                        op_ima_obj_attr_get (node, "DIAMETER Proxy Server Parameters",
                        &diameter_UAS);

                        /* Get the only row */
                        diameter_UAS_row = op_topo_child (diameter_UAS,
                        OPC_OBJTYPE_GENERIC, 0);

                        /* Get the domain name */
                        op_ima_obj_attr_get (diameter_UAS_row, "Domain Name", node_domain);

                        /* Get the area name */
                        op_ima_obj_attr_get (diameter_UAS_row, "Area Name", node_area);

                        /* If no area name is specified, ignore it */
                        if (strcmp (area, "") == 0)
                        {
                                strcpy (node_area, area);
                        }

                        /* Get the proxy type */
                        op_ima_obj_attr_get (diameter_UAS_row, "Proxy Type", node_proxy_type);
                        if (strcmp (domain, node_domain) == 0 &&strcmp (area, node_area) == 0 &&
                        strcmp (proxy_type, node_proxy_type) == 0)
                        {
                                if (matches == this_one)
                                        {
                                        break;
                                        }
                                else
                                        {
                                        matches++;
                                        }
                        }

                        /* No matching */
                        strcpy (addr, "");
                        }
                }
        FOUT;
}

static int diameter_count_proxy(const char* proxy_type, const char* domain, const char* area, char *addr)
{
        int                                             count;
        int                                             i;
        int                                             n = 0;
```

```
/* This function returns the number of addresses */
/* that match a particular diameter proxy server type */
FIN (diameter_count_proxy ());

/* Loop through every node, starting at a random one */
count = op_topo_object_count (OPC_OBJTYPE_NODE_FIX);
for (i = 0; i < count; i++)
{
	Objid	node;

	node = op_topo_object (OPC_OBJTYPE_NODE_FIX, i);
	if (op_ima_obj_attr_exists (node, "Server Address"))
	{
		char	temp_string [128];
		Objid	diameter_UAS;
		Objid	diameter_UAS_row;
		char	node_domain[128];
		char	node_area[128];
		char	node_proxy_type[128];

		/* Get the address and check if it is the right proxy */
		op_ima_obj_attr_get (node, "Server Address", temp_string);
		strcpy(addr, temp_string);

		/* Get the attribute family */
		op_ima_obj_attr_get (node, "DIAMETER Proxy Server Parameters",
		&diameter_UAS);

		/* Get the only row */
		diameter_UAS_row = op_topo_child (diameter_UAS,
		OPC_OBJTYPE_GENERIC, 0);

		/* Get the domain name */
		op_ima_obj_attr_get (diameter_UAS_row, "Domain Name", node_domain);

		/* Get the area name */
		op_ima_obj_attr_get (diameter_UAS_row, "Area Name", node_area);

		/* If no area name is specified, ignore it */
		if (strcmp (area, "") == 0)
		{
			strcpy (node_area, area);
		}

		/* Get the proxy type */
		op_ima_obj_attr_get (diameter_UAS_row, "Proxy Type", node_proxy_type);
		if (strcmp (domain, node_domain) == 0 && strcmp (area, node_area) == 0 &&
		strcmp (proxy_type, node_proxy_type) == 0)
		{
			n++;
		}

		/* No matching */
		strcpy (addr, "");
		}
	}
```

```
	FRET(n);
}

static void diameter_get_destination_domain_and_area(const char* invitee_addr, char* domain_name, char* current_domain, char* current_area)
{
	int		count;
	int		i;

	/* This function returns the domain */
	/* of a particular client */
	FIN (diameter_get_destination_domain ());

	/* Loops through every node */
	count = op_topo_object_count (OPC_OBJTYPE_PROC);
	for (i = 0; i < count; i++)
	{
		Objid		process;

		process = op_topo_object (OPC_OBJTYPE_PROC, i);
		if (op_ima_obj_attr_exists (process, "Address"))
		{
			char		temp_string [128];

			/* Get the address and check it */
			op_ima_obj_attr_get (process, "Address", temp_string);

			if (strcmp (temp_string, invitee_addr) == 0)
			{
				Objid		node;
				Objid		diameter_UAC;
				Objid		diameter_UAC_row;

				/* We found the desired process */

				/* Get the node */
				node = op_topo_parent (process);

				/* Get the attribute family */
				op_ima_obj_attr_get (node, "DIAMETER UAC Parameters", &diameter_UAC);

				/* Get the only row */
				diameter_UAC_row = op_topo_child (diameter_UAC, OPC_OBJTYPE_GENERIC, 0);

				/* Get the domain name */
				op_ima_obj_attr_get (diameter_UAC_row, "Domain Name", domain_name);

				/* Get the current domain */
				op_ima_obj_attr_get (diameter_UAC_row, "Current Domain", current_domain);

				/* Get the current area */
				op_ima_obj_attr_get (diameter_UAC_row, "Current Area", current_area);
```

```
                                        break;
                                        }
                                }
                        }

        FOUT;
}

static void diameter_packet_trace (Packet* pk_ptr)
{
        int                                     request_type;
        char                                    tipo [128];

        /* This function shows a trace related to a packet */
        FIN (diameter_packet_trace ());

        /* Get the request type from the packet */
        op_pk_nfd_get (pk_ptr, "msg", &request_type);

        if (request_type == DIAMETERC_MULTIMEDIA_AUTH_REQUEST)
        {
                strcpy (tipo, "Multimedia Auth Request");
        }
        else if (request_type == DIAMETERC_SERVER_ASSIGNMENT_REQUEST)
        {
                strcpy (tipo, "Server Assignment Request");
        }
        else
        {
                strcpy (tipo, "Invalid");
        }

        printf ("%f - %s (%s, %s, %s): DIAMETER %s packet received\n", op_sim_time (), my_name,
UAS_ptc_mem_ptr->proxy_type_ptr,      UAS_ptc_mem_ptr->domain_name_ptr,      UAS_ptc_mem_ptr-
>area_name_ptr, tipo);

        FOUT;
}

static OmsT_Qm_Tos diameter_UAS_tos_get ()
{
        /* This function specifies the type of service */
        /* to be used for diameter signalling */
        FIN (diameter_UAS_tos_get ());

        FRET (OmsC_Qm_Tos_Reserved);
        } /* End diameter_UAS_tos_get () */

static void diameter_UAS_Register_req_process (Packet* pk_ptr)
{
        DIAMETERT_Call_Info_Shell*              diameter_call_info_shell_ptr = OPC_NIL;
        DIAMETERT_Call_Info*                    diameter_call_info_ptr = OPC_NIL;

        /* This function processes the register request */
        /* It adjusts the message on the packet and   */
        /* time stamps the associated ICI          */
```

```
	FIN (diameter_UAS_register_req_process ());

	/* Check if there are open lines available to process this request */
	/* Set the packet msg_index field to indicate REGISTER */
	op_pk_nfd_set (pk_ptr, "msg", DIAMETERC_MULTIMEDIA_AUTH_REQUEST);

	/* Set packet type field */
	op_pk_nfd_set (pk_ptr, "type", DIAMETERC_Packet_Type_Request);

	/* Get the call information from the ICI */
	op_pk_nfd_access (pk_ptr, "call_info", &diameter_call_info_shell_ptr);
	diameter_call_info_ptr = diameter_call_info_shell_ptr->diameter_call_info_ptr;

	/* Save the call handle (shell) for reference */
	call_info_shell_ptr = diameter_call_info_shell_ptr;

	/* Decrement the open lines counter */
	(*(UAS_ptc_mem_ptr->open_lines_count_ptr))--;

	/* If this node is a HSS, process the registration request*/
	if (strcmp(auth_mode,"3GPP")==0)
	{
		diameter_UAS_ack_reg_unauth(pk_ptr);
	}
	else
	{
		diameter_UAS_ack_reg_OK(pk_ptr);
	}
	FOUT;
} /* End diameter_UAS_Register_req_process ()*/

static void diameter_UAS_ack_reg_unauth (Packet* pk_ptr)
{
	Ici*		session_ici_ptr;

	/* This function processes the ACK of registration unauthorised */
	FIN (diameter_UAS_ack_reg_unauth ());

	/* Set packet type field */
	op_pk_nfd_set (pk_ptr, "type", DIAMETERC_Packet_Type_ANSWER);

	/* Set packet msg field*/
	op_pk_nfd_set (pk_ptr, "msg", DIAMETERC_MULTIMEDIA_AUTH_ANSWER);


	/*********INVOKE the 3GPP AKA PROTOCOL**********************/

	diameter_AKA_AV_gen (pk_ptr);

	/* Get the session ici - whose position in the sesions list */
	/* will always be 0 for the UAC that started this call	*/
	session_ici_ptr = diameter_UAS_session_ici_from_index_get (0);

	/* Install the session ici */
	op_ici_install (session_ici_ptr);

	/* Send the packet back to the UAC */
```

```
	op_pk_send (pk_ptr, strm_to_tpal);

	FOUT;
}/* End diameter_UAS_ack_reg_unauth ()*/

static void diameter_UAS_ack_reg_OK (Packet* pk_ptr)
{
	Ici*	session_ici_ptr;

	/* This function processes the ACK of registration OK */
	FIN (diameter_UAS_ack_reg_OK ());

	/* Set packet type field */
	op_pk_nfd_set (pk_ptr, "type", DIAMETERC_Packet_Type_ANSWER);

	/* Set packet msg field*/
	op_pk_nfd_set (pk_ptr, "msg", DIAMETERC_SERVER_ASSIGNMENT_ANSWER);

	/* Get the session ici - whose position in the sesions list */
	/* will always be 0 for the UAC that started this call	*/
	session_ici_ptr = diameter_UAS_session_ici_from_index_get (0);

	/* Install the session ici */
	op_ici_install (session_ici_ptr);

	/* Send the packet back to the UAC */
	op_pk_send (pk_ptr, strm_to_tpal);

	FOUT;
}/* End diameter_UAS_ack_reg_OK ()*/

static void diameter_UAS_Register_res_req_process (Packet* pk_ptr)
{
	DIAMETERT_Call_Info_Shell*		diameter_call_info_shell_ptr = OPC_NIL;
	DIAMETERT_Call_Info*			diameter_call_info_ptr = OPC_NIL;

	/* This function processes the register request */
	/* It adjusts the message on the packet and	*/
	/* time stamps the associated ICI		*/
	FIN (diameter_UAS_Register_res_req_process ());

	/* Check if there are open lines available to process this request */
	/* Set the packet msg_index field to indicate REGISTER */
	op_pk_nfd_set (pk_ptr, "msg", DIAMETERC_SERVER_ASSIGNMENT_REQUEST);

	/* Set packet type field */
	op_pk_nfd_set (pk_ptr, "type", DIAMETERC_Packet_Type_Request);

	/* Get the call information from the ICI */
	op_pk_nfd_access (pk_ptr, "call_info", &diameter_call_info_shell_ptr);
	diameter_call_info_ptr = diameter_call_info_shell_ptr->diameter_call_info_ptr;

	/* Save the call handle (shell) for reference */
	call_info_shell_ptr = diameter_call_info_shell_ptr;

	/* Decrement the open lines counter */
	(*(UAS_ptc_mem_ptr->open_lines_count_ptr))--;
```

```
        diameter_UAS_ack_reg_OK(pk_ptr);

        /* Write the Call Setup Request statistics */
        op_stat_write (UAS_ptc_mem_ptr->call_setup_requests_stathandle, 1);

        FOUT;

} /* diameter_UAS_Register_res_req_process ()*/
```

A.4: Source Code for AKA authentication

```
typedef unsigned char   u1byte; /* an 8 bit unsigned character type */
typedef unsigned short  u2byte; /* a 16 bit unsigned integer type   */
typedef unsigned long   u4byte; /* a 32 bit unsigned integer type   */

typedef signed char     s1byte; /* an 8 bit signed character type   */
typedef signed short    s2byte; /* a 16 bit signed integer type     */
typedef signed long     s4byte; /* a 32 bit signed integer type     */

/*3GPP AKA*/
static void sip_AKA_AV_gen (Packet* pk_ptr)
{
        SIPT_Call_Info_Shell*           sip_call_info_shell_ptr = OPC_NIL;
        SIPT_Call_Info*                 sip_call_info_ptr = OPC_NIL;

        u1byte var_rand[16];
        u1byte var_res[8],cli_XRES[8];
        u1byte var_ck[16],cli_ck[16];
        u1byte var_ik[16],cli_ik[16];
        u1byte var_ak[6];
        u1byte var_autn[16];
        u1byte cli_SQN[6];
        u1byte cli_AMF[2];
        u1byte cli_MAC[8],cli_XMAC[8];
        char show_string[500];

        /* This function processes the generation of AKA Authenticaion Vectors */
        FIN (sip_AKA_AV_gen ());

        /*initialize the variables*/
        memset(var_rand,0,sizeof(var_rand));
        memset(var_res,0,sizeof(var_res));
        memset(var_ck,0,sizeof(var_ck));
        memset(var_ik,0,sizeof(var_ik));
        memset(var_ak,0,sizeof(var_ak));
        memset(var_autn,0,sizeof(var_autn));
        memset(show_string,0,sizeof(show_string));

        memset(cli_XRES,0,sizeof(cli_XRES));
        memset(cli_ck,0,sizeof(cli_ck));
        memset(cli_ik,0,sizeof(cli_ik));
        memset(cli_SQN,0,sizeof(cli_SQN));
        memset(cli_AMF,0,sizeof(cli_AMF));
        memset(cli_MAC,0,sizeof(cli_MAC));
```

```
memset(cli_XMAC,0,sizeof(cli_XMAC));

/*GET CALL_INFO FROM THE PACKET*/
op_pk_nfd_access (pk_ptr, "call_info", &sip_call_info_shell_ptr);
sip_call_info_ptr=sip_call_info_shell_ptr->sip_call_info_ptr;

/*get the random number*/
strcpy(show_string,sip_call_info_ptr->RAND);
sip_gen_byte(show_string,32,var_rand);

/*get CK*/
strcpy(show_string,sip_call_info_ptr->CK);
sip_gen_byte(show_string,32,var_ck);

/*get IK*/
strcpy(show_string,sip_call_info_ptr->IK);
sip_gen_byte(show_string,32,var_ik);

/*get AUTN*/
strcpy(show_string,sip_call_info_ptr->AUTN);
sip_gen_byte(show_string,32,var_autn);

/*generate AK*/
sip_f5_AK(u1_KEY,var_rand,u1_OP,var_ak);
sip_gen_string(var_ak,6,show_string);
if(op_prg_odb_ltrace_active("AUTH"))
{
        printf("\tAK:%s\n\n",show_string);
}

/*********from client side*****************/
/*generate SQN AMF and MAC*/
sip_gen_SQN_AMF_MAC(var_autn,var_ak,cli_SQN,cli_AMF,cli_MAC);
sip_f1_MACA(u1_KEY,var_rand,cli_SQN,cli_AMF,u1_OP,cli_XMAC);
sip_gen_string(cli_XMAC,8,show_string);
if(op_prg_odb_ltrace_active("AUTH"))
{
        printf("\tCLIENT MAC:%s\n\n",show_string);
}

/*generate RES*/
sip_f2_RES(u1_KEY,var_rand,u1_OP,cli_XRES);
sip_gen_string(cli_XRES,8,show_string);
strcpy(sip_call_info_ptr->RES,show_string);
if(op_prg_odb_ltrace_active("AUTH"))
{
        printf("\tCLIENT RES:%s\n\n",sip_call_info_ptr->RES);
}

sip_f3_CK(u1_KEY,var_rand,u1_OP,cli_ck);
sip_gen_string(cli_ck,16,show_string);
if(op_prg_odb_ltrace_active("AUTH"))
{
        printf("\tCLIENT CK:%s\n\n",show_string);
}

sip_f4_IK(u1_KEY,var_rand,u1_OP,cli_ik);
```

```
	sip_gen_string(cli_ik,16,show_string);
	/*Save the new SIP_CALL_INFO to the packet*/
	op_pk_nfd_set (pk_ptr, "call_info", sip_call_info_shell_ptr, sip_call_info_fdstruct_copy_proc,
sip_call_info_fdstruct_destroy_proc, 0);

	FOUT;
}/* End sip_AKA_AV_gen ()*/

static void sip_gen_rand(u1byte out_rand[16])
{

	int		new_seed, rand_int;
	PrgT_Random_Gen		*my_rng;
	int					i=0;

	/* This function processes the generation of random number*/
	FIN (sip_gen_rand ());

	/* create a new random number generator */
	new_seed=(int)op_sim_time();
	my_rng = op_prg_random_gen_create (new_seed);

	for (i=0;i<16;i++)
	{
		/* generate a random integer in the interval [0,255] */
		rand_int = (op_prg_random_integer_gen (my_rng) % 256);
		out_rand[i]=rand_int;
	}

	/* destroy the random number generator */
	op_prg_random_gen_destroy (my_rng);

	FOUT;

}/* End sip_gen_rand ()*/

static void sip_gen_ini(const u1byte in_SQNhe[6], const u1byte in_AMF[2], u1byte out_ini[16])
{

	int i=0;
	FIN(sip_gen_ini());

	for(i=0;i<=5;i++)
	{
		out_ini[i]=in_SQNhe[i];
	}

	out_ini[6]=in_AMF[0];
	out_ini[7]=in_AMF[1];

	i=i+2;
	for(i=8;i<14;i++)
	{
		 out_ini[i]=in_SQNhe[i-8];
	}

	out_ini[14]=in_AMF[0];
```

```
	out_ini[15]=in_AMF[1];

	FOUT;
}/*end of sip_gen_ini */

static void sip_gen_temp(const u1byte in_rand[16],const u1byte in_opc[16],const u1byte in_key[16],u1byte out_temp[16])
{
	u1byte i_temp[16];
	int i=0;

