Relaxing Trust Requirement in 3GPP Mobile Systems for Improved Subscriber Identity Privacy

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Hiten Choudhury

Registration No. 006 of 2013

Department of Computer Science and Engineering

School of Engineering

Tezpur University

May 2013

Abstract

With the increase in availability and affordability of a range of mobile services, mobile devices are becoming an integral part of an individual. However, threats like location tracking and comprehensive profiling, where data about movement, usage, etc., of a subscriber is collected and linked to his/her identity to explore various attacks, have also emerged. Thus, identity privacy in mobile systems has become an important security issue.

In today's context, when mobile operators strive to provide wider coverage, roaming agreements/pacts with third party operators to provide service in a location where an operator has not set up its own infrastructure is a common practice. Roaming allows a subscriber of one operator to use the access service of another operator when inside the latter's coverage area. 3GPP, which is a collaboration of reputed organisations having the credit of developing some popular and market winning mobile systems in recent times, has adopted a trust model for roaming that requires an operator and its subscribers to have full trust on the third party operators. As a consequence, the authentication-and-key-agreement protocols used for access security in mobile systems developed by 3GPP cannot guarantee identity privacy to its roaming subscribers from visited/serving networks that belong to third party operators. Moreover, this necessitates the operators to set up prior trust relationships through elaborate roaming agreements with the third party operators, for providing secured roaming services including identity privacy to the subscribers. This requirement limits the ease and span of extending services beyond an operator's own circle/zone. Further, the need to trust the third party operators makes identity privacy of the subscriber vulnerable to eavesdroppers and fake serving networks. Thus, there is need for a paradigm shift, such that the requirement of trust on third party operators for roaming is relaxed or even entirely eliminated.

The study reported in this thesis envisages improvement in identity privacy of the subscriber of a 3GPP specified mobile system like UMTS, LTE, 3GPP-WLAN and Non3GPP-EPS while relaxing trust requirement for roaming between operators. Towards this, we propose a new trust model that eliminates the limi-

tations of the existing trust model and is more flexible compared to the existing trust model. We also devise a security extension based on our proposed trust model that can be used to achieve relaxed trust requirement in mobile systems developed by 3GPP for improved subscriber identity privacy. Contrary to several recent proposals in this area, our solution can be adopted as an extension to the existing authentication-and-key-agreement protocols used in mobile systems developed by 3GPP. Moreover, it can be implemented at the operator's level without needing any modification in the intermediary networks. The robustness, correctness and effectiveness of the security extension are established using various methods like statistical analysis, formal analysis, computational cost analysis, complexity analysis, space overhead analysis, communication overhead analysis and security analysis.

Keywords — trust, security, identity privacy, UMTS, LTE, EPS, authentication, interworking, roaming

Declaration

I, Hiten Choudhury, hereby declare that the thesis entitled "*Relaxing Trust Requirement in 3GPP Mobile Systems for Improved Subscriber Identity Privacy*" submitted to the Department of Computer Science and Engineering under the School of Engineering, Tezpur University, in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, is based on bona fide work carried out by me. The results embodied in this thesis have not been submitted in part or in full, to any other university or institute for award of any degree or diploma.

(Hiten Choudhury)

Tezpur University

Certificate of the Supervisor

This is to certify that the thesis titled "*Relaxing Trust Requirement in 3GPP Mobile Systems for Improved Subscriber Identity Privacy*" submitted to Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment of the award of the degree of Doctor of Philosophy in Computer Science and Engineering is a record of research work carried out by Mr. Hiten Choudhury under my supervision and guidance.

All helps received by him from various sources have been duly acknowledged.

No part of this thesis has been submitted elsewhere for award of any other degree.

Signature of Research Supervisor

(Dilip Kr. Saikia)

Designation: Professor

Department: Computer Science and Engineering

School: Engineering

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Dilip kr. Saikia for accepting me as his student. There are no words that can adequately express my gratitude for the technical direction and support, which I have been so fortunate to receive from him. He has been a source of inspiration and from him I have learned much more than my area of research.

I would also like to thank my research collaborator Dr. Basab Roychouhury for his technical guidance, insightful comments and suggestions. He has been instrumental in improving the quality of my research work in various ways. I could learn a lot from my association with him.

My sincere thanks to all the members of the Dept. of Computer Science and Engineering, Tezpur University, specially to the members of my doctoral committee for their invaluable comments and suggestions.

I would like to express my gratitude to my employeer Dr. Sylvanus Lamare, principal, St. Edmund's College, Shillong and to the University Grants Commission for granting my study leave, without which it would have been extremely difficult to complete the thesis. I would also like to thank Bro. E. V. Miranda, former principal, St. Edmund's College and my colleagues at the Dept. of Computer Science, St. Edmund's College for their encouragement and support.

Many thanks to all my friends, specially Shantu and Sameer for their encouragement and for being there whenever I needed any help, advice or support.

I will always be grateful to all my family members, to my mom and dad for their unconditional love and support, to my brother for having faith in me, to my dear wife for her love and support, to my little children Rick and Riya for bringing joy to my life and to my sister in law for her encouragement. Once again, thanks a lot to all of them for always being there and for the trouble they had to take to ensure that I get quality time with my research.

Last, but not the least, I would like to thank all who have directly or indirectly helped me in different ways during these years enabling me to complete my work.

Contents

List of Tables

List of Figures

Acronyms

2G Second Generation.

3G Third Generation.

3GPP Third Generation Partnership Project.

3GPP-WLAN Interworking between 3GPP System and WLAN.

AAA Authentication Authorisation Accounting.

AES Advanced Encryption Standard.

AK Anonymity Key.

AKA Authentication and Key Agreement.

AMF Authentication and Key Management Field.

AN Access Network.

ARIB Association of Radio Industries and Businesses.

AS Access Stratum.

ASME Access Security Management Entity.

ATIS Alliance for Telecommunications Industry solutions.

AuC Authentication Centre.

AUTN Authentication Token.

AV Authentication Vector.

BTS Base Transceiver Station.

CCSA China Communications Standards Association.

CDMA Code Division Multiple Access.

CK Cipher Key.

DDoS Distributed Denial of Service.

DMSI Dynamic Mobile Subscriber Identity.

DoS Denial of Service.

E-UTRAN Evolved Universal Terrestrial Radio Access Network.

E2EUIC End to End User Identity Confidentiality.

EAP Extensible Authentication Protocol.

ECB Electronic Code Book.

EDGE Enhanced Data rates for GSM Evolution.

EMSK Extended Master Session Key.

EPC Evolved Packet Core.

ePDG Evolved Packet Data Gateway.

EPS Evolved Packet System.

ERAND Embedded RAND.

ERIC Encrypted RIC.

ETSI European Telecommunications Standards Institute.

FDD Frequency Division duplex.

FIPS PUB Federal Information Processing Standards Publication.

GERAN GSM EDGE Radio Access Network.

GPRS General Packet Radio Service.

GSM Global System for Mobile Communication.

GUTI Globally Unique Temporary Identity.

HLR Home Location Register.

HN Home Network.

HPLMN Home Public Land Mobile Network.