	FIN(sip_gen_temp());

	for(i=0;i<=15;i++)
	{
		i_temp[i]=in_rand[i]^in_opc[i];
	}

	sip_encrypt(in_key,128,i_temp,out_temp);

	 FOUT;
}/*end of sip_gen_temp()*/

static void sip_rotate(const u1byte in_var[16],int r,u1byte out_var[16])
{
	u1byte temp_var1[16];
	u1byte temp_var2[16];
	int byte_num=0,bit_num=0;
	u1byte s_part=0,a=0,b=0,c=0;
	int i=0;

	FIN(sip_rotate());

	for(i=0;i<sizeof(temp_var1);i++)
	{
		 temp_var1[i]=0;
		 temp_var2[i]=0;
	}

	byte_num=r/8;
	 bit_num=r%8;

	 for(i=0;i<16-byte_num;i++)
	{
		temp_var1[i]=in_var[i+byte_num];
	}

	 for(i=0;i<byte_num;i++)
	{
		 temp_var1[i+16-byte_num]=in_var[i];
	}

	if (bit_num>0)
	{
		for(i=0;i<sizeof(temp_var1);i++)
		{
			if(i==0)
```

```
                        {
                                s_part=in_var[i]>>(8-bit_num);
                                a=temp_var1[i]<<bit_num;
                                b=temp_var1[i+1]>>(8-bit_num);
                                c=a|b;
                                temp_var2[i]=c;
                        }
                        else if(i==sizeof(temp_var1)-1)
                        {
                                a=temp_var1[i]<<bit_num;
                                b=s_part;
                                c=a|b;
                                temp_var2[i]=c;
                        }
                        else
                        {
                                a=temp_var1[i]<<bit_num;
                                b=temp_var1[i+1]>>(8-bit_num);
                                c=a|b;
                                temp_var2[i]=c;
                        }
                }
        }
        else
        {
                for(i=0;i<sizeof(temp_var1);i++)
                {
                        temp_var2[i]=temp_var1[i];
                }
        }

        for(i=0;i<sizeof(temp_var1);i++)
        {
                out_var[i]=temp_var2[i];
        }
        FOUT;
}

static void sip_XOR_16(const u1byte in_var1[16], const u1byte in_var2[16],u1byte out_var3[16])
{
        int i=0;

        FIN(sip_XOR_16());

        for (i=0;i<=15;i++)
        {
                out_var3[i]=in_var1[i]^in_var2[i];
        }
        FOUT;
 }/*end of sip_XOR_16*/

static void sip_XOR_6(const u1byte in_var1[6], const u1byte in_var2[6],u1byte out_var3[6])
{
        int i=0;

        FIN(sip_XOR_6());
        for (i=0;i<=5;i++)
```

```
	{
		out_var3[i]=in_var1[i]^in_var2[i];
	}
	FOUT;
}/*end of sip_XOR_6*/

static void sip_f1_MACA(const u1byte in_key[16],const u1byte in_rand[16],const u1byte in_SQN[6],const u1byte in_AMF[2],u1byte in_op[16],u1byte out_MACA[8])
{
	u1byte var_eop[16];
	u1byte var_opc[16];
	u1byte var_ini[16];
	u1byte var_iniopc[16];
	u1byte var_rotiniopcr1[16];
	u1byte var_temp[16];
	u1byte var_temprot[16];
	u1byte var_temprotc1[16];
	u1byte var_etempopc[16];
	u1byte var_out1[16];
	int i=0;

	FIN(sip_f1_MACA());

	sip_encrypt(in_key,128,in_op,var_eop);
	sip_XOR_16(var_eop,in_op,var_opc);
	sip_gen_ini(in_SQN,in_AMF,var_ini);
	sip_XOR_16(var_ini,var_opc,var_iniopc);
	sip_rotate(var_iniopc,r1,var_rotiniopcr1);
	sip_gen_temp(in_rand,var_opc,in_key,var_temp);
	sip_XOR_16(var_temp,var_rotiniopcr1,var_temprot);
	sip_XOR_16(var_temprot,c1,var_temprotc1);
	sip_encrypt(in_key,128,var_temprotc1,var_etempopc);
	sip_XOR_16(var_etempopc,var_opc,var_out1);

	for(i=0;i<8;i++)
	{
		out_MACA[i]=var_out1[i];
	}
	FOUT;
}/*end of sip_f1_MACA() */

static void sip_f2_RES(const u1byte in_key[16],const u1byte in_rand[16],const u1byte in_op[16],u1byte out_RES[8])
{
	u1byte var_eop[16];
	u1byte var_opc[16];
	u1byte var_temp[16];
	u1byte var_tempopc[16];
	u1byte var_rottempopc[16];
	u1byte var_tempopcc2[16];
	u1byte var_etempopc[16];
	u1byte var_out2[16];
	int i=0;

	FIN(sip_f2_RES());
	sip_encrypt(in_key,128,in_op,var_eop);
	sip_XOR_16(var_eop,in_op,var_opc);
```

```
	sip_gen_temp(in_rand,var_opc,in_key,var_temp);
	sip_XOR_16(var_temp,var_opc,var_tempopc);
	sip_rotate(var_tempopc,r2,var_rottempopc);
	sip_XOR_16(var_rottempopc,c2,var_tempopcc2);
	sip_encrypt(in_key,128,var_tempopcc2,var_etempopc);
	sip_XOR_16(var_etempopc,var_opc,var_out2);

	for(i=0;i<8;i++)
	{
		out_RES[i]=var_out2[i];
	}
	FOUT;
}/*end of sip_f2_RES()*/


static void sip_f3_CK(const u1byte in_key[16],const u1byte in_rand[16],const u1byte in_op[16],u1byte
out_CK[16])
{
	u1byte var_eop[16];
	u1byte var_opc[16];
	1byte var_temp[16];
	u1byte var_tempopc[16];
	u1byte var_rottempopc[16];
	u1byte var_tempopcc2[16];
	 u1byte var_etempopc[16];

	FIN(sip_f3_CK());
	sip_encrypt(in_key,128,in_op,var_eop);
	sip_XOR_16(var_eop,in_op,var_opc);
	sip_gen_temp(in_rand,var_opc,in_key,var_temp);
	sip_XOR_16(var_temp,var_opc,var_tempopc);
	sip_rotate(var_tempopc,r3,var_rottempopc);
	sip_XOR_16(var_rottempopc,c3,var_tempopcc2);
	sip_encrypt(in_key,128,var_tempopcc2,var_etempopc);
	sip_XOR_16(var_etempopc,var_opc,out_CK);

	FOUT;
}/*end of sip_f3_CK()*/

static void sip_f4_IK(const u1byte in_key[16],const u1byte in_rand[16],const u1byte in_op[16],u1byte
out_IK[16])
{
	u1byte var_eop[16];
	u1byte var_opc[16];
	u1byte var_temp[16];
	u1byte var_tempopc[16];
	 u1byte var_rottempopc[16];
	u1byte var_tempopcc2[16];
	u1byte var_etempopc[16];

	FIN(sip_f4_IK());

	sip_encrypt(in_key,128,in_op,var_eop);
	sip_XOR_16(var_eop,in_op,var_opc);
	sip_gen_temp(in_rand,var_opc,in_key,var_temp);
	sip_XOR_16(var_temp,var_opc,var_tempopc);
	sip_rotate(var_tempopc,r4,var_rottempopc);
```

```
	sip_XOR_16(var_rottempopc,c4,var_tempopcc2);
	sip_encrypt(in_key,128,var_tempopcc2,var_etempopc);
	sip_XOR_16(var_etempopc,var_opc,out_IK);

	FOUT;
}/*end of sip_f4_IK()*/

static void sip_f5_AK(const u1byte in_key[16],const u1byte in_rand[16],const u1byte in_op[16],u1byte
out_AK[6])
{
	u1byte var_eop[16];
	u1byte var_opc[16];
	u1byte var_temp[16];
	u1byte var_tempopc[16];
	u1byte var_rottempopc[16];
	u1byte var_tempopcc2[16];
	u1byte var_etempopc[16];
	u1byte var_out5[16];
	int i=0;

	FIN(sip_f5_AK);
	sip_encrypt(in_key,128,in_op,var_eop);
	sip_XOR_16(var_eop,in_op,var_opc);
	sip_gen_temp(in_rand,var_opc,in_key,var_temp);
	sip_XOR_16(var_temp,var_opc,var_tempopc);
	sip_rotate(var_tempopc,r5,var_rottempopc);
	sip_XOR_16(var_rottempopc,c5,var_tempopcc2);
	sip_encrypt(in_key,128,var_tempopcc2,var_etempopc);
	sip_XOR_16(var_etempopc,var_opc,var_out5);

	for(i=0;i<6;i++)
	{
		out_AK[i]=var_out5[i];
	}
 FOUT;
}/*end of sip_f5_AK*/

static void sip_gen_AUTN(const u1byte in_SQN[6],const u1byte in_AK[6],const u1byte in_AMF[2],const
u1byte in_MAC[8],u1byte out_AUTN[16])
{
	u1byte var_SQNAK[6];
	u1byte var_outemp[16];
	int i=0;

	FIN(sip_gen_AUTN());
	sip_XOR_6(in_SQN,in_AK,var_SQNAK);
	for(i=0;i<6;i++)
	{
		var_outemp[i]=var_SQNAK[i];
	}
	var_outemp[6]=in_AMF[0];
	var_outemp[7]=in_AMF[1];

	for(i=0;i<8;i++)
	{
		var_outemp[8+i]=in_MAC[i];
	}
```

```
	for (i=0;i<16;i++)
	{
		out_AUTN[i]=var_outemp[i];
	}
 FOUT;

}/*end of sip_gen_AUTN()*/

static void sip_gen_SQN_AMF_MAC(const u1byte in_AUTN[16],const u1byte in_AK[6],u1byte
out_SQN[6],u1byte out_AMF[2],u1byte out_MAC[8])
{
	u1byte var_SQNAK[6];
	int i=0;

	FIN(sip_gen_SQN_AMF_MAC());

	for (i=0;i<6;i++)
	{
		var_SQNAK[i]=in_AUTN[i];
	}
	out_AMF[0]=in_AUTN[6];
	out_AMF[1]=in_AUTN[7];

	for(i=0;i<8;i++)
	{
		out_MAC[i]=in_AUTN[i+8];
	}
	sip_XOR_6(var_SQNAK,in_AK,out_SQN);

	FOUT;
}/*end of sip_gen_SQN_AMF_MAC*/

static void sip_gen_string(const u1byte in_bytes[],int in_len, char *outstring)
{
	int i=0;
	char my_string[300];
	char tempLow,tempHigh;

	FIN(sip_gen_string());

	for(i=0;i<sizeof(my_string);i++)
	{
		my_string[i]=0;
	}

	for(i=0;i<in_len;i++)
	{
		tempLow=in_bytes[i]&15;
		tempHigh=(in_bytes[i]>>4)&15;
		if (tempLow>=0&&tempLow<=9)
		{
			tempLow=tempLow+48;
		}
		else
		{
			tempLow=tempLow+55;
```

```
		}

		if (tempHigh>=0&&tempHigh<=9)
		{
			tempHigh=tempHigh+48;
		}
		else
		{
			tempHigh=tempHigh+55;
		}
		my_string[i*2]=tempHigh;
		my_string[i*2+1]=tempLow;
	}

	strcpy(outstring,my_string);

	FOUT;
}/*end of sip_gen_string*/

static void sip_gen_byte(char *in_string,int in_len,u1byte out_bytes[])
{
	int i=0,j=0;
	char tempLow,tempHigh;


	FIN(sip_gen_byte());


	for(i=0;i<in_len;i=i+2)
	{
		tempHigh=in_string[i+0];
		tempLow=in_string[i+1];

		if ((tempHigh>=0x30)&&(tempHigh<=0x39))
		{
			tempHigh=tempHigh-0x30;
		}
		else
		{
			tempHigh=tempHigh-55;
		}
		tempHigh=tempHigh<<4;

		if ((tempLow>=0x30)&&(tempLow<=0x39))
		{
			tempLow=tempLow-0x30;
		}
		else
		{
			tempLow=tempLow-55;
		}
		out_bytes[j]=tempHigh|tempLow;
		j++;
	}

	FOUT;
}/*end of sip_gen_byte*/
```

Appendix B

2011 International Conference on Information and Networking (ICOIN 2011)
26-28 January, 2011, Kuala Lumpur, Malaysia.

IMPROVED ONE-PASS IP MULTIMEDIA SUBSYSTEM AUTHENTICATION FOR UMTS

Lili Gu, Mark A Gregory
School of Electrical and Computer Engineering.
† RMIT University
Melbourne, Australia.
l.gu@student.rmit.edu.au, mark.gregory@rmit.edu.au

Abstract—***As defined in the 3rd Generation Partnership Project specifications, a Universal Mobile Telecommunications System (UMTS) user device has to carry out two authentication steps to access multimedia services in IP Multimedia Subsystem (IMS). The first authentication step is used to gain UMTS General Packet Radio Service (GPRS) network admission. The second authentication step is the IMS authentication. The authentication steps utilize Authentication and Key Agreement (AKA) and many of the authentication steps are similar. This paper proposed a simplified authentication process known as Improved AKA (IAKA) which reduced the two authentication steps used to access the UMTS GPRS network and IMS services to one authentication step. Simulation results showed the IAKA protocol could significantly reduce the authentication delay and improve overall efficiency.***

Keywords: *Authentication; IP Multimedia Subsystem (IMS); Universal Mobile Telecommunications System (UMTS); Authentication and Key Agreement (AKA); Session Initiation Protocol (SIP); Call Session Control Function (CSCF).*

I. INTRODUCTION

IP Multimedia Subsystem (IMS) was first specified in 3rd Generation Partnership Project (3GPP) and the IMS provides multimedia services (i.e. audio, video, text, image, and combinations) over Packet Switched Networks (PSN) for multiple access networks [1], such as 3G/4G, DSL, and WLAN. The centralized Home Subscriber Server (HSS) makes it possible to deliver applications using single sign-in, unified charging policy and other features, for different handsets or clients over different access networks. However, the complexities of the service procedures raise performance concerns and affect the IMS deployment scenarios.

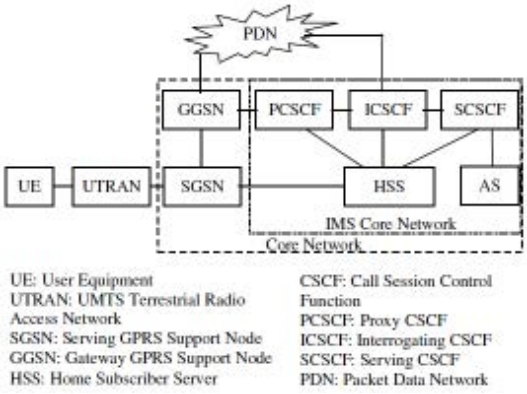

Figure1 IMS architecture for UMTS

As shown in Fig. 1, in order to access multimedia services, a user of Universal Mobile Telecommunications System (UMTS) has to pass two Authentication and Key Agreement (AKA) authentication steps [2] [3] which include the key material generation, distribution and the user's authentication. First, the user sends requests to the SGSN; the SGSN fetches the user's Authentication Vectors (AVs) from the HSS and enables the user to access UMTS and the packet data network. Secondly, through the Proxy CSCF (PCSCF) and the Interrogating CSCF (ICSCF), the User Equipment (UE) sends requests to the SCSCF and the SCSCF retrieves the user's AVs from the HSS and authenticates the user. After

the second authentication step the user can access IMS services. It was observed that the two-pass AKA authentication includes two executions of the AKA mechanism and many of the steps are similar.

In this paper, an Improved-AKA (IAKA) authentication mechanism is proposed to replace the original two-pass authentication process. In IAKA, AKA is executed once to generate authentication vectors by using the user's unique IP Multimedia Private-user Identity. The network entities use the authentication vector to do mutual authentication and enable the user to access the UMTS GPRS network and IMS services. The IAKA approach has the following features: (1) reduced authentication time, (2) reduced computation on the level of the HSS and terminal, (3) No compromising of the security level, and (4) minimal impact on the existing UMTS and IMS system.

The rest of paper is organized as follows: Section II is the related work; Section III describes the legacy two-pass AKA authentication; Section IV introduces the proposed IAKA authentication process; Section V includes simulation results that show the IAKA is more efficient than the legacy two-pass authentication process; Section VI includes an analysis of the security level; Section VII includes an analysis of the authentication process; VIII summarizes the features of the proposed IAKA authentication process and provides future work.

II. Related Work

Authentication is an important function carried out for user devices connecting to the network and network subsystems. Research is being carried out to reduce authentication complexity and improve the authentication process efficiency. However, a common limitation of research presented in the literature is that proposed methods require extensive modification to the network infrastructure or there is a reduction in security.

Lin et al. [4] proposed a simple one-pass authentication scheme that only needs to perform UMTS GPRS authentication. At the IMS level the CSCF stores a International Mobile Subscriber Identity (IMSI) and IP Multimedia Private-user Identity (IMPI) pair to do an implicit authentication without the duplicated AKA operations which may save up to 50% of the IMS authentication traffic. However, the one-pass authentication solution disclosed the user's private identity and is vulnerable to adversary attacks such as the fake attack, eavesdropping attack and others.

Huang and Li [5] proposed an Evolutionary IMS AKA (E-IMS AKA) to replace the IMS AKA which not only adheres to the security requirements of IMS AKA but also maintains the efficiency of Lin et al.'s one-pass authentication. However, E-IMS used a temporary key in the authentication and compromised security. Two years later, the authors [6] improved the E-IMS authentication procedures and presented a simpler and more flexible solution. However, it used the terminal to generate key information and distribute the key information to the authentication server which increased the system's complexity.

Nitantogian and Xenakis [7] proposed an improved authentication procedure for the 3G-WLAN integrated networks that enables a WLAN user to get access to the 3G packet switched services or to the public Internet through the 3G public land mobile network . The proposed one-pass EAP-AKA procedure combines the initial and the second authentication steps by making a security binding between them which significantly reduces the authentication overhead without compromising the security services when compared to the legacy two-pass authentication process. In 2010, based on [4] and [7], Nitantogian et al. [8] extended their research to the Beyond 3rd Generation (B3G) networks and proposed a generic mechanism for efficient authentication for B3G networks. However, on the IMS layer, the proposed authentication disclosed the user's identity and this is a security risk.

III. 3GPP two-pass authentication

In this section, the two-pass AKA authentication process including the UMTS GPRS AKA authentication and the IMS AKA authentication are described.

A. UMTS GPRS AKA Authentication

If the user wants to access GPRS services, the UE invokes an attach request which triggers the authentication procedure. The authentication procedure is shown in Figure 2 which is defined in 3GPP TS 33.102 [9].

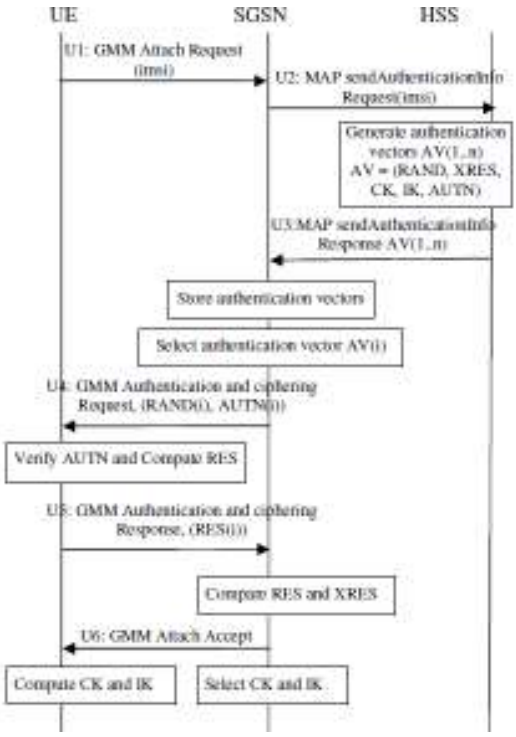

Figure2 UMTS GPRS AKA autentication procedure

The UMTS GPRS AKA authentication process includes:

1) U1: In order to access GPRS services, UE sends SGSN an GMM Attach Request with the parameter IMSI=imsi. The protocol between UE and SGSN is GMM (GPRS Mobility Management);

2) U2: if the SGSN does not have a previously stored authentication vector, it sends a sendAuthenticationInfo Request to HSS with IMSI=imsi. The protocol between SGSN and HSS is MAP (Mobile Application Part);

3) U3: Upon receipt the request, the HSS uses IMSI=imsi to retrieve the user's record and generate Authenticaion Vectors. One Authentication Vector includes RAND, XRES, CK (Cipher Key), IK (Integrity Key), AUTN. The AVs are ordered by sequence number; Then the HSS sends the ordered array of n Authentication vectors to the SGSN;

4) U4: Upon receipt the ordered AVs, the SGSN stores the AVs, selects the next AV(i), and sends (RAND(i), AUTN(i)) to the user through a GMM Authentication and Ciphering Request;

5) U5: The UE checks whether the received AUTN can be accepted. If so, it produces a response RES which is sent back to the SGSN through a GMM Authentication and Ciphering Response message;

6) U6: The SGSN compares the received RES with the XRES. If they match, the SGSN considers the authentication and key agreement exchange to be successfully completed and sends back the UE GMM Attach Accept to accept the Attach.

B. IMS AKA Authentication

After the GPRS attach and GPRS PDP Context activation, the UE can access the GPRS services and the packet data network. The UE may then invoke the IMS layer authentication to request the IMS service access. The IMS AKA authentication procedure is shown in Fig. 3 which is defined in 3GPP TS 33.203 [12].

The IMS AKA authentication procedure includes:

1) I1: After activation of the PDP context, the UE sends a SIP Register request to the PCSCF with IMPI=impi and security-setup line. The security–setup line is used to negotiate security association (SA) parameters with PCSCF to create SAs which includes Security Parameters Index (SPI), the port number chosen, and the supported integrity and encryption algorithms.

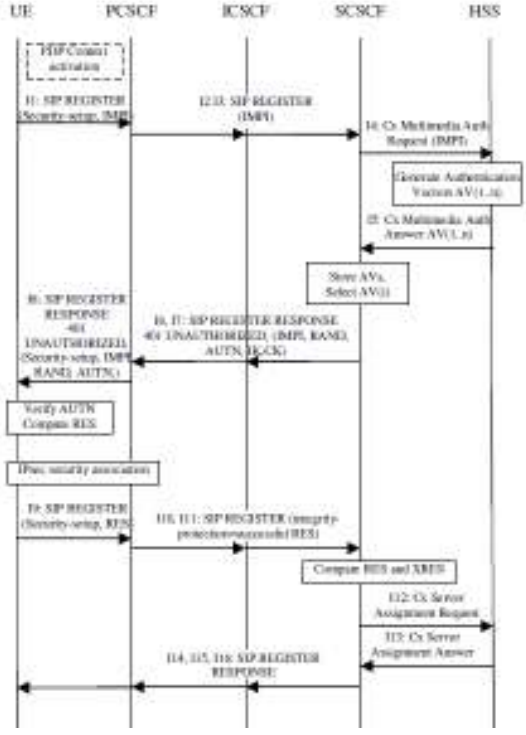

Figure 3 IMS AKA autentication procedure

2) I2, I3: PCSCF temporary stores the security-setup parameters; removes the security-setup line from the packet and forwards it to SCSCF through ICSCF;

3) I4: if the SCSCF does not have a previously stored authentication vector, it sends a Multimedia Auth Request to HSS with IMPI=impi through the Cx reference point;

4) I5: Upon receipt the request, HSS uses IMPI=impi to retrieve the user's record and generate Authenticaion Vectors (RAND, XRES, CK, IK, AUTN); then the HSS send the array of AVs to the SCSCF which is ordered by sequence number;

5) I6, I7: The SCSCF stores the AVs, selects the next AV(i) and sends (IMPI, RAND, AUTN, CK, IK) back to ICSCF through SIP 401 Unauthorized response; and ICSCF forwards it to PCSCF;

6) I8: Upon receipt the message, the PCSCF keeps the received authentication challenges, selects security parameters for creating the SAs and sends the (Security-setup, IMPI, RAND, AUTN) to the UE;

7) I9: The UE checks whether the received AUTN can be accepted. If so, it produces a response RES. On the other hand, the UE computes CK and IK, creates IPsec security associations which is used to protect the following SIP messages and sends SIP register message to PCSCF with the parameter Security-setup and RES in the security channel;

8) I10, I11: after PCSCF successfully checked the parameter of Security-setup, it would send the SIP Register packet to SCSCF through ICSCF with RES and integrity-protection=successful

9) I12: Upon receipt of the SIP request with the RES from the UE, SCSCF compares the received RES with the XRES. If they match, the SCSCF considers the authentication and key agreement exchange to be successfully completed and sends Server Assignment Request to HSS;

10) I13: Upon receipt of the Server Assignment Request, the HSS stores the SCSCF name and replies a Cx Server Assignment Answer message to the SCSCF;

11) I14, I15, I16: SCSCF sends SIP Register response 200 OK to UE through ICSCF and PCSCF.

IV. Improved one-pass authentication

3GPP uses the duplicated two-pass AKA authentication to enable the end user to access IMS services because of the following two reasons:

1) Two-pass authentication achieves mutual authentication when the IMPI and IMSI don't have a one to one relationship. According to 3GPP TS 23.228 [3] the relationship between IMPI and IMSI is shown in Figure 4; and

2) Two-pass authentication is a good way to keep the IMS system layers independent because the network layer and the service layer complete authentication separately.

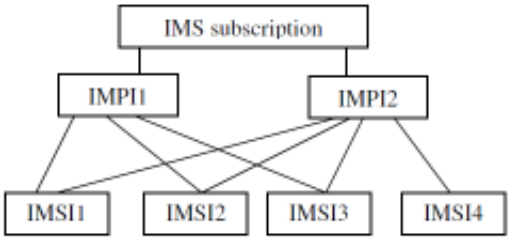

Figure 4. The relationship between IMPI and IMSI

The proposed IAKA authentication binds the two AKA procedures by using the unified IMPI number which is defined in 3GPP TS 23.228[3]. If the unified IMPI is used as an index in the HSS to do AKA authentication, the generated AV are the same in the two procedures. Hence both of the UMTS network layer and IMS service layer could use the unified AV to do mutual security authentication. The proposed IAKA authentication could not only reduce the authentication time but also keep the features of the legacy two-pass authentication.

A. The improved UMTS AKA authentication (initial authentication)

Compared to the initial UMTS AKA authentication procedures in the legacy two-pass authentication which is shown in Figure 3, the proposed initial IAKA UMTS AKA authentication has the following improvements:

1) In step U1: In order to access GPRS services, UE sends SGSN a GMM Attach Request with the parameter IMSI=imsi and IMPI=impi in which the parameter IMPI is optional;

2) In step U3: Upon receipt the request, HSS uses IMPI=impi as an index to retrieve the user's profile and generate AVs. If there isn't IMPI in the request, IMPI=imsi; and

3) In step U6: The SGSN compares the received RES with the XRES. If they match, the

SGSN considers the authentication and key agreement exchange to be successfully completed, SGSN distribute the AV(i) to PCSCF. At the same time the SGSN sends back the UE GMM Attach Accept to accept the Attach request.

After completing the initial authentication, the following has been achieved: □The UE and the SGSN have authenticated each other; □The AVs were generated by using the index IMPI (IMPI=imsi, if no IMPI is available); □The SGSN stores the AVs; □The UE has CK, IK to do further authentication; and □he PCSCF has the AV(i) from SGSN. Because SGSN and PCSCF are located in the same network, it is assumed that it is secure to distribute the AV(i)

B. The improved IMS AKA authentication (second authentication)

After the initial UMTS AKA authentication, both of the UE and the PCSCF have the same CK and IK which would be used in the second IMS AKA authentication to create IPSec security associations to transmit data safely. The procedure is shown in Figure 5 in which the first and second steps are used to negotiate the parameters of creating the security associations, and the step 3 to step 10 are used to do user registration in which all the SIP messages are integrity and confidentiality protected.

1) II1: After activation of the PDP context, the UE and the PCSCF starts to negociate the parameters of creating the security associations. Firstly, the UE sends a SIP Register request to the PCSCF with the parameters of IMPI and security-setup. The security-setup line includes Security Parameters Index (SPI), the port number chosen, and the supported integrity and encryption algorithms which are used to create the SA.

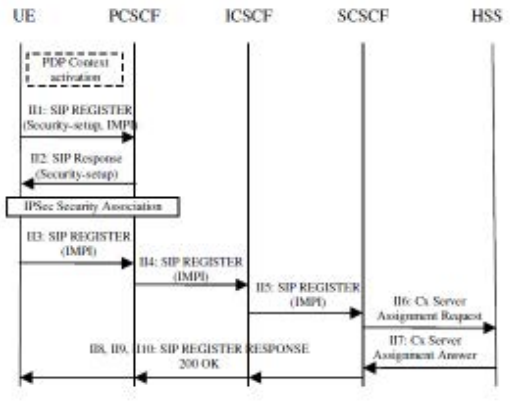

Figure 5 IMS AKA authentication procedure (IAKA)

2) II2: Upon receipt of II1, the PCSCF selects the corresponding SPI, port number, supported integrity and encryption algorithms and sends them back to the UE through the parameter of security-setup in the packet of SIP REGISTER RESPONSE 401 Unauthorised.

3) Upon receipt of II2, the UE creates the security association with the PCSCF by using the CK and IK obtained in the initial UMTS authentication and the following SIP messages would be integrity and confidentiality protected.

4) II3, II4, II5: the UE sends a SIP REGISTER request with IMPI=impi to PCSCF. The PCSCF extracts the SIP request from the IPSec Security Association and forwards it to SCSCF through ICSCF;

5) II4: Upon receipt the request, SCSCF sends a Cx Server Assignment Request to HSS with IMPI=impi;

6) II5: HSS uses IMPI=impi as an index to retrieve the user's profile and assign user's status; and then it sends back a Cx Server Assignment Answer to SCSCF;

7) II6, II7, and II8: if the registration is successful, the SCSCF sends a SIP 200 OK response to PCSCF through ICSCF, PCSCF sends the request to UE through IPSec Security association.

V. IAKA AUTHENTICATION SIMULATION

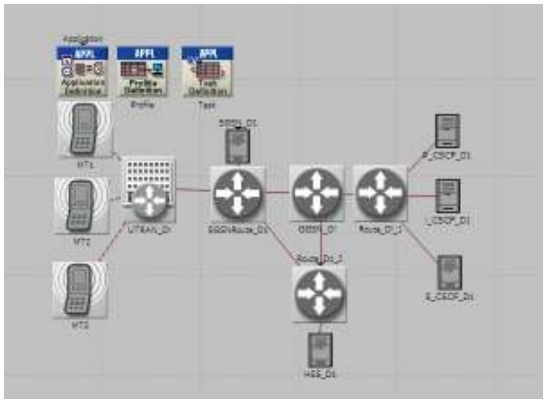

Figure 6 IAKA simulator system architecture

Following the references 3GPP TS 23.060 [2] and 3GPP TS 23.228 [3], the IAKA simulation system was developed using Opnet Modeler V16 and is shown in Figure 6.

Two authentication scenarios were developed to permit comparison between the legacy two-pass authentication and the proposed IAKA one-

pass authentication. In the first scenario, the legacy two-pass authentication procedures were defined based on 3GPP TS 33.102 [9] and 3GPP TS33.203 [12], and the proposed IAKA authentication procedures were defined in the second scenario. The Opnet Modeler [15] definition of the IMS AKA for the two authentication algorithms is shown in Figure 7 and Figure 8.

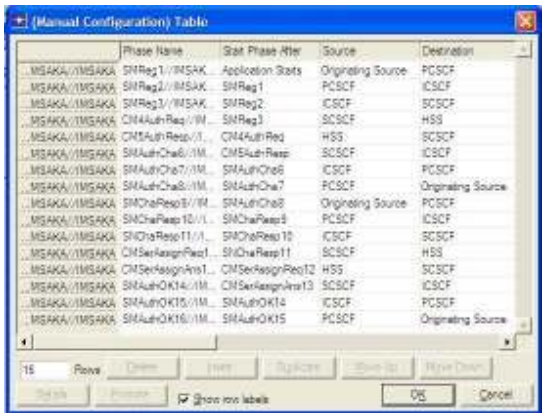

Figure 7 IMS AKA procedure definition for two-pass authentication

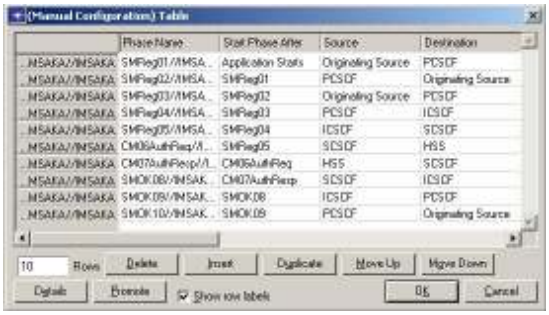

Figure 8 IMS AKA procedure definition for IAKA authentication