HSS Home Subscription Server.

IK Integrity Key.

IKEv2 Internet Key Exchange version 2.

IMSI International Mobile Subscriber Identity.

IMT International Mobile Telephony.

IP Internet Protocol.

IPsec Internet Protocol Security.

ITU International Telecommunications Union.

K_{ASME} Key for Access Security Management Entity.

KPA Known Plain Text Attack.

LAI Location Area Identification.

LTE Long Term Evolution.

MAC Message Authentication Code.

MCC Mobile Country Code.

MIMO Multiple Input Multiple Output.

MME Mobility Management Entity.

MMOG Multimedia Online Gaming.

MNC Mobile Network Code.

MSC Mobile Switching Centre.

MSIN Mobile Subscription Identification Number.

MSK Master Session Key.

NAI Network Access Identifier.

NAS Non Access Stratum.

NIST National Institute of Standards and Technology.

Non3GPP-EPS Interworking between Non-3GPP Accesses and the EPS.

OFDMA Orthogonal Frequency Division Multiple Access.

PDG Packet Data Gateway.

PDN-GW Packet Data Network Gateway.

PLMN Public Land Mobile Network.

PRNG Pseudo Random Number Generator.

QoS Quality of Service.

RAI Routing Area Identification.

RAND Random Number.

RES Response.

RIC Random-number for Identity Confidentiality.

RN Random Number.

RNC Radio Network Controller.

RRC Radio Resource Control.

SAE System Architecture Evolution.

SGW Serving Gateway.

SIM Subscriber Identity Module.

SMC Security Mode Command.

SN Serving Network.

SQN Sequence Number.

TAU Tracking Area Update.

TDD Time Division Duplex.

TMSI Temporary Mobile Subscriber Identity.

TR Technical Report.

TRAND Transformed RAND.

TS Technical Specification.

TTL_{DMSI} Time to Live for DMSI.

TTA Telecommunications Technology Association.

TTC Telecommunication Technology Committee.

UE User Equipment.

UICC Universal Integrated Circuit Card.

UMTS Universal Mobile Telecommunication System.

UP User Plane.

USIM Universal Subscriber Identification Module.

UTRAN Universal Terrestrial Radio Access Network.

VCR Variable for Collision Resolution.

VLR Visitor Location Register.

VoIP Voice over IP.

VPLMN Visitor Public Land Mobile Network.

W-CDMA Wideband Code Division Multiple Access.

WAG WLAN Access Gateway.

WiMAX Worldwide Interoperability for Microwave Access.

WLAN Wireless Local Area Network.

XRES Expected Response.

Chapter 1

Introduction

Privacy has been a concern for people since the ancient times. Exposure of many of the activities such as movement, access to resources, usage behaviour, etc., of a person may lead to his/her risk of physical security as well as security of his/her resources. Ones activities may be revealed if his identity is known to the adversaries [1]. Hence, the confidentiality of ones identity is of paramount importance.

According to a recent press release of The World Bank, around three-quarters of the World's inhabitants now have access to a mobile phone [2] and the number is increasing with every passing day.

These days, a subscriber uses a mobile phone to access variety of services including voice, rich communication services, and value added services. These services are used for making important communications, accessing valuable resources, and for carrying out financial transactions, because of which a mobile phone is becoming an important tool for an individual's existence. Therefore, the need to protect an individual's identity that is used in a mobile system is as important as the need to protect other important personal identities like social security number and bank account numbers.

A common practice among mobile operators to extend their services beyond their own service area is to establish roaming agreements with third party operators [3]. Roaming allows a subscriber of one operator to use the services of another operator when inside the latter's coverage area. The trust model

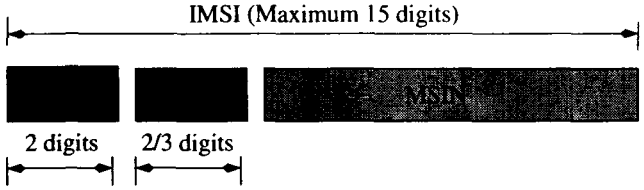

Figure 1.1: Structure of the IMSI.

adopted for roaming in mobile systems developed by Third Generation Partnership Project (3GPP), a collaboration of reputed organisations that is responsible for developing some popular and market winning mobile systems, does not protect the identity privacy of the subscribers from the visited/serving networks. In fact, the visited/serving networks that may often belong to third party operators are required to be trusted with the identity of the subscribers. Such trust requirement makes roaming agreements between operators difficult as it requires elaborate trust agreement amongst them, thereby limiting the ease and span of extending an operator's services beyond its own circle or zone [4]. Moreover, it makes the identity privacy of the subscriber vulnerable to eavesdroppers (over an unencrypted wireless link) and fake serving networks. The study reported in this thesis, aims at improving identity privacy of the subscriber of a 3GPP mobile system while relaxing the trust requirement for roaming between operators.

1.1 Subscriber Identity in Mobile Systems

In mobile networks, each subscriber is registered with a home network. During registration, the subscriber is assigned a Subscriber Identity Module (SIM) that contains a unique and a permanent identity called the International Mobile Subscriber Identity ($IMSI$) that identifies the subscriber. The $IMSI$ is a number (Figure 1.1) that constitutes of a maximum of 15 decimal digits [5]. The first 3 digits are the Mobile Country Code (MCC), which is followed by the Mobile Network Code (MNC) (either 2 digits, in case of European standard or 3 digits, in case of North American standard). The length of the MNC depends on the value of the MCC. The remaining digits are the Mobile Subscription

Identification Number ($MSIN$) [6]. Thus,

$$IMSI = MCC||MNC||MSIN \tag{1.1.1}$$

where, '||' denotes concatenation. The $IMSI$ is used by the home network to uniquely identify each and every subscriber for authentication, authorisation and billing purposes. The MCC identifies the country of domicile of the mobile subscriber, whereas the MNC identifies the home network of the mobile subscriber. The $MSIN$ is used to uniquely identify a subscriber within the subscriber's home network.

1.2 Subscriber Identity Privacy in Mobile Systems

Identity Privacy is considered a standard security requirement in any mobile telecommunication system [7][8][9]. The identity privacy of a subscriber is compromised if his/her permanent identity (i.e., the $IMSI$) is exposed to an adversary. Knowledge of the $IMSI$ may allow an adversary to track and amass comprehensive profiles about individuals - where, data about movement, usage, etc., of a subscriber is collected over a period of time and linked with his/her $IMSI$. Such profiling may expose an individual to various kind of unanticipated risks and above all will deprive an individual of his privacy. Thus, with more and more people accessing voice, Internet, rich communication services, value added services, mobile banking, mobile commerce, etc., through mobile networks, the importance of identity privacy cannot be underestimated.