Two statistics were chosen to compare the results. The first is the task response time which is the time taken in the second step of the IMS AKA authentication. As shown in Figure 9, the average time is 2.483s in the legacy two-pass authentication, and the average time is 1.704s in the proposed IAKA authentication. The proposed IAKA algorithm reduced the IMS-AKA authentication time by up to 31.4%.

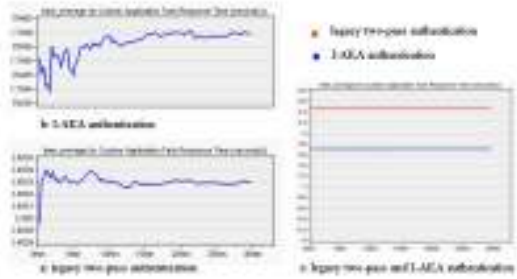

Figure 9 IMS AKA task response time

The second statistic is the application response time which is the time taken for the complete authentication process (UMTS AKA and IMS AKA). As shown in Figure 10, the average authentication time in the two-pass authentication is 3.254s, and the average authentication time in the proposed IAKA is 2.475s. The proposed IAKA algorithm reduced the complete UMTS IMS authentication by up to 24%.

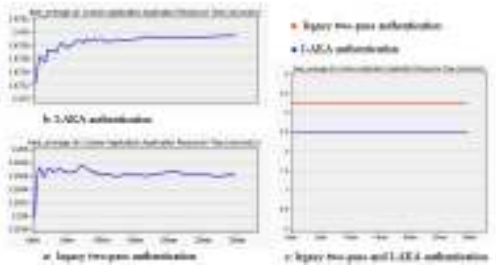

Figure 10 Application response time

The performance results shown in Fig. 9 and Fig. 10 demonstrate that the proposed IAKA authentication algorithm out performs the legacy two-pass authentication.

VI. Security Analysis

The IAKA authentication process provides a robust secure mechanism that satisfies the 3GPP requirements. In this section the IAKA security approach is presented.

A. Mutual Authentication between UE and PCSCF

After the initial UMTS GPRS authentication, the UE, the SGSN, and the PCSCF have the same keys generated by the HSS by using the unified IMPI number. By using the shared keys, the UE and the PCSCF could do mutual authentication to ensure they are connecting to a trusted entity. Hence an adversary terminal couldn't pretend to be a valid user to access IMS services and attack the network. Also, adversary terminals couldn't

pretend to be a valid network device and attempt to get UE security information.

B. Advanced security services

Before exchanging confidential information on the network, there is an IPSec security association created between the UE and the PCSCF which provides confidentiality and integrity of data transmissions. The IPSec security association provides a secure data transmission tunnel to withstand eavesdropping.

VII. IAKA authentication

In the initial authentication, the HSS retrieves the UE profile and generates AVs by using IMPI as an index. After completing the first authentication, the UE, SGSN, and PCSCF share the AV. The AV is made up of RAND, XRES, CK, IK, and AUTN parameters.

A cipher key CK = $f3_K$ (RAND) where f3 is a key generating function;

An integrity key IK = $f4_K$ (RAND) where f4 is a key generating function;

K is a pre-shared key between the HSS and the UE related to the unified IMPI number.

In the second authentication, the UE and the PCSCF authenticate each other by using CK and IK, because CK and IK are derived from the same pre-shared key which is allocated to the user when the user subscribes to the service by using IMPI number. This means that only UE with a valid IMPI number, valid pre-shared key, and that have passed the initial authentication can register with the IMS system. Therefore, the proposed IAKA authentication procedure can be used to replace the legacy two-pass authentication procedure.

VIII. Conclusion

This paper presents a new IAKA authentication procedure which has the following features: (1) reduced authentication time; (2) reduced the AV calculation time within the HSS and the UE; (3) achieves the 3GPP security requirements; and (4) minimal impact on the existing system. The IAKA authentication procedure provides improved performance without compromising security. The research simulation model was based upon UMTS and future work is to extend the research to the other access network technology, such as Long Term Evolution and Next Generation Networks.

IX. References

[1] Rebecca Copeland, Converging NGN Wireline and Mobile 3G Network with IMS, Taylor & Francis Group, U.S.A, 2009

[2] 3GPP TS 23.060 V9.4.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service description; Stage 2 (Release 9)”, March 2010

[3] 3GPP TS 23.228 V10.0.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 10)”, March 2010

[4] Y.B. Lin, M.F. Chang, M.T. Hsu, L.Y. Wu, "Onepass GPRS and IMS Authentication Procedure for UMTS", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 6, pp: 1233-1239, Jun.2005.

[5] Chung-Ming Huang and Jian-Wei Li, “Efficient and Provably Secure IP Multimedia Subsystem Authentication for UMTS”, The Computer Journal, Vol 50, No.6, pp739–757, 2007

[6] Chung-Ming Huang and Jian-Wei Li, “Reducing Signaling Traffic for the Authentication and Key Agreement Procedure in an IP Multimedia Subsystem”, Wireless Personal Communications, Vol51, pp95-107, 2009

[7] Christoforos Ntantogian and Christos Xenakis, “One-pass EAP-AKA Authentication in 3G-WLAN Integrated Networks”, Wireless Personal Communications, Vol48, pp569-584, 2009

[8] Christoforos Ntantogian, Christos Xenakis, Ioannis Stavrakakis, ” A generic mechanism for efficient authentication in B3G networks”, Computers & Security, Vol29, pp460-475, 2010

[9] 3GPP TS 33.102 V9.1.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 9)”, Dec 2009

[10] 3GPP TS 29.002 V9.1.0, “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mobile Application Part (MAP) specification (Release 9)”, March 2010

[11] 3GPP TS 24.008 V9.2.0, “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 9)”, March 2010

[12] 3GPP TS 33.203 V9.3.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 9)”, Dec 2009

[13] 3GPP TS 24.228 V5.15.0 “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 5)”, Sep 2006

[14] 3GPP TS 24.229 V9.3.1 “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 9)”, Mar 2010
[15] OPNET Technologies, Inc., Opnet Modeler - ver. 16.0, http://www.opnet.com

Appendix C

Australasian Telecommunication Networks And Applications Conference (ATNAC 2011)
9-11 November 2011, Melbourne, Australia

A Green and Secure Authentication for the 4Th Generation Mobile Network

Lili Gu, Mark A Gregory
School of Electrical and Computer Engineering
RMIT University
Melbourne, Australia
l.gu@student.rmit.edu.au, mark.gregory@rmit.edu.au

Abstract*—The 4th generation (4G) mobile access network and the core network are evolving towards a secure, fast, and Internet Protocol-based network. With the emergence of next generation battery-powered smart mobile phone and open source application platforms, security and the terminal's energy consumption have become big issues. Long Term Evolution (LTE) is one of the most popular 4G technologies defined by 3rd Generation Partnership Projects (3GPP). It is observed that the end user requires two authentication steps to access multimedia services. The first is the LTE network layer authentication, and the second is the IP Multimedia Subsystem (IMS) service layer authentication. The authentication steps utilize energy and are carried out using the Authentication and Key Agreement (AKA) protocol. This paper proposes an Improved AKA (IAKA) authentication protocol which authenticates the user on both the network layer and the service layer without double execution the AKA protocol and simplifies the authentication steps. Furthermore, the security and energy consumption were analyzed and the results showed the proposed IAKA could save up to 81.82% of the terminal's energy consumption related to authentication with increased security.

***Keywords*—4G; Authentication; IP Multimedia Subsystem (IMS); Long Term Evolution (LTE); Authentication and Key Agreement (AKA); Session Initiation Protocol (SIP); Call Session Control Function (CSCF) .**

I. introduction

The 4th Generation (4G) mobile access network and the core network are evolving towards a common IP based secure and fast transport layer. As a popular 4G mobile technology Long Term Evolution (LTE) may provide up to 1Gbps peak data rate. Within the core network, the IP Multimedia Subsystem (IMS) is the candidate for providing the next generation wired and wireless Packet Switched Networks (PSN) with multimedia services (i.e., audio, video, text, image, and combinations). However, with the emergence of new battery-powered smart mobile phones and tablets, security and the terminal's energy consumption have become more important issues [1][2]. It is observed that a LTE user device carries out two authentication steps to get access to the multimedia services and this increases authentication complexity and the terminal's energy consumption.

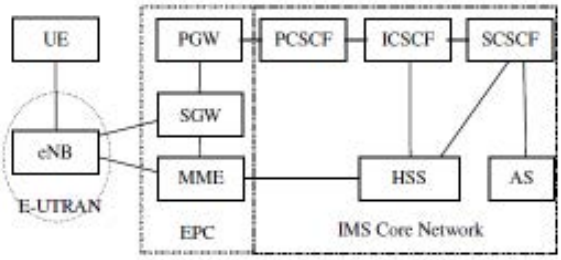

Figure1 LTE and IMS integrated system architecture

LTE and IMS were both defined by 3rd Generation Partnership Projects (3GPP) and the integrated system architecture is shown in Fig. 1 [3] [4]. In order to access the multimedia services, LTE users have to be authenticated in both the LTE network layer and the IMS service layer.

The LTE network layer authentication occurs first. After receiving the user's request, the Mobility Management Entity (MME) sends a message to the Home Subscriber Server (HSS) to request an Authentication Vector (AV); the HSS generates the AV by using the user's International Mobile Subscriber Identity (IMSI) number and distributes the AV to the MME and the User Equipment (UE); and the UE and the MME authenticate each other by using the received AV. After a successful network

layer authentication, the terminal can access the LTE and packet data network.

The second authentication is the IMS service layer authentication. Through the Proxy Call Session Control Function (PCSCF) and Interrogating CSCF (ICSCF), the terminal sends a request to the Serving CSCF (SCSCF); the SCSCF requests the AVs from the HSS which was generated by the user's IP Multimedia Private-user Identity (IMPI) number; and then the SCSCF sends the AVs to the ICSCF, PCSCF and the terminal. The terminal and the network entities authenticate each other by using the received AV. After a successful service layer authentication, the terminal can access the multimedia services in IMS layer.

This 3GPP defined authentication protocol suffers from two main limitations.

Firstly, it increased the terminal's energy consumption and the system complexity. The 3GPP defined authentication approach executed the Authentication and Key Agreement (AKA) protocol twice, once in each layer, including requesting, generating and distributing the AV and mutual authentication by using the received AV. To the battery-powered mobile terminal which is very sensible to energy consumption, this approach brings high energy cost and reduces the terminal battery life.

Secondly, the 3GPP defined authentication protocol is vulnerable to a Denial of Service (DOS) attack. After receiving a request, the PCSCF/MME sends the request to the core network (ICSCF/SCSCF/HSS) to do authentication, which means, a malicious attack could flood the ICSCF/SCSCF/HSS by sending correct packets with invalid IMSI/IMPI numbers.

In this paper, an Improved AKA (IAKA) authentication protocol is proposed to reduce energy consumption and increase security. A secure binding of the network layer and service layer authentication by using the IMPI number is used to avoid the double execution of the AKA protocol. Energy consumption and security were modeled and the results are provided in this paper. Results and analysis showed: (a) the proposed IAKA authentication protocol could reduce energy consumption by up to 81.82% in the IMS service layer and 39.13% across the two layers; and (b) the proposed IAKA approach can not only meet 3GPP defined security requirements but will also provide improved security against DOS attacks.

The remainder of the paper is organized as follows: Section II is the related work; Section III describes the 3GPP defined security architecture; Section IV describes the proposed IAKA authentication protocol; Section V is the energy consumption analysis, Section VI provides an analysis of the security of the proposed IAKA; and Section VII provides a summary, conclusion and discussion of future work.

II. Related Work

Authentication is a very important aspect of access networks and research is constantly being carried out regarding next generation network design [5] … [12]. However the common limitation is either a lack of security or the need to authenticate on more than one network layer. In this section recent research into authentication is presented.

Huang and Li [8] proposed a one-pass IMS AKA to replace the 3GPP defined IMS which could save 45% of the authentication signaling and 76.5% of the storage space without compromising security. In the proposed mechanism, the terminal is responsible for generating a random number, IMS key, digest password, and digest response, distributing the AV to the server, and checking the digital password after receiving the REGISTER response. The SCSCF has to check the random number, generate digest password, Cipher key (CK), and Integrity Key (IK). Compared to the legacy IMS AKA, the proposed mechanism introduced extensive modification and increased complexity.

Ntantogian et al. [12] proposed a generic mechanism for efficient authentication in Beyond 3rd Generation (B3G) networks to reduce the execution of the 3GPP defined multi-pass authentication steps by using a security binding mechanism. The proposed mechanism authenticates a user in the second and third step of a multi-pass authentication by using the user's authentication credentials of the initial step. The proposed IMS service layer authentication used a (impi, imsi) pair to authenticate the user without the security protection between the UE and the PCSCF. However, this approach is very vulnerable to the fraudulent use of IMS services, eavesdropping attack, Fake server attack, and Temporary cheat.

III. 3GPP security architecture

A. 3GPP LTE key hierarchy

In order to provide adequate security protection for different traffic flows, 3GPP LTE introduced the key hierarchy.

As shown in Fig. 2 [14], K is the pre-shared key passed between the UE and the HSS. K is used to generate CK and IK and kept in the Universal Subscriber Identity Module (USIM) of the UE and the Authentication Center (AuC) of the HSS. KASME is derived from CK and IK by HSS and sent to MME, and then, KASME is stored in the MME

and is used to derive five keys for protection of three different flow types.

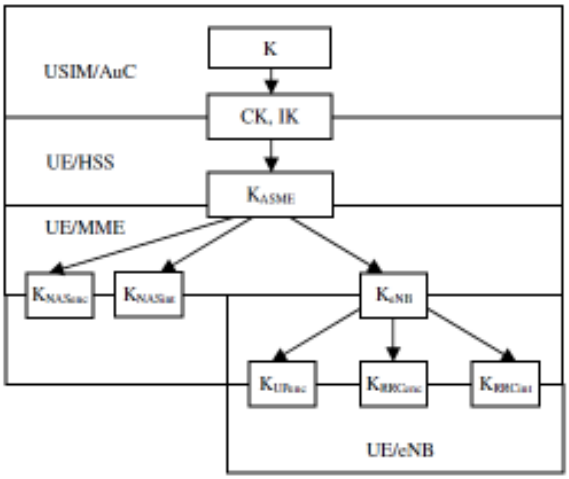

Figure2 LTE key hierarchy

Non Access Stratum (NAS) signaling between the terminal and the MME is confidentiality and integrity protected by K_{NASenc} and K_{NASint}. The traffic between the terminal and the eNB is confidentiality and integrity protected by K_{RRCenc} and K_{RRCint}. And the user plane data between the terminal and the SGW is protected by K_{UPenc}.

B. 3GPP IMS key hierarchy

As shown in Fig. 3 [15], IMS uses the two level key hierarchy to protect traffic which is a subset of the LTE five level key hierarchy. In IMS, the HSS generates CK and IK by using the pre-shared key K and distributes them to SCSCF and PCSCF. The traffic between the terminal and the PCSCF is confidentiality and integrity protected by CK and IK.

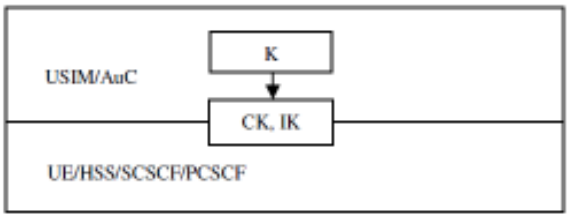

Figure3 IMS key hierarchy

IV. Improved AKA Authentication

Since the 4G LTE and IMS network could provide voice, video and data services in one IP-based network, the IMPI number could be used in both of the network layer and service layer to do registration, authorization, administration and accounting. This paper proposes IAKA, an improved authentication protocol, which provides a secure binding of the network layer and service layer authentication by using the IMPI number which avoids the double execution of the AKA authentication protocol. Furthermore, in order to enhance the security and improve the compatibility between the LTE and IMS, a 4 layer key hierarchy is proposed. As the results show, the proposed IAKA could save the terminal's energy consumption significantly with enhanced security.

A. The proposed IMS key hierarchy

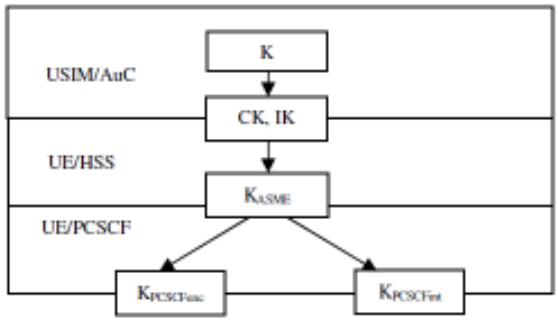

Figure4 proposed IMS key hierarchy

The proposed IMS four layer key hierarchy is shown in Fig. 4. Compared to the original two layer key hierarchy, the intermediate key K_{ASME} is used to derive $K_{PCSCFenc}$ and $K_{PCSCFint}$ to protect the traffic between the UE and the PCSCF with a particular encryption or integrity algorithm.

$$K_{PCSCFenc} = KDF (K_{ASME} , s) \quad (1)$$

$$K_{PCSCFint} = KDF (K_{ASME} , s) \quad (2)$$

Where KDF is key derivation function and s is the input string to the KDF. P0 is a part to construct input string s which needs to be extended to support the key derivation of $K_{PCSCFenc}$ and $K_{PCSCFint}$ as shown in Table I.

Table I Algorithm Type distinguishers

Algorithm distinguisher	Value
PCSCF_enc_alg	0x06
PCSCF_int_alg	0x07

B. The improved EPS AKA authentication

When the mobile terminal is powered on, it invokes the Attach procedure to access the LTE network. The Evolved Packet System (EPS) AKA [14] is triggered by the Attach procedure to provide mutual authentication and agree on key KASME.