1.3 3GPP Mobile Systems: An Overview

3GPP is a collaboration of six influential telecommunications standard development organisations worldwide (ARIB of Japan, ATIS of USA, CCSA of China, ETSI of France, TTA of korea and TTC of Japan). The original scope of 3GPP was to produce Technical Specifications (TSs) and Technical Reports (TRs) for

a third generation (3G) mobile system that is based on a second generation (2G) mobile telecommunication system called the Global System for Mobile Communication (GSM). The scope was subsequently amended to include the maintenance and development of TSs and TRs for GSM and evolved radio access technologies (e.g., General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE)). The technologies standardised by 3GPP are constantly evolving through generations of mobile systems. Since the completion of the first Long Term Evolution (LTE) specification, 3GPP has become the focal point for mobile systems beyond 3G. In this section, we present an overview of some prominent mobile systems developed by 3GPP, viz.: Universal Mobile Telecommunication System (UMTS), LTE, interworking between 3GPP System and WLAN (3GPP-WLAN) and Non 3GPP Access to the EPS (Non3GPP-EPS).

1.3.1 UMTS

UMTS is a 3G mobile system developed and maintained by 3GPP. It has evolved from the GSM standard and therefore borrows heavily from the established network architecture of GSM. In fact, many of the network elements used in GSM are reused (with some enhancements) in UMTS. The radio access network, however, in UMTS is significantly different from that of GSM, GPRS, and EDGE. In UMTS, the radio access network is known as the Universal Terrestrial Radio Access Network (UTRAN). The components that make up the UTRAN are significantly different from the corresponding elements in the GSM architecture. It supports both circuit switched and packet switched connections for real time and data communication services respectively [10]. It uses Wideband Code Division Multiple Access (W-CDMA) radio access technology to offer greater spectral efficiency and bandwidth compared to its 2G counterpart (i.e., GSM). It includes two of the air interface proposals submitted to the International Telecommunications Union (ITU) as proposed solutions to meet the requirements laid down for International Mobile Telephony 2000 (IMT-2000). One solution uses Frequency Division Duplex (FDD) and the other uses Time Division Duplex (TDD) [11]. In the FDD option, paired 5-MHz carriers are used in the uplink and downlink

(1920 MHz to 1980 MHz for uplink and 2110 MHz to 2170 MHz for downlink). For the TDD option, a number of frequencies have been defined. including 1900 MHz to 1920 MHz and 2010 MHz to 2025 MHz.

1.3.2 LTE

The increase in demand for higher data rates and quality of service due to emergence of applications such as MMOG (Multimedia Online Gaming), mobile TV, Web 2.0, etc., have inspired the 3GPP to work on the Long Term Evolution (LTE). LTE is the latest standard in the mobile network technology tree that previously realised the GSM/EDGE and UMTS network technologies [12]. LTE is designed to meet the continued demand for cost reduction and to ensure the continuity of competitiveness of the 3GPP systems for the future.

LTE, whose radio access network is called Evolved Universal Terrestrial Radio Access Network (E-UTRAN) [13], substantially improves end-user throughputs, sector capacity and reduces user plane latency, bringing significantly improved user experience with full mobility [14]. With the emergence of Internet Protocol (IP) as the protocol of choice for carrying all types of traffic, LTE provides support for IP-based traffic with end-to-end Quality of service (QoS). Voice traffic is supported mainly as Voice over IP (VoIP) enabling better integration with other multimedia services.

As a part of its System Architecture Evolution (SAE) initiative, 3GPP has specified a new flat IP based core network called the Evolved Packet Core (EPC) to support the E-UTRAN through a reduction in the number of network elements, simpler functionality, improved redundancy and allowing connections to other fixed/wireless access technologies, giving the service providers the ability to deliver a seamless mobility experience

LTE is designed to meet aggressive performance requirements that rely on physical layer technologies, such as, Orthogonal Frequency Division Multiplexing (OFDM), Multiple-Input Multiple-Output (MIMO) systems and smart antennas to achieve these targets. The main objectives of LTE are to minimise the system and user equipment complexities, allow flexible spectrum deployment in existing

or new frequency spectrum, and to enable co-existence with other 3GPP radio access technologies. LTE supports scalable carrier bandwidths, from 1.4 MHz to 20 MHz and supports both FDD and TDD. With LTE, the highest theoretical data rate is 170 Mbps in uplink and with MIMO the rate can be as high as 300 Mbps in the downlink.

1.3.3 3GPP-WLAN

Mobile telecommunication systems like EDGE and UMTS that are proposed by 3GPP are called 3GPP systems. 3GPP systems have large coverage, high speed mobility, efficient subscriber management, expertise in billing and nearly universal roaming. Whereas, WLAN provides hot-spot/limited coverage with a data rate much higher and cost which is much lesser than that of 3GPP systems. The combination of 3GPP systems and WLAN technologies offer the possibility of achieving any time, anywhere services, bringing benefits of both technologies to the end users and the service providers. Thus, with the intent to extend 3GPP services and functionality to the WLAN access environment, 3GPP has proposed specification for interworking between 3GPP system and WLAN [15].

1.3.4 Non3GPP-EPS

As a part of the 3GPP's LTE/SAE initiative for the evolution of GSM, EDGE and UMTS architecture, a purely IP based system called the Evolved Packet System (EPS) is standardised. E-UTRAN is the access part of the EPS whereas the EPC is its core network. In order to expand the reach of 3GPP services beyond 3GPP defined accesses, viz., GSM EDGE Radio Access Network (GERAN) of GSM/EDGE, UTRAN of UMTS, E-UTRAN of LTE, etc., 3GPP has proposed the technical specification for interworking between the EPS and accesses that were not defined by 3GPP (Non-3GPP accesses) [16]. This specification provides description for providing IP connectivity using Non-3GPP accesses, viz., Worldwide Interoperability for Microwave Access (WiMAX), Code Division Multiple Access 2000 (CDMA2000), WLAN, etc., to the EPC.

1.4 Subscriber Identity Privacy in 3GPP Mobile Systems

For access security in mobile systems developed by 3GPP, an Authentication and Key Agreement (AKA) protocol is performed between the subscriber's user equipment and the subscriber's home network. During this procedure, both the user equipment and the home network mutually authenticate each other. To initiate the AKA procedure during roaming, the subscriber is required to present its identity to the visited/serving network through the wireless link between them, for onward transmission to the home network. Since, identity presentation during an AKA precedes all other security, a challenging task at this stage, is to protect the identity privacy of the subscriber from the visited/serving network and from eavesdroppers in the vulnerable wireless link.

In order to provide identity privacy to the subscribers in mobile systems developed by 3GPP, the permanent identity (i.e., the $IMSI$) of the subscriber is replaced by temporary identities and pseudonyms. Instead of the $IMSI$, short lived temporary identities/pseudonyms are used for identity presentation. In case of mobile systems like UMTS and LTE, temporary identities are allotted to the user equipment by the visited/serving network. Whereas, in case of mobile systems like 3GPP-WLAN and Non3GPP-EPS, pseudonyms are allotted to the user equipment by the home network. A mapping between the temporary identities and the corresponding $IMSIs$ are also maintained by serving network (in case of UMTS and LTE) or the home network (in case of 3GPP-WLAN and Non3GPP-EPS), so that the serving/home network can resolve them back to corresponding $IMSI$ when required. While generating temporary identities/pseudonyms, the following is ensured:

- A temporary identity/pseudonym should not have any correlation with any previously generated temporary identity/pseudonym.
- It should not be possible for anyone except the visited network (in case of UMTS and LTE) or the home network (in case of 3GPP-WLAN and Non3GPP-EPS) to resolve the corresponding $IMSI$ from a given tempo-

rary identity/pseudonym.

1.5 Motivation of the Thesis

In this section, we discuss the vulnerabilities related with identity privacy in mobile systems developed by 3GPP. These vulnerabilities provide scope for further improvement and are the motivating factors behind the work presented in this thesis.

The trust model adopted by 3GPP for roaming requires an operator and its subscribers to have full trust on the visited/serving networks (that may even belong to third party operators). The existing mechanism for identity privacy in the AKA protocols used in 3GPP mobile systems (Section 1.4) is based on this trust requirement. As a result, the mobile systems developed by 3GPP have the following limitations:

1. In situations when the visited/serving network cannot resolve a presented temporary identity/pseudonym to its corresponding $IMSI$, there is a backup mechanism under which the visited/serving network has a provision to request the subscriber for its permanent identity. A subscriber has to oblige such a request by transmitting its $IMSI$ in clear text through the radio path. Such a provision makes the subscriber vulnerable to the following types of adversaries:

    - Eavesdroppers: An eavesdropper listening to the radio link will be able see the $IMSI$ being transmitted in clear text.
    - Impersonators/Man-in-the-middle: A man-in-the-middle can impersonate as a genuine network and after drowning the signals of the actual network with its own signals, can send a request for permanent identity to the subscriber. A response will contain the $IMSI$ in clear text.
    - Corrupt visited/serving network: Because of the above provision, a roaming subscriber will have to trust a serving network with its $IMSI$,

even if the serving network does not belong to the same mobile operator as the home network. A corrupt serving network operator may take advantage of this situation by compromising the identity related information to a malicious third party.

2. To Provide secured roaming services including identity privacy to the subscribers, the operators are required to set up prior trust relationships through elaborate roaming agreements with the third party operators. However, such roaming agreements limit the ease and span of extending services beyond an operator's own circle/zone.

Hence, there is need that the requirement of trust on third party operators for roaming is relaxed or even entirely eliminated.

1.6 Contributions of the Thesis

The contribution of the work reported in this thesis can be summarised as follows:

1. In order to relax trust requirement for roaming, a new trust model which is more flexible compared to the existing trust model is proposed. In this trust model, the need to trust the visited/serving network is relaxed with respect to identity of the subscriber.

2. A security extension for the AKA protocols used in 3GPP mobile systems is developed to implement the proposed trust model. In this extension, the permanent identity of the subscriber is restricted to the subscriber's mobile device and the home network. Hence, we call it End to End User Identity Confidentiality (E2EUIC). In situations when the visited/serving network cannot resolve a presented temporary identity/pseudonym to its corresponding $IMSI$, in E2EUIC a dynamic identity whose value keeps changing after every successful AKA is transmitted instead of the permanent identity. Advantages of E2EUIC are the following:

    - There is no need to trust the serving network for identity privacy of the subscriber.

- Contrary to several recent proposals in this area, E2EUIC can be adopted as an extension to the existing AKA protocols used in 3GPP mobile systems.
- E2EUIC can be implemented at the operator's level without needing any modification in the intermediary networks.

3. Adoptions of this extension are developed for 3GPP mobile systems like UMTS, LTE, 3GPP-WLAN and Non3GPP-EPS.

4. Various analyses like statistical analysis, formal analysis, computational cost analysis, complexity analysis, space overhead analysis, communication overhead analysis and security analysis of E2EUIC are performed that establishes its robustness, correctness and effectiveness.

5. A comprehensive review of the literature has been carried out which provides an insight into the existing methodology used for identity privacy in 3GPP mobile systems and the proposed alternatives. It also helped us in formulating an approach that is different from the solutions already proposed in the literature and we believe, is more suitable for mobile systems developed by 3GPP.

1.7 Organisation of the Thesis

The thesis is organised as follows:

- In *Chapter* 2, the existing trust model used for roaming in 3GPP mobile systems is discussed. A new trust model that overcomes the limitations of the existing trust model is proposed in this chapter.
- In *Chapter* 3, a security extension called E2EUIC for the AKA protocol used in UMTS is proposed.
- In *Chapter* 4, the adoption of E2EUIC to the AKA protocol used in LTE is presented.

- In *Chapter* 5, the adoption of E2EUIC to the AKA protocol used in interworking systems proposed by 3GPP, viz., 3GPP-WLAN and Non3GPP-EPS, are presented.

- In *Chapter* 6, various analyses like formal analysis, computational cost analysis, complexity analysis, space overhead analysis, communication overhead analysis and security analysis of E2EUIC are presented that establishes its robustness, correctness and effectiveness.

- In *Chapter* 7, a comprehensive review of the literature is performed and discussed with reference to E2EUIC.

- In *Chapter* 8, concluding remarks are given and some of the future research directions are highlighted.

Chapter 2

Trust Model

As mobile systems evolved from the 1st generation of analog technologies to the recent 4th generation marked by advanced technologies like LTE, so has the subscriber's expectation. This has led to the demand of 'anywhere' service, immaterial of the location of the subscriber or the coverage area of the mobile operator. Due to the existing trust model adopted for roaming by the mobile systems, such services necessitates the mobile operators to set up elaborate prior trust relationships and agreements between themselves. This however limits the ease and span of extending the services by an operator across a wider geographical area. Moreover, it does not do any good to the subscriber's identity privacy. In this chapter, we look into the existing trust model adopted in the mobile systems developed by 3GPP. We also look into the weaknesses of this model. We then propose a new trust model that can enhance identity privacy of the subscribers and has the potential to reduce trust requirements (needed for establishing roaming agreements) between mobile operators.

2.1 Introduction

Across all the generations of mobile systems developed by 3GPP, be it GSM, a popular 2G standard or be it LTE, an upcoming 4G standard, a common architectural framework for roaming is used. In this framework, three parties are involved, viz., the User Equipment (UE), the Home Network (HN) and the

Figure 2.1: Basic architecture of a mobile system.

Visited/Serving Network (SN) (Figure 2.1). The UE that a subscriber owns is registered with a HN. The association between the UE and the HN is created from the moment the subscriber procures a Subscriber Identity Module (SIM) from the HN and installs it into his/her UE. The HN offers roaming services to its registered UEs through SNs that are located outside its own service area. Communication between the UE and the SN happens through radio link, whereas communication between the SN and the HN happens through wired medium. The radio link is vulnerable to various kinds of attacks by adversaries as it is too open by nature, whereas the wired link is considered to be secured [17]. The secure nature of the wired link is a result of trust relationship that exists between the SN and the HN.

In the existing security architecture adopted by 3GPP (for the mobile systems developed by it), an effort is made to protect identity privacy of the subscriber by limiting the transmission of $IMSI$ over the radio link through the use of temporary identities [18][19][20]. After every successful mutual authentication between the UE and the SN, a temporary identity is allocated to the UE by the SN through a secured channel. The association between a temporary identity and the corresponding $IMSI$ is maintained in the SNs local database. To have access to a particular service, an UE has to send a service request along with its temporary identity to the SN. The SN, in turn presents the corresponding $IMSI$ to the HN - in order to obtain relevant authentication data for the UE. It then uses this authentication data in a challenge response mechanism to authenticate the requesting UE [21].