The 3GPP defined EPS AKA authentication protocol utilizes the IMSI number as the identity to do authentication [14]. The improved EPS AKA

authentication protocol presented in this paper supports mutual authentication by using the IMPI number within the HSS to generate AV. The procedure is shown in Fig. 5.

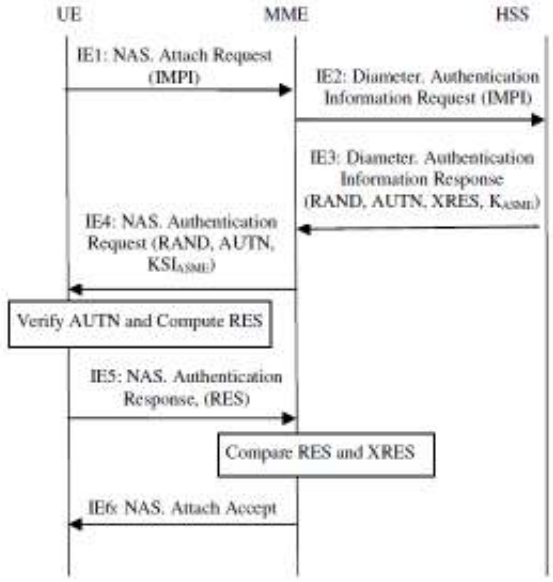

Figure 5 Improved LTE EPS AKA autentication procedure

1) IE1: In order to access the LTE network, the UE sends an Attach Request to the MME with IMPI number through the NAS protocol.

2) IE2: If there isn't a valid AV in the MME, the MME sends an Authentication Information Request (AIR) to the HSS to fetch the AV by using the Diameter protocol.

3) IE3: The HSS uses the IMPI number to fetch the user's profile; generate the AV (RAND, AUTN, XRES, K_{ASME}); and send the AV back to the MME.

4) IE4: After receiving the AV from the HSS, the MME sends Authentication Request (RAND, AUTN, KSI_{ASME}) to the UE to start the authentication procedure. KSI is Key Set Identifier which is used to identify KASME.

5) IE5: After receiving the Authentication Request, The UE checks the authentication code AUTN first. If this is a valid AUTN number, the terminal considers this is a trusted network entity and keeps working with it. Then, the UE calculates RES number and sends it back to the MME.

6) IE6: The MME checks whether the RES from the UE matches the XRES from the HSS, if they match, the MME considers that the authentication and key agreement exchange is successfully completed and sends a message back to the UE to accept the Attach request.

After a successful IAKA EPS authentication, the UE and the MME have completed the authentication steps and have the same K_{ASME} which is used to derive more keys for different security protection purposes.

C. The improved IMS AKA authentication

The improved IMS layer authentication is shown in Fig.6. Steps II1 and II4 are used to build security associations; steps II2 and II3 are for AV synchronization; and steps II5 to II12 are used to do authentication and all of the SIP messages are integrity and confidentiality protected by using IPSec Encapsulating Security Payload (ESP)[18].

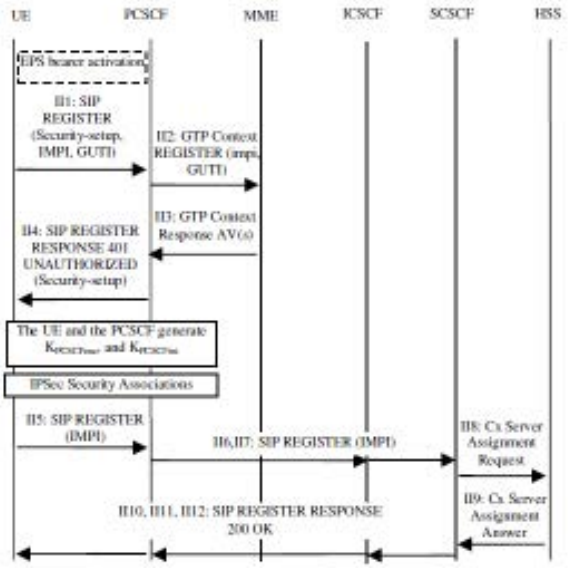

Figure 6 The improved IMS AKA autentication procedure

1) II1: After creation of the EPS bearer, the UE sends a SIP REGISTER request to the PCSCF to negotiate the parameters of building SAs with the IMPI number, Global Unique Temporary Identity (GUTI) number and security-setup line. The security-setup line includes the parameters to build the IPSec SAs;

2) II2: The PCSCF derives the MME address from GUTI and sends a Context Request to the MME to fetch the AV with IMPI and GUTI. The protocol between the PCSCF and the MME is the GTP [16] and the security is defined in [17].

3) II3: The MME obtains the user's AV by using GUTI number and sends the AV back to the PCSCF;

4) II4: Upon receipt of the AV from the MME, the PCSCF chooses the parameters to build the SA, uses the parameters to construct the security-setup line; and sends it back to the UE. Then the PCSCF derives two keys $K_{PCSCFenc}$ and the $K_{PCSCFint}$;

5) II5: The UE derives $K_{PCSCFenc}$ and KPCSCFint and builds the IPSec SAs; then it sends the SIP

REGISTER request to the PCSCF through the IPSec SA;

6) II6: The PCSCF checks the received SIP packet by decryption and calculation the integrity code. If this is a valid packet, the PCSCF forwards it to the ICSCF;

7) II7: The ICSCF fetches the user and the SCSCF information from the HSS, locates the SCSCF address, and sends the packet to the SCSCF;

8) II8: The SCSCF sends a Server Assignment Request to HSS with IMPI=impi by using the Diameter protocol;

9) II9: The HSS fetches the use's profile and assigns status by using the IMPI number, and then sends the Server Assignment Answer back to the SCSCF;

10) II10 and II11: The SCSCF sends the SIP response to the ICSCF and the ICSCF forwards it to the PCSCF.

11) II12: By using IPSec, the PCSCF sends the packet to the UE and the UE authenticates the PCSCF by checking the ICV (Integrity Checking Value) of the packet. If this is a valid ICV code, the network entity passes the authentication and continues interacting with the terminal.

A comparison of the IMS layer authentication of the proposed IAKA and the 3GPP defined approach is provided in TABLE II. The comparison identifies the proposed IAKA would simplify the authentication procedures significantly.

IMS layer authentication comparison

IAKA	3GPP
SIP Register (UE->PCSCF)	SIP Register (UE->PCSCF, PCSCF ->ICSCF, ICSCF->SCSCF)
-	Diameter Multimedia Auth Request (SCSCF->HSS)
Obtaining AV from MME (PCSCF->MME, MME->PCSCF)	-
-	Calculation of AVs by HSS
-	Diameter Multimedia Auth Answer (HSS->SCSCF)
-	Storing of AVs by SCSCF
SIP response 401 UNAUTHORIZED (PCSCF->UE)	SIP response 401 UNAUTHORIZED (SCSCF->ICSCF, ICSCF->PCSCF, PCSCF->UE)
-	Verification of AUTN and calculation of RES number by UE
Creating IPSec SAs	Creating IPSec SAs
SIP Register (UE->PCSCF, PCSCF->ICSCF, ICSCF->SCSCF)	SIP Register (UE->PCSCF, PCSCF ->ICSCF, ICSCF->SCSCF)
-	Compare RES and XRES by SCSCF
Diameter Server Assignment Request / Answer (SCSCF->HSS, HSS->SCSCF)	Diameter Server Assignment Request / Answer (SCSCF->HSS, HSS->SCSCF)
SIP REGISTER response 200OK (SCSCF->ICSCF, ICSCF->PCSCF, PCSCF->UE)	SIP REGISTER response 200OK (SCSCF->ICSCF, ICSCF->PCSCF, PCSCF->UE)

V. Energy Cost Analysis

In this section, the energy cost of the user terminal's authentication related security activities was calculated. It is shown that the IAKA authentication protocol could save 81.82% of the energy consumption in IMS layer authentication and save 39.13% of the terminal's energy consumption in both the network layer and service layer.

For the 3GPP defined and proposed IAKA authentication protocols, the terminal's security activities include the execution of the EPS AKA and the IMS AKA authentication protocol. Therefore, $E = E_{EPS\text{-}AKA} + E_{IMS\text{-}AKA}$ where E denotes the energy consumption.

In the 3GPP defined authentication protocol, the terminal's energy cost to execute the EPS AKA authentication protocol includes (a) checking the AUTN number which is made up of generating AK and MAC (b) generating the RES number, and (c) generating CK, IK and deriving KASME as shown in (3).

$$E_{3GPP\text{-}EPS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} + E_{KASME} \quad (3)$$

The energy cost to execute the IMS AKA authentication protocol is shown in (4).

$$E_{3GPP\text{-}IMS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} \quad (4)$$

For the IMS layer, the IPSec energy consumption wasn't considered because IPSec is mainly used for message transmission and it is the same in the two approaches. This paper uses the AV generation functions f1, f2, f3, f4, f5 defined in [19] to generate Message Authentication Code (MAC), RES, CK, IK, Anonymity Key (AK), and choose Advanced Encryption Standard (AES) as the kernel encryption algorithm. Also, HMAC-SHA-256 is used as the key derivation function to derive K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$. For the energy consumption calculation, only the energy used in encryption is considered. The security operations of bitwise exclusive-OR and bitwise rotation aren't calculated as they won't affect the results significantly. The energy consumption of AES and HMAC was studied in [1], and the results showed the energy consumption of the AES algorithm is made up of two phases, the first is in the key setup phase EKEY-SETUP which is 7.87 μJ and the second is in the encryption/decryption phase $E_{ENC/DEC}$. In the encryption/decryption phase, the energy consumption per byte (EPB) is 1.21 μJ and for the HMAC-SHA-256, the EPB is 1.16 μJ. Therefore the energy can be calculated as shown in (5) and (6).

$$E_{AES}(n) = 7.87\ \mu J + 1.21\ \mu J * n \quad (5)$$

$$E_{HMAC}(n) = 1.16\ \mu J * n \quad (6)$$

where n is the length of the input string in bytes. Although the MAC, RES, CK, IK, and AK have different lengths, they were all generated by encrypting 16 byte data blocks three times. Therefore, referring to (5), $E_{AK} = E_{MAC} = E_{RES} = E_{CK} = E_{IK} = 81.69\ \mu J$. For K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$, they were derived by using 32 byte data blocks so the energy consumption is 37.12 μJ by using (6). For the 3GPP defined authentication protocol, referring to (3) and (4), (a) $E_{3GPP\text{-}EPS\text{-}AKA} = 445.57\ \mu J$, and (b) $E_{3GPP\text{-}IMS\text{-}AKA} = 408.45\ \mu J$.

For the improved IAKA authentication protocol, the LTE network layer energy consumption is same as it in the 3GPP approach which is 445.57 μJ ($E_{IAKA\text{-}EPS\text{-}AKA} = 445.57\ \mu J$). The energy cost of the IMS service layer includes key derivation of $K_{PCSCFint}$ and $K_{PCSCFenc}$ by using HMAC-SHA-256 which is 74.24 μJ ($E_{IAKA\text{-}IMS\text{-}AKA} = 74.24\ \mu J$).

$$ESR_{I\text{-}IMS\ AKA} = (E_{3\text{-}IMS\text{-}AKA} - E_{I\text{-}IMS\text{-}AKA}) / E_{3\text{-}IMS\text{-}AKA} * 100\% \quad (7)$$

$$ESR_{IAKA} = (E_{3GPP} - E_{IAKA}) / E_{3GPP} * 100\% \quad (8)$$

By using (7) and (8), $E_{SRI\text{-}IMS\ AKA} = 81.82\%$ and $E_{SRIAKA} = 39.13\%$. This means that by using the improved authentication protocol, the terminal could save 81.82% of IMS layer security related activity energy consumption; and save 39.13% of the security activity energy consumption in both the network layer authentication and the IMS service layer authentication by using the improved authentication protocol.

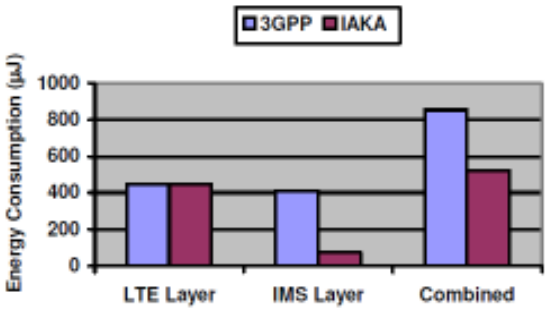

Figure 7 Energy consumption comparison

VI. Security Analysis

A. Authentication accuracy and mutual authentication

In the proposed IAKA authentication protocol, the UE and the network authenticate each other in both of the network layer and IMS service layer. Only terminals and the network entities with a valid key identified by the impi number can pass authentication and build a trust relationship.

At the network layer, the UE authenticates correctly using the impi number. It is assumed there is a mobile terminal with IMPI = impi and a pre-shared key K = kue, and a network entity with a pre-shared key K = khss. Firstly, the network entity generates a random number RAND and authentication code AUTN and sends them to the UE.

$$AUTN = SQN \oplus AK \,||\, AMF \,||\, MAC \quad (9)$$

$$MAC = f1\,(SQN \,||\, RAND \,||\, AMF, K) \quad (10)$$

Where SQN is the sequence number, AMF is authentication management field, f1 is the message authentication function and K is the key used. After receiving the packet, the UE gets MAC from the AUTN, calculates its own XMAC, and compares them. If MAC is equal to XMAC, the network entity passes the authentication and continues interacting with the terminal. Only the network entity with a correct key khss identified by the terminal's IMPI number could generate the MAC which is equal to the XMAC generated by the terminal by using the key kue = khss, otherwise, XMAC ≠ MAC, and the terminal will reject the network's authentication request.

The network also can authenticate the terminal correctly on the network layer by checking the RES number according to (11).

RES=f2 (RAND, K) (11)

Where f2 is a message authentication function and K is the pre-shared key identified by IMPI number. To the MME, if the RES received from the terminal equals to the XRES received from the HSS, the terminal passes the authentication and has the right to access the network. For an invalid terminal with a stored key kue ≠ khss, its generated RES must be different with the XRES generated by the HSS by using a different khss. Therefore, the MME won't let the terminal access the network resources.

On the IMS service layer, the UE and the Network entity could authenticate each correctly by checking the Integrity Checking Value (ICV) received in SIP Register request/response packets encapsulated by IPSec ESP (Encapsulating Security Payload) protocol as shown in (12).

ICV = fint (KPCSCFint, M) (12)

Where fint is the integrity code generation function and M is the input string used to generate ICV. For the IPSec ESP protocol, the input string M is the ESP packet minus the ICV. The IMS service layer authentication is performed in the PCSCF and the terminal. After a successful network layer authentication, the valid terminal and PCSCF have the same key K_{ASME} to derive $K_{PCSCFint}$ to protect the data transmitted between them and $K_{PCSCFintue}$ = $K_{PCSCFinthss}$. For the sender (UE/ PCSCF), it calculates the ICV by using the key $K_{PCSCFint}$, placing the ICV at the end of the ESP packet and sending the packet to the receiver. After receiving the packet, the receiver (PCSCF/ UE) gets the ICV from the last part of the ESP packet, calculates the XICV by using the key $K_{PCSCFint}$ and compares the ICV and the XICV. If the ICV is equal to the XICV, the sender passes authentication and continues interacting with the receiver. Without a successful network layer authentication, the UE and the PCSCF won't have the same K_{ASME} identified by the IMPI number to derive the same $K_{PCSCFint}$ and generate the same integrity checking code ICV to pass the authentication. Therefore, in the proposed IAKA authentication protocol, the UE and the PCSCF authenticate each other correctly in the IMS service layer.

B. Confidential and integrity protection

The proposed IAKA supports confidential and integrity protection of the messages transmitted between the UE and the PCSCF to avoid the disclosure of confidential information. The network layer security isn't discussed here for it is the same as the 3GPP definition.

For the IMS service layer, as shown in Fig. 6, after parameter negotiation, two IPSec SA pairs are created to protect the information transmitted between the terminal and the PCSCF. All of the information transmitted in IPSec SAs is encrypted by the symmetric encryption algorithm. For the sender, encrypted text = fenc (plain text, $K_{PCSCFenc}$) where fenc is the symmetric encryption function. Only the receiver with the same key $K_{PCSCFenc}$ can decrypt the message.

The messages between the terminal and the PCSCF are also integrity protected by using the ICV number. By using (12), (a) the received ICV = fint ($K_{PCSCFint}$, m_{sent}), (b) the calculated XICV = fint ($K_{PCSCFint}$, $m_{received}$), and (c) $m_{sent} \neq m_{received}$, if the received message is incomplete or it is modified in transit. This is a complete message if the received ICV equals the calculated XICV. Otherwise, with the correct key $K_{PCSCFint}$ and ICV ≠ XICV, the message is incomplete.

C. Possible Attacks

This section analyzed how the proposed IAKA prevents the replay attack and DOS attack. Firstly, the proposed IAKA avoids replay attacks in the network and service layers. At the network layer, an adversary might sit between the terminal and the MME and capture packets passing between them. The adversary might capture the (RAND, AUTN) and send it to the terminal. After receiving the faked packet, the terminal checks the SQN which is a part of the AUTN and the value should be different in different sessions. If the received SQN is equal to the SQN accepted previously, the terminal considers the packet to be an invalid packet and sends an error message to the server. Therefore, the faked server couldn't carry out a replay attack on the terminal get confidential information. Secondly, the faked server might capture RES and send it to the MME, however, the adversary couldn't pass a check of the RES number. In each new session, the MME receives XRES from the HSS which is generated by using the key and the new random number RAND. If the received RES from the terminal doesn't match the new XRES from the HSS in this session, the MME will reject the terminal's request and close the session. Therefore, in the network layer, the adversary couldn't carry out a replay attack on the server and get access to the network. At the service layer, all of the packets transmitted between the terminal and the PCSCF are protected by the IPSec ESP protocol. Sequence number is a field in the ESP packet used to prevent a replay attack. Upon receiving a packet, the receiver (UE/PCSCF) verifies the sequence number first to make sure this is not a duplicated packet. If it is, the receiver will discard the packet and send an error message back to the sender. Therefore, the adversary could not achieve a replay attack by sending captured packets in the service layer to get access to confidential information.

The proposed IAKA has a higher security level against DOS attacks than the legacy 3GPP

authentication protocol. For the improved IAKA authentication, it is assumed that the adversary is trying to flood the core network by sending SIP Register request with a valid IMPI number which belongs to a subscriber. Firstly, if the subscriber hasn't registered with the LTE network and there is no AV stored in the MME, the PCSCF can't get the authentication vectors from the MME and keep on serving the terminal. Secondly, if the subscriber has registered with the LTE network, the adversary will be asked to transmit information from the IPSec SAs. Without the correct pre-shared key, the messages sent by the adversary can't get past the PCSCF and reach the core network. Therefore, the adversary cannot jeopardize the core ICSCF / SCSCF / HSS servers. The DOS attack can only reach the PCSCF and the MME and the core network is fully protected.