To ensure safety of the $IMSI$ while it is stored at the SNs local database, adequate trust relationship needs to be established between the SN and the HN through negotiations and agreements. While this might not be an issue for SNs owned by the same mobile operator as the HN, elaborate agreements are necessary if that is not the case. With the demand for anywhere service, it is not

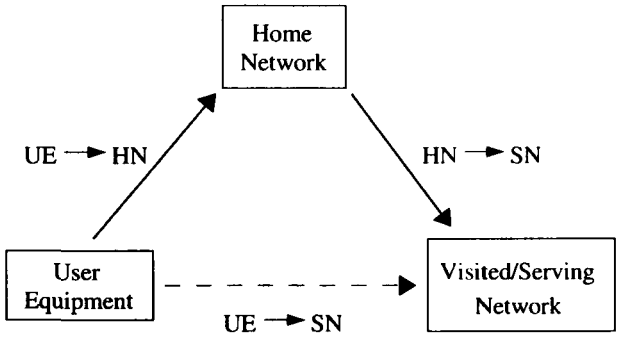

Figure 2.2: Existing trust model.

possible to always ensure that the SNs providing the support belong to the same home mobile operator. Thus, an operator will require roaming agreements with third party mobile operators to ensure the necessary level of trust amongst them. This, being a prior requirement, will limit the extent of serviceable geography. In the following sections, we look at the existing trust model adopted for roaming in the mobile systems developed by 3GPP. We then look into the limitations of this model, and finally we propose a new trust model. The new trust model can strengthen identity privacy of the subscribers while relaxing trust relationship requirement for roaming between mobile operators.

2.2 Existing Trust Model

In the existing trust model used for roaming in 3GPP mobile systems, the following trust requirements with reference to the permanent identity of a subscriber exist (Figure 2.2):

1. **UE → HN**: As the UE is registered to, and has a direct service agreement with the HN, it trusts the HN with its $IMSI$.

2. **HN → SN**: Since the HN serves its roaming subscribers through SNs, the HN confers full trust in the SN with regards to the $IMSI$ of the subscriber. As a result, for authentication, authorisation and billing purposes, the $IMSI$ is exchanged unabated between the HN and the SN.

3. **UE → SN**: This trust relation is a transitive outcome of the previous two

trust relations, because of which, the UE has to trust the SN with its $IMSI$ and it transmits the $IMSI$ immediately upon request from the SN.

2.3 Limitations of the Existing Trust Model

The following vulnerabilities/limitations exist in the above model:

- To fulfil the second trust requirement in the model (i.e., HN $\rightarrow$ SN), prior agreements needs to be set up. Thus, this trust requirement may deprive a roaming subscriber from services in a location where there is no SN having a prior agreement with the subscriber's HN operator.

- Due to the third trust requirement (i.e., UE $\rightarrow$ SN), the UE has to transmit its $IMSI$ in clear text through the radio link any time when the SN requests for it. The SN has provision to make such a request when it cannot map the received temporary identity of an UE with the corresponding $IMSI$. Such a provision makes identity privacy of the subscriber vulnerable to eavesdroppers [7], man-in-the-middle (fake serving networks) [22] and corrupt serving networks [21].

- During the very first connection, there is no temporary identity by which the UE can be identified by the SN [23]. In such a situation, making use of the third trust requirement in the model (i.e., UE $\rightarrow$ SN), the SN requests the UE for its $IMSI$; in response to which the UE has to transmit its $IMSI$ in clear text.

Roaming agreements with third party mobile operators to provide service in a location where an operator has not set up its own infrastructure is a common practice. The trust model adopted for roaming in 3GPP mobile systems call for unconditional trust requirement on part of the subscribers. In today's context when multiple operators collaborate with each other to extend their services across a wider geographical area, such trust relationship requirement imposes restriction and brings in overheads towards providing 'anywhere' service to the

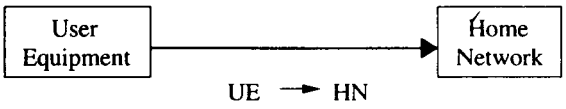

Figure 2.3: Proposed trust model.

subscriber. Thus, there is need for a paradigm shift such that the requirement of trust on third party SNs is relaxed or even entirely eliminated.

2.4 Proposed Trust Model

In this section, we propose a new trust model [24] for roaming in 3GPP mobile systems that is based on the following observations:

- To overcome the vulnerabilities discussed in Section 2.3, the HN/UE should not have the need to trust the SN with the $IMSI$ of the UE.
- Due to low power and low computational capability of an UE, public key based solutions are not feasible for mobile networks [25].
- With an alternate mechanism of identity presentation sans $IMSI$, the HN $\rightarrow$ SN trust relationship can be considerably relaxed.

The proposed trust model is more flexible than the existing model. It has only one trust requirement (Figure 2.3), which is as follows:

1. **UE $\rightarrow$ HN**: The UE should trust only the HN with which it is registered and no one else. The $IMSI$ should not be shared with any third party and under no circumstance should leave the UE or the HN.

2.5 Implementing the Proposed Trust Model

In this section, we present a scheme that can be used for successful implementation of our proposed trust model (Section 2.4). We formulate this mechanism to work as an extension on top of the existing AKA protocols used for access security in mobile systems proposed by 3GPP. In this mechanism, a second set

of temporary identities (say TI_{HN}), over and above those that are already being used between the UE and the SN (say TI_{SN}), are to be used between the UE and the HN, and these are to be generated and distributed by the HN. These temporary identities (i.e., TI_{HN}) should be transmitted by the UE to the SN, for identity presentation, in situations that otherwise needs transmitting the $IMSI$. Where as, the role of the TI_{SN} remains the same. TI_{HN} distribution mechanism should ideally be integrated with the AKA procedure, to avoid extra communication latency. Towards this, the HN may include TI_{HN} in that part of the authentication data, which reaches the UE as a challenge via the SN (of course, in an encrypted form which can only be decrypted by the UE). A TI_{HN} should be such that when presented by the UE, should reveal the owner HN identity to the SN, so that the latter can approach the corresponding HN for authentication data. A mapping between TI_{HN} and the $IMSI$ should be maintained at the HN. Such a mapping would help the HN to easily identify the corresponding $IMSI$ when a TI_{HN} is presented to it. For convenience of identity presentation during the very first connection in the life time of a SIM, the HN should embed a TI_{HN} into the SIM's memory. This embedding should be done before SIM distribution. During successive connections, if a presented TI_{SN} cannot identify the UE, a TI_{HN} (that is received by the UE from the HN during the previous successful authentication) is used for identity presentation instead of the $IMSI$. The UE can use the same TI_{HN} more than once for identity presentation, as long as it does not receive the next TI_{HN} from the HN. Like other sensitive information viz., the subscriber's security credentials, billing details, etc., the TI_{HN} to $IMSI$ mappings should be robust against database failures.