VII. Conclusion

The paper proposes a green and secure authentication protocol for the 4th generation LTE and IMS integrated system which reduces the authentication procedures significantly by a secure binding of the network and service layer authentication using the IMPI number. An in-depth analysis of the energy consumption and security was provided and it is shown that the proposed IAKA authentication protocol could save up to 81.82% of the energy consumption for the IMS layer authentication with increased security. Future research work will focus on performance analysis.

VIII. References

[1] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols," IEEE Trans. on Mobile Computing, vol. 5, pp. 128-143, Mar. 2006.

[2] P. Prasithsangaree, and P. Krishnamurthy, "On a framework for energy-efficient security protocols in wireless networks," Computer Communications, vol. 27, pp. 1716-1729, 2004.

[3] 3GPP TS 23.401 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 10)", Jun. 2010.

[4] 3GPP TS 23.228 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 10)", Mar. 2010.

[5] Y. Lin, M. Chang, M. Hsu, and L. Wu, "One-Pass GPRS and IMS Authentication Procedure for UMTS," IEEE J. Selected Areas in Communications, Vol. 23, pp. 1233-1239, Jun. 2005.

[6] Y. Zhang, and M. Fujise, M, "An Improvement for Authentication Protocol in Third-Generation Wireless Networks," IEEE Trans. Wireless Communications, vol. 5, pp. 2348–2352, Sep. 2006.

[7] C. Huang and J. Li, "Efficient and Provably Secure IP Multimedia Subsystem Authentication for UMTS," The Computer Journal, Vol. 50, pp. 739–757, Oct. 2007.

[8] C. Huang and J. Li, "Reducing Signaling Traffic for the Authentication and Key Agreement Procedure in an IP Multimedia Subsystem", Wireless Personal Communications, Vol. 51, pp. 95-107, 2009.

[9] C. Ntantogian and C. Xenakis, "One-pass EAP-AKA Authentication in 3G-WLAN Integrated Networks," Wireless Personal Communications, Vol. 48, pp. 569-584, 2009.

[10] L. Gu and M. A. Gregory, "Improved One-Pass IP Multimedia Subsystem Authentication for UMTS", International Conf. on Information Networking, Kuala Lumpur Malaysia, Jan 2011.

[11] Y. Deng et al., "A Novel 3GPP SAE Authentication and Key Agreement Protocol", in Proc. IEEE International Conf. on Network Infrastructure and Digital Content, 2009, pp. 557–561.

[12] C. Ntantogian, C. Xenakis, and I. Stavrakakis, "A generic mechanism for efficient authentication in B3G networks," Computers and Security, Vol. 29, pp. 460-475, 2010.

[13] 3GPP TS 33.102 V9.1.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 9)", Dec. 2009.

[14] 3GPP TS 33.401 V9.4.0 "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 9)", Jun. 2010.

[15] 3GPP TS 33.203 V9.3.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 9)", Dec. 2009.

[16] 3GPP TS 29.274 V10.1.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 (Release 10)", Dec 2010

[17] 3GPP TS 33.210 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Network Domain Security (NDS); IP network layer security (Release 10)", Oct. 2010.

[18] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, Nov. 1998

[19] 3GPP TS 35.206 V9.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 9)", Dec 2009

Appendix D

Revised version was submitted to IEEE Transactions on Wireless Communications at 7 Dec. 2011

Optimized Authentication and Key Agreement Protocol for 4G Long Term Evolution and IP Multimedia Subsystem

Lili Gu, *Student Member*, and Mark A Gregory, *Senior Member*, IEEE

***Abstract*—Long Term Evolution (LTE) is a 4th generation mobile network system technology. The core network is evolving towards a converged packet framework for all services and the Internet Protocol Multimedia Subsystem (IMS) provides multimedia services over packet switched networks. As defined by 3rd Generation Partnership Projects (3GPP), in order to access the multimedia services, LTE users require authentication in the LTE network layer and the IMS service layer by using similar Authentication and Key Agreement protocol (AKA) transactions. The requirement for two AKA transactions increases authentication delay and the terminal's energy consumption. This paper proposes an Improved AKA (IAKA) authentication protocol which binds the two layer's authentication procedures by using the Internet Protocol Multimedia Private-User Identity as a link between the LTE network layer and the IMS service layer. The proposed algorithm significantly reduces the authentication process complexity. Simulations were carried out for the IAKA and the original 3GPP defined AKA authentication protocols. Results including security, performance and energy consumption are presented. The results showed that by using IAKA a reduction of up to 38% of the IMS layer authentication delay is possible, and a saving of 39.13% of the terminal's energy consumption occurs with increased security.**

***Index Terms*—4G; Authentication; IP Multimedia Subsystem (IMS); Long Term Evolution (LTE); Authentication and Key Agreement (AKA); Session Initiation Protocol (SIP); Call Session Control Function (CSCF) .**

I. Introduction

Driven by the rapid growth in mobile broadband networks and the insatiable bandwidth requirements of new applications, mobile networks are evolving towards converged Internet Protocol (IP) based 4th generation (4G) mobile broadband networks. As a major 4G technology, Long Term Evolution (LTE) aims to provide a high data rate, low latency and all IP mobile networks. Correspondingly, the core network is evolving towards a converged packet-based framework for all services. As a part of the evolved core network the IP Multimedia Subsystem (IMS) provides multimedia services (audio, video, text, image, and combinations) over Packet Switched Networks (PSN) for wired and wireless access networks.

IMS was first specified by 3GPP (3rd Generation Partnership Projects) in 2003 for the 3rd generation (3G) network, and it was extended to fixed network by TISPAN (Telecommunication and Internet converged Services and Protocols for Advanced Networking) as a subsystem of NGNs (Next Generation Networks). As defined, a subscriber has to have different identities for different access networks and core network, i.e., IMSI (International Mobile Subscriber Identity) for mobile network and IMPI (IP Multimedia Private-User Identity) for IP core network, therefore, a subscriber has to pass at least two layers authentication to access the data services. The reasons for this approache are (1) improve access-network independent, (2) ensure the security and QoS (Quality of Service) of the IP packet layer. This approach is quite necessary in the 3G network since the voice service and data services are

separated, but it becomes quite redundant in the 4G network since all of the services (voice and data) are based on IP network.

As defined by 3GPP, a LTE user has to be authenticated twice to access the service layer. The first authentication step is the LTE layer authentication which enables the user to access the LTE and packet data network by checking the user's IMSI number; and the second authentication step is the IMS service layer authentication which authenticates the user by checking the user's IMPI number to enable the user to access the services (voice and data). It was observed that the two authentication steps use the same Authentication and Key Agreement (AKA) protocol including requesting, generation and distribution of the Authentication Vectors (AV), and mutual authentication by using the received AVs which increases the system's authentication delay and the terminal's energy consumption.

In order to reduce the overhead, an Improved AKA (IAKA) authentication protocol was proposed which binds the two authentication steps by using the user's IMPI number without double execution of the AKA protocol. In the proposed IAKA, the Home Subscriber Server (HSS) generates the AV by using the IMPI number and distributes the AV to the User Equipment (UE), the Mobility Management Entity (MME), and the Proxy Call Session Control Function (PCSCF). In both of the network layer authentication and the service layer authentication, the terminal and the network entities authenticate each other by using the same AV generated by the HSS using the IMPI number.

The security, performance and energy consumption of the proposed IAKA was analyzed and the results are presented in this paper. The results demonstrate that the IAKA provides adequate authentication, supports mutual authentication, and implements confidential and integrity message protection. Furthermore, two possible security attacks are discussed in this paper: (1) replay attack; and (2) Denial of Service (DOS) attack. The proposed IAKA algorithm protects the system from the malicious replay attack and provides stronger protection against the DOS attack than the legacy 3GPP approach. An OPNET simulation model was developed to simulate the proposed IAKA algorithm and the 3GPP authentication protocols under different scenarios. The simulation results showed that the proposed IAKA algorithm simplified the authentication protocol significantly and reduced the IMS layer authentication delay by up to 38%. The terminal energy consumption used in the authentication and security activities was calculated and the results showed the proposed IAKA algorithm could save up to 81.82% in the IMS service layer authentication and save up to 39.13% in the combined two layer authentication process.

The remainder of the paper is organized as follows: Section II is the related work found in the literature; Section III provides the background technologies and the 3GPP defined AKA authentication protocols; Section IV introduces the proposed IAKA authentication protocol; Section V provides a comprehensive comparison between the proposed IAKA and the legacy 3GPP approach for security, performance and energy consumption; and Section VI provides a summary, conclusion and discussion of future work.

II. Related Work

Due to the importance of authentication research has been carried out with the aim of improving authentication efficiency [1]-[9]. Related work found in the literature focused on the 3G network [1]-[7] or network layer authentication [1], [3], [6], [8]. Ntantogian et al. carried out research focused on reducing overhead in the network and service layer for 4th generation networks [9]; however, the approach suffers from security issues.

Ntantogian et al. proposed a generic mechanism for efficient authentication in Beyond 3rd Generation (B3G) networks that attempted to reduce the execution of the 3GPP defined multi-pass authentication steps by using a security binding mechanism [9]. The proposed mechanism authenticates a user in the second and third step of multi-pass authentication by using the user's authentication credentials from the initial step. The proposed IMS service layer authentication used an (IMSI, IMPI) pair to authenticate the user without security protection between the UE and the PCSCF. However, this approach suffers the following vulnerabilities: (a) Lack of mutual authentication. In the proposed IMS layer authentication, the terminal doesn't support the server authentication. Therefore, an adversary can pretend to be the PCSCF/ICSCF (Interrogating CSCF) /SCSCF (Serving CSCF) /HSS servers to get the user's confidential information. (b) Lack of confidential and integrity protection. The proposed IMS layer authentication doesn't support the confidential and integrity protection of the messages transmitted between the Packet Data Gateway (PDG) and the PCSCF which makes it very vulnerable to the eavesdropping or man in the middle attack. (c) Vulnerable to the replay attack. It is assumed the adversary is between the PDG and the PCSCF and captures packets transmitted between the two. The adversary can initiate a replay attack by using captured packets to send a SIP Register request with the (IMPI, IMSI) pair to the server. Since the

SCSCF only checks whether the IMSI received from the terminal equals the IMSI received from the HSS, with the captured correct (IMPI, IMSI) pair, the adversary will successfully pass the authentication step and gain access to the service layer. The adversary can pretend to be the IMS servers and send the captured SIP 200 OK response to the terminal. Without the right security protection, the terminal will create a trust session with fake server and send confidential information to it. The proposed IMS layer authentication is very vulnerable to replay attacks. (d) Vulnerable to Denial Of Service (DOS) attacks. It is assumed that an adversary sends a SIP Register request with an IMPI number which belongs to a valid user to the PCSCF. The PCSCF would forward this packet to the ICSCF, the SCSCF, and the HSS. Therefore, the adversary could flood the core network and jeopardize the whole system.

III. Background

A. Architecture

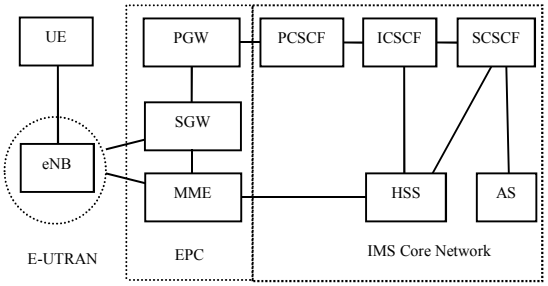

Figure1 LTE and IMS integrated system architecture

The LTE and IMS integrated system architecture is shown in Fig. 1 [10], [11] and the system component functionalities are briefly described here to provide a guide that will assist with understanding the proposed IAKA authentication protocol: (a) Evolved Universal Terrestrial Radio Access Network (E-UTRAN) consists of Evolved Node B (eNB) and provides OSI reference model layer 1 and layer 2 features; (b) MME controls the Evolved Packet System (EPS) mobility management and EPS session management; (c) The Serving Gateway (SGW) which serves as a mobility anchor routing and forwarding packets between eNBs and between eNB and Packet Data Network Gateway (PGW); (d) The PGW provides access to the external packet data network; (e) The Proxy CSCF (PCSCF) is the first contact point in the IMS; (f) The Interrogating CSCF (ICSCF) is used to hide the network's topology and its main task is to identify the relevant SCSCF; (g) The Serving CSCF (SCSCF) performs the session control services for the UE; and (h) The HSS is the central database for the whole system and contains subscription-related information.

B. LTE EPS AKA authentication

LTE EPS AKA is used to provide mutual authentication between the terminal and the network and agree on a key KASME which is stored in the MME and used to derive further keys for protection of different traffic flow types. The triggering of the LTE EPS AKA procedure could be due to tracking area updates. In most cases, the EPS AKA authentication is performed in the attach procedure when the user terminal is switched on and requests access to packet services.

The EPS AKA procedure is shown in Fig. 2 [10], [12], [13] and includes:

- E1: In order to register with the LTE network to receive services, the UE initiates an Attach Request to the MME with the IMSI number using the Non Access Stratum (NAS) protocol;
- E2: If there aren't valid AV in the MME, the MME will send Authentication Information Request (AIR) to the HSS to fetch AV by using the Diameter protocol;
- E3: Upon receipt of the request, the HSS uses the IMSI number to fetch the user's profile and generates AV (RAND, AUTN, XRES, KASME); then the HSS sends the AV back to the MME by using the Authentication Information Answer (AIA) command message.
- E4: Upon AV receipt from the HSS, the MME initiates the authentication procedure by sending an Authentication Request to the UE which contains the parameter of RAND, AUTN and KSIASME. Key Set Identifier (KSI) is used to identify KASME.
- E5: The UE checks the received authentication token AUTN first. If the AUTN can be accepted, the UE considers that the MME has passed the authentication correctly. After the successful AUTN verification, the UE will calculate the response (RES) number and send it back to the MME by using a Authentication Response message;
- E6: The MME checks whether the RES from the UE matches the XRES from the HSS, if they match, the MME considers that the authentication and key agreement exchange has successfully completed and sends back Attach Accept to the UE.

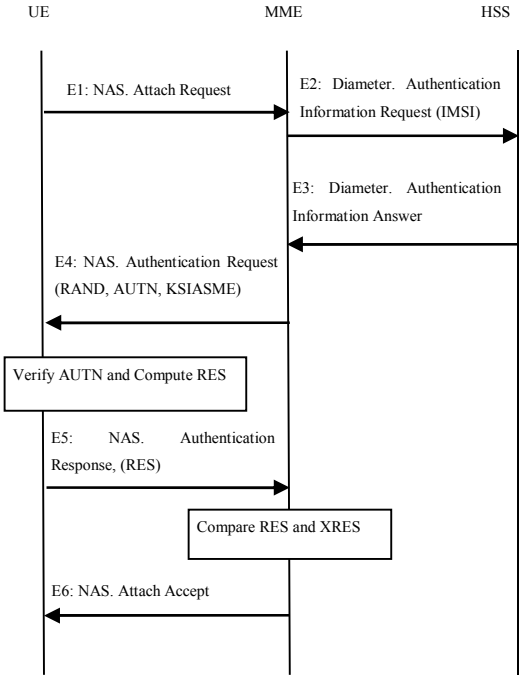

Figure 2. LTE EPS AKA autentication procedure

C. IMS AKA authentication

After a successful LTE network layer authentication and the creation of the EPS bearer, the terminal is eligible to access the packet network. In order to access the IMS services, the user has to complete IMS layer authentication and the procedures are shown in Fig. 3 [11], [12], [14].

- I1: After the successful creation of the EPS bearer, the UE sends a REGISTER request to the PCSCF using the SIP protocol. The SIP REGISTER request incorporates parameters including IMPI and security-setup line in which the security–setup line is used to create security associations (SA) with PCSCF.