2.6 Advantages of the Proposed Trust Model

With our proposed trust model, the need for the UE to transmit its $IMSI$ to the SN is completely eliminated. Thereby, securing identity privacy of the subscriber from eavesdroppers, man-in-the-middle and corrupt serving networks. With this trust model, there will no longer be a requirement for prior trust agreement between the HN operator and the SN operator with respect to the subscriber's

identity privacy. This opens up an opportunity to have on-demand/on-the-fly roaming agreements between mobile operators, instead of the current prior agreements. Through such agreements, any SN available in the serving area will be able to serve the UE. This might also allow for auctioning of service (including QoS) by various SNs serving a given area based on their current load. The involvement of the HN in generation of temporary identities and its ability to associate a temporary identity with the corresponding $IMSI$ would ensure a process for billing of the services provided by the SN. In addition to improved identity privacy, subscribers who wish to protect their data from being interpreted by the SNs may use end to end application layer based ciphering and integrity protection solutions [26][27]. Moreover, mobile networks are gradually moving towards all-IP packet switched mode, where end to end Internet Protocol Security (IPsec) based solutions can be used to protect the IP packets [28][29].

2.7 Summary

In this chapter, the existing trust model used for roaming in 3GPP mobile systems is analysed. It is found that if the requirement of having to trust the SNs is relaxed from the existing trust model, it then opens up an opportunity to have on-demand/on-the-fly roaming agreements between mobile operators, instead of the current prior agreements. This would ensure that a subscriber will be serviceable in any location as long as there is at least one network serving that location. With more and more mobile operators taking a plunge into the competitive cellular market, collaborations through roaming agreements amongst them is a key issue. Thus, the benefits of a relaxed UE/HN $\rightarrow$ SN trust requirement would be difficult to ignore in the foreseeable future. Hence, a new trust model is proposed, which is more flexible compared to the existing trust model. In this trust model, the need for the UE/HN to trust the SN is relaxed. A scheme to implement this trust model with reference to the identity privacy of the subscriber is also proposed. In this scheme, the knowledge of the $IMSI$ is restricted to the UE and the HN, where the need for the UE to transmit the $IMSI$ is eliminated.

Chapter 3

Relaxing Trust Requirement in UMTS

Identity Privacy is considered a standard security feature in any mobile telecommunication system. UMTS is no exception and this requirement is clearly spelt out in 3GPP TS 33.102 [19]. However, there is a security vulnerability in UMTS, due to which the identity privacy of a subscriber gets compromised. In addition, in UMTS it is assumed that there is no threat from the SNs, even if they belong to third party operators. All of this may be attributed to the existing trust model adopted in UMTS.

In this chapter, we propose a new security extension called End to End User Identity Confidentiality (E2EUIC) for the AKA protocol used in UMTS. The extension is based on our proposed trust model (Chapter 2, Section 2.4), and has the potential to improve the subscriber's identity privacy while relaxing trust requirement for roaming between mobile operators. While designing the extension, we question the trustworthiness of the SNs themselves, on which the existing mechanism to protect identity privacy is based. Our extension relaxes HN to SN and hence, UE to SN trust relationship requirement with respect to the subscriber's identity privacy. We propose to replace the transmission of $IMSI$ with a Dynamic Mobile Subscriber Identity ($DMSI$). Unlike $IMSI$, the value of $DMSI$ is not static, thus enhancing identity privacy/confidentiality.

3.1 Introduction

Like all other mobile systems proposed by 3GPP, a unique $IMSI$ is assigned to identify the subscriber in UMTS. The identity privacy of the subscriber gets compromised if his/her $IMSI$ gets exposed. Towards this end, 3GPP endeavours to limit the transmission of $IMSI$ to the wired part of the network, as the wireless link is too open for various kinds of attacks [17][23].

In order to limit transmission of $IMSI$ over the wireless link, the SN assigns a local temporary identity called Temporary Mobile Subscriber Identity ($TMSI$) to the UE through a ciphered channel. The value of this temporary identity is short lived and is refreshed frequently by the SN. The SN keeps the association between a $TMSI$ and its corresponding $IMSI$ in its local database. Whenever the UE needs to present its identity, it transmits the $TMSI$ instead of its $IMSI$. The SN can easily correlate this $TMSI$ to the corresponding $IMSI$ through the $TMSI$ to $IMSI$ mapping maintained in its local database. If the subscriber roams into a new SN and it produces a $TMSI$ obtained from the old SN, the association between the produced $TMSI$ and the corresponding $IMSI$ is obtained by the new SN from the old SN.

In UMTS, there are circumstances when a $TMSI$ fails to identify the UE, forcing transmission of $IMSI$ in clear-text over the wireless link. Such situations make identity privacy of the subscriber vulnerable [30]. Moreover, there is full trust relationship among the agents in the wired network and the $IMSI$ is exchanged freely among them. Such trust requirement limits interoperability between mobile operators as it complicates roaming agreements.

3.2 Security Architecture of UMTS

The security architecture of UMTS (Figure 3.1) involves three primary participants, namely: the UE, the SN and the HN [31]. The UE is any device that is used by the subscriber to communicate. It can be a hand-held telephone, a laptop computer, or any other device that is fitted with a Universal Subscriber Identity Module (USIM) [32]. Every UE has to be registered with a HN (with

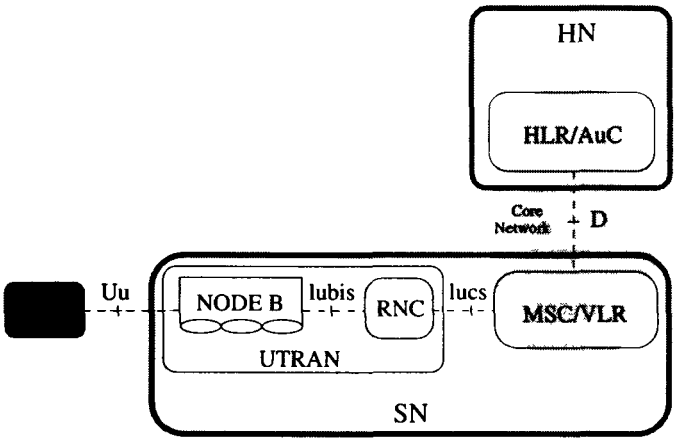

Figure 3.1: Security architecture of UMTS.

their security credentials stored at the HN's database). The HN contains key security elements like the Home Location Register (HLR) and the Authentication Centre (AuC). The HLR stores permanent sensitive information of the subscribers such as identity, service profile, activity status, etc., whereas the AuC is a protected database that stores association between subscriber identities and long-term keys. The HN extends its services to its roaming subscribers through the SNs. The SN contains elements like the Visitor Location Register (VLR) and the Mobile Switching Center (MSC). The VLR stores temporary information about the subscribers visiting the SN and maintains temporary to permanent identity associations, whereas the MSC offer circuit-switching domain services. The UE directly communicates with a Base Transceiver Station (BTS) or NodeB (through the Uu reference point/interface). One or more NodeBs are connected with a Radio Network Controller (RNC) (through the Iubis reference point/interface). The RNC manages the radio resources and is the interface between the UE and the core network (through the Iucs reference point/interface). Communication between the UE and the SN happens over radio link, whereas communication between the SN and the HN happens through wired link. While the radio link is considered to be vulnerable, it is assumed that the wired links are adequately secure.

3.3 UMTS-AKA

UMTS-AKA is the AKA protocol used in UMTS for access security [19]. It is carried out in two stages [23][17], which are as follows:

- In the first stage, the UE presents its identity to the SN. The SN, with the help of this identity, obtains the security credentials of the UE in the form of a set of Authentication Vectors (AVs) from the HN.
- In the second stage, the SN utilises one of these AVs to perform mutual authentication with the UE through a challenge response mechanism. In this phase, a Cipher Key (CK) and an Integrity Key (IK) are established between the UE and the SN, so that communication over the otherwise vulnerable radio link (between the UE and the SN) can happen in a secured and reliable way.

In order to facilitate the authentication mechanism, each UE shares with its HN a long term secret key Ki and certain cryptographic algorithms, viz., f_0, f_1, f_2, f_3, f_4, f_5, f_8 and f_9. Where, f_0 is the random challenge generating function, f_1 is the network authentication function, f_2 is the user challenge response authentication function, f_3 is the cipher key derivation function, f_4 is the integrity key derivation function, f_5 is the anonymity key derivation function, f_8 is the confidentiality key stream generating function and f_9 is the integrity stamp generating function. A set of example algorithms for these functions called MILENAGE are proposed in [33]. In order to assure freshness of authentication data, two counters, viz., SQN_{UE} and SQN_{HN} are maintained at the UE and the HN respectively. Detailed functionality of both the stages of UMTS-AKA are described in the following two subsections:

3.3.1 Distribution of Authentication Data

1. The UE presents its identity to the SN by transmitting it through the radio channel.
2. In case, the presented identity is a temporary identity, the SN locates the

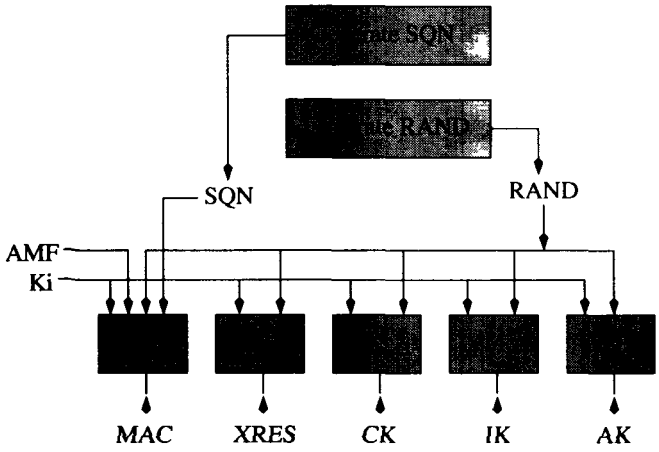

Figure 3.2: Generation of AV.

corresponding $IMSI$ using the $TMSI$-$IMSI$ mapping maintained in its local database. If the SN already has unused authentication data stored in its local database against this IMSI, the remaining steps of this stage are skipped. Otherwise, the SN sends an authentication data request to the HN along with the UE's $IMSI$.

3. Upon receipt of the request, the HN generates an ordered array of M authentication vectors denoted by $AV[1..M]$. Each AV is a quintet, consisting of five elements, viz.: a Random Number ($RAND$), an Expected Response ($XRES$), a Cipher Key (CK), an Integrity Key (IK), and an Authentication Token ($AUTN$). An AV in $AV[1..M]$ is generated according to the following steps (Figure 3.2):

   (a) The HN generates a Random Number $RAND$ using the function f_0, and a Sequence Number SQN from the counter SQN_{HN}.

   (b) The HN then calculates the following values:

$$XRES = f_{2K_i}(RAND) \tag{3.3.1}$$

$$CK = f_{3K_i}(RAND) \tag{3.3.2}$$

$$IK = f_{4K_i}(RAND) \tag{3.3.3}$$

$$AK = f_{5K_i}(RAND) \tag{3.3.4}$$

$$MAC = f_{1K_i}(SQN||RAND||AMF) \tag{3.3.5}$$

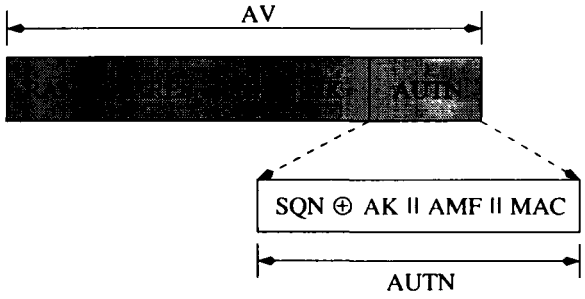

Figure 3.3: Content of AV.

Where, AK: Anonymity Key, MAC: Message Authentication Code, AMF: Authentication and Key Management Field, and '||' denote concatenation. AK is used to conceal the sequence number, as the later may expose the location of the user. If no concealment is needed, AK is set to zero.

(c) After this, HN assembles the Authentication Token:

$$AUTN = SQN \oplus AK \| AMF \| MAC \tag{3.3.6}$$

and the Authentication Vector:

$$AV = (RAND, XRES, CK, IK, AUTN) \tag{3.3.7}$$

where '$\oplus$' is bit wise Exclusive OR operation (Figure 3.3).

(d) The HN increments SQN_{HN} by 1.

4. Finally, the HN sends $AV[1..M]$ back to the SN.

3.3.2 Authentication and Key Agreement

1. The SN selects the first unused AV from the received $AV[1..M]$. It then extracts $RAND$ and $AUTN$ from the selected AV and sends it to the UE as a challenge.

2. Upon receipt of $RAND$ and $AUTN$, the UE calculates AK using Equation 3.3.4. Using the calculated AK, the sequence number SQN is then

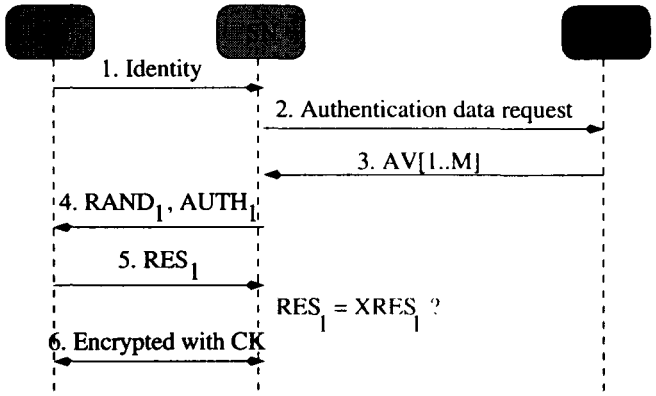

Figure 3.4: UMTS authentication and key agreement.

retrieved from $AUTN$ (Equation 3.3.6) and compared with SQN_{UE} in order to verify freshness of the challenge. The UE then computes MAC using Equation 3.3.5 and compares this value with the MAC included in $AUTN$ (Equation 3.3.6). If they are different, the UE rejects the connection procedure, otherwise it accepts it. Finally, the UE computes the following:

$$RES = f_{2Ki}(RAND) \tag{3.3.8}$$

and sends RES back to the SN.