- I2, I3: The PCSCF stores the security-setup parameters; removes the security-setup line from the packet and forwards it to the ICSCF. The ICSCF fetches the user and the SCSCF information from the HSS, locates the address of the next hop SCSCF and forwards the packet to the SCSCF.

- I4: After receiving the SIP REGISTER request, the SCSCF checks whether it has a previously stored AV. If it hasn't, the SCSCF sends a Multimedia Authentication Request to the HSS with the IMPI using the Diameter protocol.

- I5: The HSS uses the IMPI number to fetch the user's information and generate AVs (RAND, XRES, CK, IK, AUTN); then sends the AVs back to the SCSCF.

- I6, I7: The SCSCF stores the AVs, selects the next AV(i) and sends it back to the PCSCF through the ICSCF.

- I8: The PCSCF keeps the received AV, selects parameters for creating the SAs and sends the packet with parameters including security-setup line, IMPI, RAND, and AUTN to the UE.

- I9: After receiving the SIP REGISTER response, the UE checks the validity of the AUTN. If this is a valid AUTN, the UE generates RES, calculates the CK and the IK, and creates the IPSec security association to protect the following SIP messages, and then it sends a SIP REGISTER message to the PCSCF with parameters including security-setup and RES in the security channel.

- I10, I11: After the received packet passes PCSCF validity checking, the PCSCF sends the SIP REGISTER packet to the SCSCF through the ICSCF with RES and integrity-protection=successful.

- I12: The SCSCF compares the received RES from the UE with the XRES received from the HSS. If RES and XRES match, the SCSCF considers the AKA is successfully completed and sends a Server Assignment Request to the HSS.

- I13, I14, I15, and I16: After receiving a response from the HSS, the SCSCF sends a SIP REGISTER response 200 OK to the UE through the ICSCF and the PCSCF.

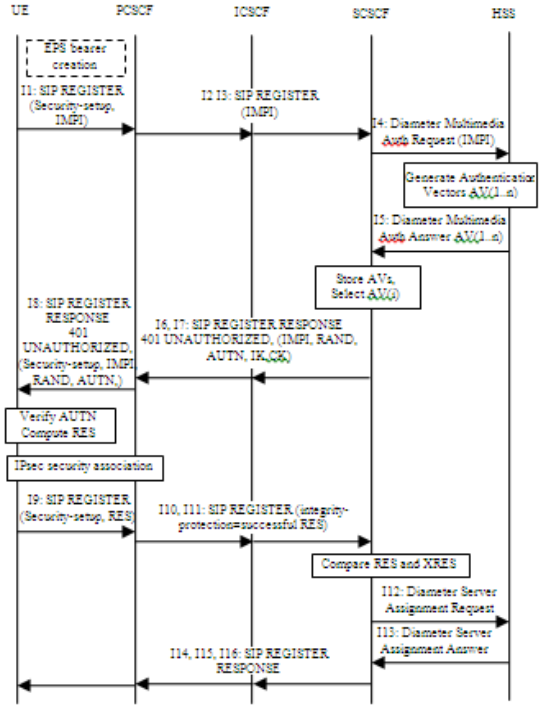

Figure 3. IMS AKA autentication procedure

D. IMS AKA authentication requirement

It is necessary to apply IMS AKA authentication in a LTE IMS system, although the AKA protocol used in the two authentication steps is identical. If the system just does LTE network layer authentication and bypasses the IMS service layer authentication it is possible to carry out fraudulent IMS usage which was analyzed by Lin et al. [2].

IV. Improved One-pass Authentication

A. Outline

The proposed IAKA authentication protocol binds the two AKA steps by using the unified IMPI number. As described in [11], IMPI is used to do registration, authorization, administration and accounting and its level is higher than the IMSI number, hence, it could be used as an index to generate AV in both the network and service layers. The AV generated by using the same IMPI number would be the same in the two layers and therefore the AV could be synchronized from the network layer to the service layer before the IMS layer authentication which could avoid the AKA protocol execution in the IMS layer. Furthermore, an IMS four layer key hierarchy was proposed in IAKA protocol to improve security and maintain LTE and IMS consistency. The proposed IAKA protocol significantly improves performance and security.

B. The proposed IMS key hierarchy

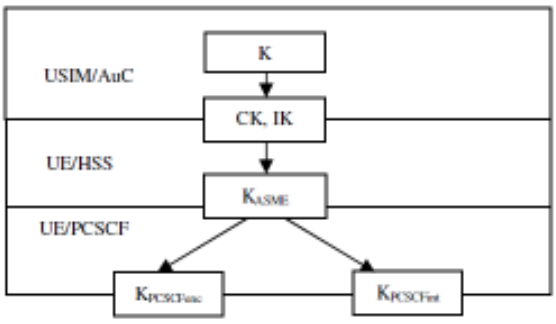

Figure4 proposed IMS key hierarchy

In order to increase security and maintain LTE and IMS consistency, a layered IMS key hierarchy was proposed and is shown in Fig. 4.

K is the pre-shared key between the UE and the HSS which is used to generate Cipher Key (CK) and Integrity Key (IK) and is kept in the Universal Subscriber Identity Module (USIM) of the UE and the Authentication Center (AuC) of the HSS. KASME is derived from CK and IK by HSS and is used to derive further keys for protection of different message flow types.

$K_{PCSCFenc}$ is derived by the UE and the PCSCF from K_{ASME}, and used to protect the traffic between the UE and the PCSCF with a particular encryption algorithm. And $K_{PCSCFint}$ is derived by the UE and the PCSCF from K_{ASME}, and is used to protect the traffic between the UE and the PCSCF with a particular integrity algorithm.

$$K_{PCSCFenc} = KDF\ (K_{ASME}, S) \quad (1)$$

$$K_{PCSCFint} = KDF\ (K_{ASME}, S) \quad (2)$$

Where Key Derivation Function (KDF) is described in [15] and S is the input string to the KDF which keeps the same definition as [13] except for P0. P0 is the algorithm type distinguisher which needs to be extended to support the key derivation of $K_{PCSCFenc}$ and $K_{PCSCFint}$ as:

- Algorithm distinguisher = PCSCF_enc_alg and the value = 0x06.
- Algorithm distinguisher = PCSCF_int_alg and the value = 0x07.

C. The improved EPS AKA authentication

In the improved LTE EPS AKA authentication, instead of sending the IMSI number, the UE sends the IMPI number to the HSS to obtain AV. Compared to the 3GPP defined EPS AKA authentication described in Fig. 2, the proposed EPS authentication has the following improvements:

- In step E1: In order to access the LTE network, the UE sends an Attach Request to the MME with IMSI=imsi and IMPI=impi. IMPI=imsi if there is no impi available;
- In step E2: If there isn't a valid AV in the MME, the MME will send Authentication Information Request (AIR) to the HSS with the IMPI number;
- In step E3: Upon receipt of the request, the HSS uses the IMPI number as an index to fetch the user's profile and generate AV; the HSS then sends AV back to the MME.

After a successful EPS authentication, both the UE and the MME have passed authentication and have the same KASME which is used to derive more keys for different security protection purposes.

D. The improved IMS AKA authentication

After a successful EPS AKA authentication and activation of the EPS bearer, the UE initiates a REGISTER request to login to the IMS network.

The procedure is shown in Fig. 5 in which step II1 and II4 are used to negotiate SA parameters; II2 and II3 are used to synchronize AV. After II4, all of the SIP messages are integrity and confidentiality protected by using the IPSec Encapsulating Security Payload (ESP) protocol [18].

- II1: After activation of the EPS bearer, the UE sends a SIP REGISTER request to the PCSCF to negotiate the parameters to be used to build SAs with the IMPI number, Global Unique Temporary Identity (GUTI) number and security-setup line. The security-setup line includes the Security Parameters Index (SPI) values, the protected ports selected by the UE and the integrity and encryption algorithm list that the UE supports.

- II2: Upon receipt of II1, the PCSCF derives the MME address from the GUTI and sends a Context Request to the MME to fetch the AV with IMPI and GUTI. The protocol between the PCSCF and the MME is the GTP (General packet radio services Tunneling Protocol) [16] and the security is defined in [17].

- II3: The MME obtains the user's AV by using the GUTI number and responds to the PCSCF.

- II4: Upon receipt of the AV from the MME, the PCSCF chooses the SPIs, port numbers, supported integrity and encryption algorithms which are used to construct the security-setup line; and sends the security-setup line back to the UE by using SIP REGISTER response 401 Unauthorised. Meanwhile, the PCSCF derives the $K_{PCSCFenc}$ and the $K_{PCSCFint}$ for further message protection.

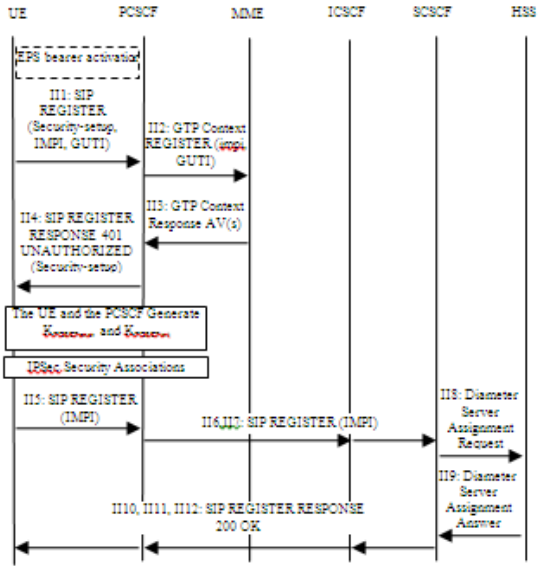

Figure 5. The improved IMS AKA autentication procedure

- II5: Upon receipt at II4, the UE derives the $K_{PCSCFenc}$ and the $K_{PCSCFint}$ and builds the SAs by using $K_{PCSCFenc}$, $K_{PCSCFint}$, SPIs, ports and the integrity and encryption algorithms. The UE sends the SIP REGISTER request to the PCSCF using the IPSec SA.

- II6: The PCSCF checks the received SIP packet by decrypting and calculating the integrity code. If there is a valid integrity code, the PCSCF forwards the packet to the ICSCF.

- II7: The ICSCF fetches the user and the SCSCF information from the HSS, locates the next hop SCSCF address, and sends the packet to the SCSCF.

- II8: The SCSCF sends a Server Assignment Request to the HSS with IMPI=impi using the Diameter protocol.

- II9: The HSS fetches the user's profile and assigns the user's status using the IMPI number, and then the HSS sends the Server Assignment Answer back to the SCSCF.

- II10 and II11: Upon receipt the II9, the SCSCF responds the ICSCF by using SIP 200 OK, and the ICSCF forwards the message to the PCSCF.

- II12: The PCSCF protects the message using IPSec and sends the message to the UE. The UE checks the packet validity. If the packet is valid, the UE considers that it is working with a legal network entity and continues communicating with it.

V. Analysis

A. Security Analysis

This section describes the proposed IAKA authentication protocol security mechanism, and shows that the proposed IAKA protocol carries out authentication correctly, supports mutual authentication, confidential and integrity message protection, and prevents possible malicious attacks.

1) Authentication Accuracy

This section describes the proposed IAKA authentication process in the network and service layers. The UE authenticates with network entities in the network layer and it is assumed there are a mobile terminal with a pre-shared key K_{ue} and a network entity with the key K_{hss}. The network entity initiates a NAS Authentication Request to the UE with the random number RAND and authentication code AUTN as shown in (3).

$$AUTN = SQN \oplus AK \,||\, AMF \,||\, MAC \quad (3)$$

Where SQN is a sequence number, AK is anonymity key, AMF is authentication

management field and MAC is the message authentication code described in (4).

$$MAC = f1_K (SQN||RAND||AMF) \quad (4)$$

Where f1 is the message authentication function and K is the key used. After receiving the Authentication Request with the RAND and the AUTN, the terminal computes the AK and the SQN first, then calculates XMAC by using (4). The terminal checks whether the received MAC calculated by the server is equal to the XMAC calculated by the terminal. If XMAC = MAC, the network entity is accepted and continues to communicate with the terminal. For a server with an invalid key identified by the IMPI number $K_{ue} \neq K_{hss}$, the calculated MAC won't be equal to the XMAC calculated by the terminal, and the terminal rejects the server connection and ceases the server connection and communication process.

After receiving the RES from the UE, the MME checks the RES number. If the RES is equal to the XRES received from the HSS, the terminal authentication is accepted and the terminal can access the network. XRES and RES are calculated by using (5).

$$XRES = f2_K (RAND) \quad (5)$$

Where f2 is a message authentication function and K is the key. If it is an invalid terminal without a valid key so that that $K_{ue} \neq K_{hss}$, the RES wouldn't be equal to XRES and the MME would reject the terminal attach attempt. Therefore, only the terminal with the correct key identified by the same IMPI number can pass authentication and successfully login to the LTE network.

In the IMS service layer, the PCSCF performs authentication for the network by checking the Integrity Checking Value (ICV), described in (6), received in SIP Register Request using the IPSec ESP protocol.

$$ICV = fint_K (M) \quad (6)$$

Where fint is the function used to generate the integrity code, M is the ESP packet minus the ICV and K is the key $K_{PCSCFint}$. A valid UE, after a successful network layer authentication, can derive an intermediate key K_{ASME} which is equal to the K_{ASME} stored in the PCSCF and identified by the IMPI number. Using (2), both the terminal and the PCSCF derive $K_{PCSCFint}$ by using the same K_{ASME}, hence, the $KP_{CSCFintUE} = K_{PCSCFinthss}$. The PCSCF, after receiving the ESP packet, retrieves the ICV from the ESP packet and calculates XICV by using (6), and if the terminal and the PCSCF use the same key and the M wasn't changed during transmission, the ICV would be equal to XICV. Without a valid pre-shared key to generate the correct K_{ASME} and $K_{PCSCFint}$, ICV ≠ XICV, and the PCSCF rejects the register request.

Finally, the UE authenticates the network in the IMS service layer by checking the ICV received in the SIP Register Response encapsulated by the IPSec ESP protocol packet. Using (6), when the UE and the PCSCF have the same key and the input string M, the received ICV will be equal to the calculated XICV. Any server without a correct key K_{ASME} can't generate a valid ICV to pass authentication at the terminal, and the communication session is rejected.

2) Mutual Authentication

As described, the network and service layers authentication using the proposed IAKA protocol support mutual authentication. Only the terminals and the network entities with the correct keys and a valid IMPI number authenticate successfully and build a trust relationship with each other. Without the valid pre-shared key identified by the IMPI number, adversaries couldn't pretend to be a legal terminal to access network and the service resources. Also, adversaries couldn't pretend to be a valid network entity to get user's private information.

3) Confidential and integrity protection

This section describes the confidential and integrity protection of the proposed IAKA IMS service layer authentication. The network layer is not discussed in this section because the confidential and integrity protection of the IAKA network layer is the same as the 3GPP definition.

The proposed IAKA IMS layer authentication protocol creates a pair of IPSec security associations to protect the information transmitted between the terminal and the PCSCF. Before transmission, the sender encrypts the data by using the symmetric encryption algorithm and the key $K_{PCSCFenc}$. Only a receiver with the same key $K_{PCSCFenc}$ can decrypt the message and obtain the correct plain text.

The message between the terminal and the PCSCF are also integrity protected in the IPSec SAs. The receiver checks the message's integrity by using the ICV number. If the received ICV doesn't equal to the calculated XICV, the received message is incomplete and will be discarded.

4) Possible attacks

This section analyzed how the proposed IAKA prevented the possible replay attack and DOS attack.

A replay attack can't be achieved in the network and service layers because in the network layer, if an adversary was eavesdropping on the

conversation between the terminal and the MME and captured the NAS Authentication Request and NAS Authentication Response (RES), it might initiate a replay attack by sending the Authentication Request (RAND, AUTN and KSI_{ASME}) packet to the terminal; however, the adversary couldn't pass a SQN number check which is a part of the AUTN. As defined in [13], the SQN should be different in the new session and the terminal should check the SQN number validity first. If the received SQN is the same as the previously accepted SQN number, the terminal would send an error response to the fake server. The adversary might do a replay attack by sending the Authentication Response packet to the server, however, a RES number check would identify this as a fake message. Using (5), RES is calculated by using the HSS randomly generated RAND which is different for each session. Therefore in a new session, the captured packet with the old RES number would be identified as a fake message and access to the services would be prevented. In the service layer, adversaries can't get the user or the server's information by sending the captured packets because the packets are protected by using IPSec ESP which includes a monotonically increasing sequence number. After receiving a packet, the receiver verifies the sequence number first to make sure this is not a duplicated packet; hence, the captured packet with the old sequence number would not be accepted. Therefore, the proposed IAKA authentication protocol can protect the system from the malicious replay attack.

The proposed IAKA protocol has higher security against a DOS attack than the legacy 3GPP defined authentication protocol. With the legacy 3GPP defined authentication protocol the IMS core system is vulnerable to a DOS attack. Assume the adversary floods the PCSCF/ICSCF/SCSCF/HSS by sending SIP REGISTER requests with a valid IMPI number which belongs to a valid subscriber X. The PCSCF would forward the requests to the core network servers ICSCF/SCSCF/HSS. However, for IAKA authentication, the flood messages can only reach the PCSCF and the MME, leaving the core network fully protected. If the subscriber X hasn't logged into the LTE network and the PCSCF is not able to retrieve the subscriber's AV from the MME, the PCSCF considers this to be an invalid terminal and will reject the Register request. If the subscriber X has already registered with the LTE network, the adversary is required to transmit SIP Register requests in the IPSec security associations, and without the valid key, the messages transmitted to the PCSCF will not be validated and forwarded to the next hop. Therefore the adversaries cannot jeopardize the core ICSCF / SCSCF / HSS servers.

B. Performance Analysis

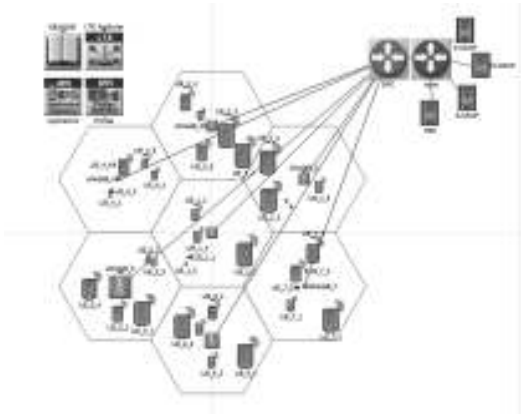

Figure 6. The simulated LTE-IMS integrated system architecture

In order to analyze system performance, a simulation of the LTE IMS integrated system was developed and the system architecture is shown in Fig. 6. The IMS layer authentication protocols were simulated based on [18]-[22]. The EPS AKA wasn't simulated since it is quite similar in the two approaches.

As shown in Fig. 6, the LTE-IMS system model includes the following entities and functions: (a) 7 cells with 5 mobile terminals in each which are connected to the eNB located in the same cell. The terminals support SIP registration, the generation of AVs by using the Advanced Encryption Standard (AES) algorithm, key derivation by using the authentication algorithm HMAC-SHA-256, and the IPSec ESP protocol by using AES and the authentication algorithm HMAC-SHA-1-96; (b) 7 eNBs which are connected to the Evolved Packet Core (EPC) server; (c) An EPC server which is the integration of the MME, SGW, and PGW; (d) The PCSCF, ICSCF servers support SIP registration and SIP packet routing. The PCSCF also supports key derivation, and the IPSec ESP protocol; (e) The SCSCF supports SIP and the Diameter protocol and protocol translation; (f) The HSS server which supports the Diameter protocol, AV generation and key derivation; and (g) Two different authentication protocols were implemented in the simulation, the legacy 3GPP IMS AKA authentication protocol developed as shown in Fig. 3, and the proposed improved IMS AKA authentication protocol developed as shown in Fig. 5. The following activities were not implemented and were simulated as a network delay:

- The request processing on the PCSCF, ICSCF, SCSCF and HSS which should be supported in the two approaches and is currently simulated as $DELAY_{processing}$.
- The ICSCF locating the next hop SCSCF address by checking the information from the HSS

which should be supported in the two approaches and is currently simulated as $DELAY_{ICSCFHSS}$.

- The database operations on the HSS such as searching the user's profile by using IMPI number should be supported in the two approaches and is currently simulated as $DELAY_{HSS}$.
- The PCSCF obtaining the AVs from the MME should be supported in the IAKA authentication protocol and is currently simulated as $DELAY_{AVretrieval}$.

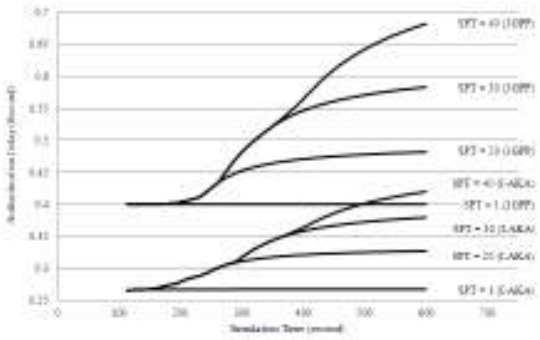

Figure7. IMS layer Authentication Delay under fixed network delay configuration

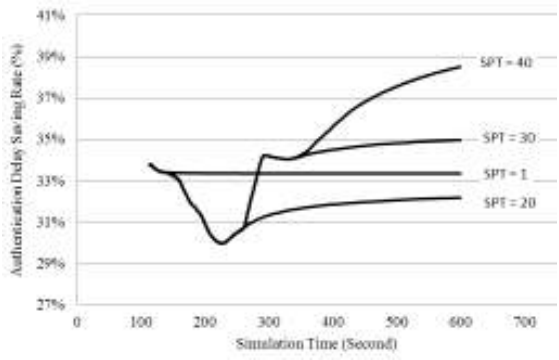

Figure 8. IMS layer Authentication Delay Saving Rate under fixed network delay configuration

Two configuration types were utilized for the assumed delays including a fixed network delay configuration and a varying network delay configuration.

The fixed network delay configuration, after starting the simulation, includes all of the terminals initiating SIP register request repeatedly and simultaneously. Simultaneous rate Per Terminal (SPT) is used to describe the number of SIP registration procedures one terminal initiated simultaneously and the effect on system load. The statistics for the IMS layer authentication delay was collected from the UE and it is the time taken between sending the first SIP Register request and receiving the SIP Register response 200 OK. Referring to [23], it was assumed $DELAY_{processing}$ = 0.025s, $DELAY_{HSS}$ = 0.055s, $DELAY_{ICSCFHSS}$ = 0.055s; and $DELAY_{AVretrieval}$ = 0.025s. Four different system loads were chosen with SPT = 1, SPT = 20, SPT = 30, and SPT = 40. Under the different SPT rates, the authentication delay of the 3GPP and the IAKA approaches are shown in Fig. 7, and the IMS layer Authentication Delay Saving Rate (ADSR) is shown in Fig. 8.

It was identified that in non-loaded condition, the proposed IAKA protocol could save 33.34% of IMS layer authentication delay. When SPT = 1, no background system load is assumed. Under such conditions, the authentication delay is very stable. The average IMS layer authentication delay of the 3GPP defined AKA protocol is 0.4s and the average delay of the proposed IAKA is 0.267s. Therefore, compared to the legacy 3GPP defined AKA protocol the proposed IAKA could save 33.34% of the IMS layer authentication delay in a non-loaded environment.

The authentication delay increased with increased system load as expected from Fig. 7. The effect of the average authentication delay under different system loads during the simulation period 500 to 599 seconds is shown in Table I.

TABLE I AVERAGE SIMULATION RESULTS DURING SIMULATION PERIOD (500 SECOND–599 SECOND)

SPT	Authentication Delay (second)		ASDR (%)	System load of eNB/EPC (requests/ responses per second)	
	3GPP	IAKA		3GPP	IAKA
20	0.4798	0.3257	32.12%	5664	8400
30	0.5775	0.376	34.89%	7006	10850
40	0.6641	0.4112	38.08%	7168	12201

In the loaded network condition, as system load increases the ADSR increases as expected from Fig. 8. As shown in Table I, the average ASDR is 38.08% when SPT = 40 which is much higher than the ASDR for SPT = 20 and SPT = 30.

Under low system load (SPT = 20), the IMS authentication delay of the IAKA protocol increased faster than the 3GPP AKA protocol. The main reason for this outcome was found at the eNB / EPC which in the IAKA scenario processed more requests / responses during the same period which is shown in Table I. The wireless network is more sensitive than the wired network in low system load conditions. Therefore, the ASDR under low system load (SPT =20) is lower than the ASDR without system load (SPT = 1).

Under a high system load, the IMS authentication delay of the 3GPP protocol increased faster than the IAKA protocol. Although the wireless network IAKA approach processed more packets, the entire authentication steps saved in the proposed IAKA authentication protocol occurred in the wired network. Under certain system loads, the wired network is very sensitive to

the packets processed. Therefore, the wired part of the authentication delay of the 3GPP approach increased faster than the proposed IAKA authentication approach and the ADSR of SPT =30 and SPT = 40 is higher than the ADSR without system load (SPT = 1).

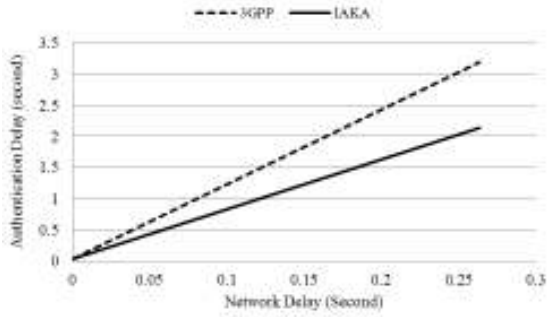

Figure 9. IMS layer Authentication Delay under increased network delay configuration

For the varying network delay configuration the average network delay was used. It was assumed that $DELAY_{processing} = DELAY_{HSS} = DELAY_{ICSCFHSS} = DELAY_{AVretrival}$ = averageDelay. The average delay started from 1ms and the terminals carried out SIP registration procedures repeatedly and each procedure has a 1ms increment. In the non-loaded condition (SPT = 1), the statistics for the authentication delay and ADSR were collected from the UE and the results are shown in Fig. 9 and Fig. 10. Fig. 9 and Fig. 10 show that the authentication delay and the ADSR increased with increasing network delay, and that under the non-loaded network environment, with increasing the network delay, the proposed IAKA authentication protocol could save up to 33% of the authentication delay.

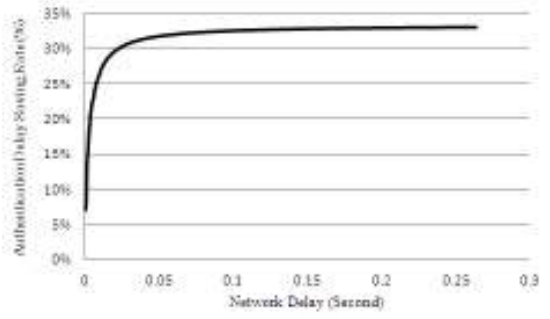

Figure 10. IMS layer Authentication Delay Saving Rate under increased network delay configuration

C. Energy Cost Analysis

This section provides an energy cost analysis including calculation of the energy consumed by the terminal's security activities during the authentication period, and a comparison between the legacy 3GPP approach and the proposed IAKA approach. It was found that the proposed IAKA protocol could reduce terminal energy consumption significantly.

The terminal's energy cost E for the authentication process is given in (7) and is made up of two components including the execution of the EPS AKA, and the execution of the IMS AKA.

$$E = E_{EPS\text{-}AKA} + E_{IMS\text{-}AKA} \quad (7)$$

Where E denotes the energy consumption, $E_{EPS\text{-}AKA}$ denotes the energy consumption that occurs in the EPS network layer which is given in (8), and $E_{IMS\text{-}AKA}$ denotes the energy consumption that occurs in the IMS service layer. The energy consumed in LTE network layer is the same as in the 3GPP authentication approach and the proposed IAKA authentication approach and the security activities include: (a) checking the AUTN number which is made up of generating AK and MAC; (b) generating the RES number; and (c) generating CK, IK and deriving K_{ASME}.

$$E_{EPS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} + E_{KASME} \quad (8)$$

The terminal's energy consumption in the IMS layer for the legacy 3GPP authentication protocol derived in (9) includes generation of the AK, MAC, RES, CK, and IK. The energy consumption for the proposed IAKA protocol derived in (10) includes key derivation of $K_{PCSCFint}$, and $K_{PCSCFenc}$. The energy consumption of the IPSec ESP protocol wasn't considered because IPSec is mainly used for message transmission and it is the same in the two approaches.

$$E_{3GPP\text{-}IMS\text{-}AKA} = E_{AK} + E_{MAC} + E_{RES} + E_{CK} + E_{IK} \quad (9)$$

$$E_{IAKA\text{-}IMS\text{-}AKA} = E_{KPCSCFENC} + E_{KPCSCFINT} \quad (10)$$

Using the AV generation functions f1, f2, f3, f4, f5 defined in [20] to generate MAC, RES, CK, IK, and AK, the encryption energy consumption is considered. The combination bitwise exclusive-OR and bitwise rotation operations weren't considered for these operations won't affect the results significantly. Choosing AES as the kernel algorithm, refer to [24], the AES algorithm energy consumption EAES is made up of the energy consumption in the key setup phase $E_{KEY\text{-}SETUP}$ and in the encryption/decryption phase $E_{ENC/DEC}$. $E_{ENC/DEC} = EPB_{AES} * n$ where EPB_{AES} is energy-per-byte using the AES algorithm and n is the length of the input string. Although MAC, RES, CK, IK, and AK have different lengths, the values are generated by encrypting 16 byte data blocks which are executed 3 times. Therefore the energy consumption is shown in (11).

$$E_{AK} = E_{MAC} = E_{RES} = E_{CK} = E_{IK} = 3 * E_{AES}\ (16) = 3 * (EK_{EY\text{-}SETUP} + EPB_{AES} * 16) \quad (11)$$

To derive K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$, as shown in (12), the HMAC-SHA-256 algorithm is chosen and the energy consumption E_{HMAC} (n) =

EPB_{HMAC} * n. The length of the data block size used in this algorithm is 32 bytes.

$$E_{KASME} = E_{KPCSCFENC} = E_{KPCSCFINT} = E_{HMAC}\ (32) = EPB_{HMAC} * 32 \quad (12)$$

Using the research results found in [24], it is assumed $E_{KEY\text{-}SETUP}$ = 7.87 μJ, EPB_{AES} = 1.21 μJ, and EPB_{HMAC} = 1.16 μJ. Referring to (11) and (12), the energy consumption to generate MAC, RES, CK, IK, and AK is equal to 81.69 μJ; and the energy consumption to generate K_{ASME}, $K_{PCSCFint}$, and $K_{PCSCFenc}$ is equal to 37.12 μJ. Put these results into (8), (9) and (10), the following results were found: (a) $E_{EPS\text{-}AKA}$ = 445.57 μJ, (b) $E_{3GPP\text{-}IMS\text{-}AKA}$ = 408.45 μJ, and (c) $E_{IAKA\text{-}IMS\text{-}AKA}$ = 74.24 μJ. Using (13) and (14) to calculate Energy Saving Rate (ESR), and it was found that $ESR_{IAKA\text{-}IMS\text{-}AKA}$ = 81.82%, and ESR_{IAKA} = 39.13%. The results identified that through the execution of the proposed IAKA authentication protocol, the proposed IAKA protocol could save up to 81.82% of the terminal energy consumption for security activities including IMS service layer authentication; and save 39.13% for the combination of the network and service layers.

$$ESR_{IAKA\text{-}IMS\text{-}AKA} = (E_{3GPP\text{-}IMS\text{-}AKA} - E_{IAKA\text{-}IMS\text{-}AKA}) / E_{3GPP\text{-}IMS\text{-}AKA} * 100\% \quad (13)$$

$$ESR_{IAKA} = (E_{3GPP} - E_{IAKA}) / E_{3GPP} * 100\% \quad (14)$$

VI. Conclusion

This paper presented an improved IMS authentication protocol for the 4th generation LTE network which combines the authentication for the network and service layers by using a binding of the IMSI and IMPI numbers. Compared to the legacy 3GPP defined AKA protocol, the proposed IAKA protocol simplified the authentication steps, reduced authentication delay, reduced the user terminal's energy consumption significantly and provided advanced security. Future research work will focus on developing a generic improved IMS authentication protocol for next generation access networks. Currently work is being carried out in further development of the model and removing necessary assumptions that were made during this study.

VII. References

[1] Y. Lin, and Y. Chen, "Reducing Authentication Signaling Traffic in Third-Generation Mobile Network," IEEE Trans. Wireless Communications, vol. 2, pp. 493–501, May 2003.

[2] Y. Lin, M. Chang, M. Hsu, and L. Wu, "One-Pass GPRS and IMS Authentication Procedure for UMTS," IEEE J. Selected Areas in Communications, Vol. 23, pp. 1233-1239, Jun. 2005.

[3] Y. Zhang, and M. Fujise, M, "An Improvement for Authentication Protocol in Third-Generation Wireless Networks," IEEE Trans. Wireless Communications, vol. 5, pp. 2348–2352, Sep. 2006.

[4] C. Huang and J. Li, "Efficient and Provably Secure IP Multimedia Subsystem Authentication for UMTS," The Computer Journal, Vol. 50, pp. 739–757, Oct. 2007.

[5] C. Huang and J. Li, "Reducing Signaling Traffic for the Authentication and Key Agreement Procedure in an IP Multimedia Subsystem", Wireless Personal Communications, Vol. 51, pp. 95-107, 2009.

[6] C. Ntantogian and C. Xenakis, "One-pass EAP-AKA Authentication in 3G-WLAN Integrated Networks," Wireless Personal Communications, Vol. 48, pp. 569-584, 2009.

[7] L. Gu and M. A. Gregory, "Improved One-Pass IP Multimedia Subsystem Authentication for UMTS", International Conf. on Information Networking, Kuala Lumpur Malaysia, Jan 2011.

[8] Y. Deng, H. Fu, X. Xie, J. Zhou, and Y. Zhang, "A Novel 3GPP SAE Authentication and Key Agreement Protocol", in Proc. IEEE International Conf. on Network Infrastructure and Digital Content, 2009, pp. 557–561.

[9] C. Ntantogian, C. Xenakis, and I. Stavrakakis, "A generic mechanism for efficient authentication in B3G networks," Computers and Security, Vol. 29, pp. 460-475, 2010.

[10] 3GPP TS 23.401 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 10)", Jun. 2010.

[11] 3GPP TS 23.228 V10.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 10)", Mar. 2010.

[12] 3GPP TS 33.102 V9.1.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 9)", Dec. 2009.

[13] 3GPP TS 33.401 V9.4.0 "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 9)", Jun. 2010.

[14] 3GPP TS 33.203 V9.3.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 9)", Dec. 2009.

[15] 3GPP TS 33.220 V9.3.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 9)", Jun. 2010.

[16] 3GPP TS 29.274 V10.1.0, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 (Release 10)", Dec 2010

[17] 3GPP TS 33.210 V10.0.0, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Network Domain Security (NDS); IP network layer security (Release 10)”, Oct. 2010.
[18] S. Kent and R. Atkinson, “IP Encapsulating Security Payload (ESP)”, RFC 2406, Nov. 1998
[19] Opnet Modeller v16, (2011, Jun. 20), Available: http://www.opnet.com/
[20] 3GPP TS 35.206 V9.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 9)", Dec 2009
[21] A. Vazquez, and J. Fernandez, (2005, Oct. 13) "SIP-IMS Model for OPNET Modeler," Available: http://www.dit.upm.es/asignaturas/opnet/SIP-IMSmodel.pdf
[22] D. Eastlake, and T. Hansen, (2011, Feb. 15). “US Secure Hash Algorithms (SHA and SHA based HMAC and HKDF)”, Available: http://tools.ietf.org/html/draft-eastlake-sha2b-07
[23] G. Foster, M.I. Pous, D. Pesch, A. Sesmun, and V. Kenneally, “Performance Estimation of Efficient UMTS Packet Voice Call Control,” Proc. IEEE Vehicular Technology Conf., vol. 3, pp. 1447-1451, Sep. 2002
[24] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, “A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols,” IEEE Trans. on Mobile Computing, vol. 5, pp. 128-143, Mar. 2006

Lili. Gu received her Bachelor of Engineering from China Textile University of Computer Application, Shanghai, China in 1999. She is a Master of Engineering candidate in the School of Electrical and Computer Engineering, RMIT University, Australia. Lili Gu is a tutor at RMIT University and a Network Engineer with Nextgen Networks, Australia. Research interests include wireless networks and information security.

Mark A. Gregory (M’82–SM’06) became a Member (M) of IEEE in 1982, and a Senior Member (SM) in 2006. Mark A Gregory was born in Melbourne, Australia and received a PhD and a Master of Engineering from RMIT University, Melbourne, Australia in 2008 and 1992 respectively, and a Bachelor of Engineering (Honors) from University of New South Wales, Sydney, Australia in 1984.

He is a Senior Lecturer in the School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia. Research interests include fiber optic network design and operation, wireless networks, security, privacy and technical risk.

Dr Gregory is a Fellow of the Institute of Engineers Australia, has reviewed journal papers for the IEEE Engineering Management Society and is an associated editor of the Australasian Journal of Engineering Education.