3. Upon receipt of the RES, the SN compares it with $XRES$ ($XRES$ is a constituent of the selected AV, Equation 3.3.7). If these values match, the authentication process is considered successful. CK and IK, calculated at either end (using Equation 3.3.2 and Equation 3.3.3) are used to secure further communications between the SN and the UE.

The UMTS-AKA procedure is schematically expressed in Figure 3.4.

3.4 Identity Privacy in UMTS

To achieve identity privacy in UMTS, temporary identities (i.e., $TMSIs$) are used. A $TMSI$ is assigned to the UE by the SN only after a secure channel is established between them. The channel has to be secured using CK and IK generated during the previous successful UMTS-AKA. The $TMSI$, when

available, is used instead of the $IMSI$ to identify the subscriber over the radio access path, for paging requests, location update requests, attach requests, service requests, connection re-establishment requests and detach requests. A $TMSI$ only has local significance in the location area or the routing area in which the user is roaming. Outside that area, it should be appended with an appropriate Location Area Identification (LAI) or Routing Area Identification (RAI) in order to avoid ambiguities. The association between a $TMSI$ and its $IMSI$ is maintained at the SN. To avoid user traceability, which may lead to the compromise of identity privacy, the user should not be identified by means of the same $TMSI$ for a long period. The allocation of a new $TMSI$ is initiated by the SN. The SN generates a new $TMSI$ (say $TMSI_n$) and stores the association of $TMSI_n$ and the $IMSI$ in its database. $TMSI_n$ should be unpredictable. The SN then sends this new $TMSI_n$ and (if necessary) the new location area identity (say LAI_n) to the user through a ciphered channel. Upon receipt, the UE stores $TMSI_n$ and automatically removes the association with any previously allocated $TMSI$. The UE sends an acknowledgement back to the SN. Upon receipt of the acknowledgement, the SN removes the association (if there was any) between the old temporary identity $TMSI_o$ and the $IMSI$ from its database. If the SN does not receive an acknowledgement from the UE (informing it of the successful allocation of a temporary identity), the SN shall maintain both the $TMSI_n$ to $IMSI$ and $TMSI_o$ to $IMSI$ associations.

When the subscriber roams into a new region, he/she presents his/her identity to the new SN (say SN_n) by transmitting the $TMSI$ that was allocated to it by the old SN (say SN_o) along with the identity of SN_o. SN_n obtains the association between the $TMSI$ and the $IMSI$ from SN_o, and uses this $IMSI$ to request necessary authentication data from the HN.

3.5 Motivation

In spite of the security mechanism used for identity privacy in UMTS (Section 3.4), there are situations in UMTS-AKA when the identity privacy of a subscriber

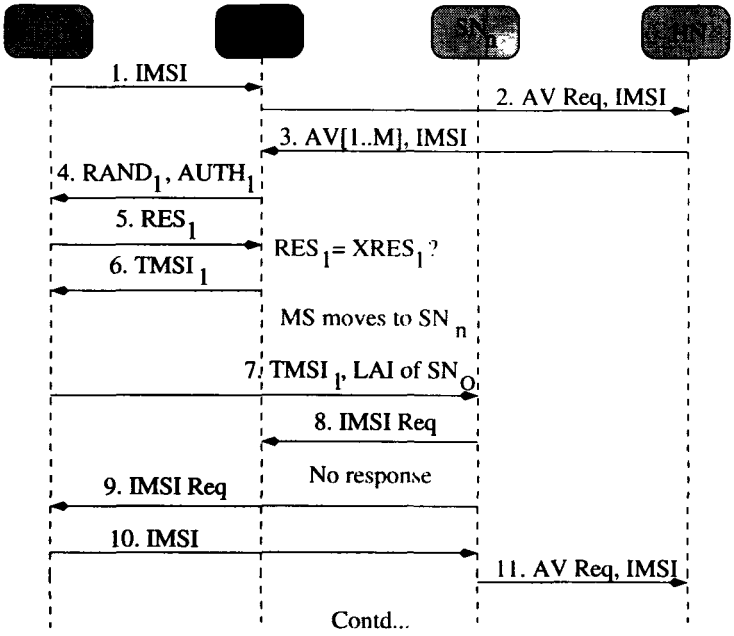

Figure 3.5: Transmission of IMSI in clear text.

becomes vulnerable. Such situations, which are also the motivation of the work behind this chapter, are as follows:

- *The UE is switched on for the first time and has not yet received a TMSI:* In such a situation, the UE is forced to present its identity by transmitting its $IMSI$ in clear-text through the radio link (Message 1, Figure 3.5).

- *The SN cannot map a presented $TMSI$ to its corresponding $IMSI$:* In such a situation (that may arise due to reasons like database failure, etc.), the SN has a provision to request the UE for its $IMSI$. Such a request requires the UE to transmit its $IMSI$ in clear-text through the radio link.

- *A new SN cannot contact the old SN for the $TMSI$-to-$IMSI$ mapping of a roaming subscriber:* When a subscriber moves into the region of a new SN (say SN_n), the UE will present its identity to SN_n through the $TMSI$ allocated to it by the previous SN (say SN_o) along with the LAI of SN_o (Message 7, Figure 3.5). In order to request for a new set of AV from the HN, SN_n will need to have knowledge of the $IMSI$. Normally, this will be obtained by presenting the $TMSI$ to SN_o. However, in case SN_o cannot be contacted, SN_n will be forced to ask the UE for its $IMSI$. The later

will then have to be transmitted in clear-text over the radio link by the UE (Message 10, Figure 3.5). This vulnerability can in fact be exploited by an attacker who can masquerade as a new SN.

- UMTS-AKA assumes full trust relationship within the wired intermediary service network components, and hence $IMSI$ is transmitted freely between the SN and the HN. But, in practice the HN operator does not own all the SNs through which it provides services to its roaming subscribers, and as such trusting the SNs with the $IMSI$ may end up compromising the identity privacy of the subscriber.

3.6 UMTS-AKA-with-E2EUIC

In this section, we propose a new security extension called End to End User Identity Confidentiality (E2EUIC) [34]. This extension, which is based on our trust model proposed in Chapter 2, Section 2.4, when adopted to the AKA protocol used in a 3GPP mobile system, has the potential to enhance identity privacy and relax trust requirement for roaming. Here, we present this extension with reference to UMTS-AKA (the authentication and key agreement protocol used for access security in UMTS). E2EUIC not only takes care of identity privacy over the wireless network, but goes one step ahead to ensure the same over the wired part as well. It enables mutual authentication without requiring the SNs to have access to the $IMSI$ of the UE. This ensures that even hostile SNs that are placed in between the UE and the HN will not be able to compromise identity privacy, thereby relaxing the need for the HN to trust the SN with respect to identity privacy during roaming agreements, specially when the HN and the SN belongs to two different operators. Thus, we call the proposed extension as End to End User Identity Confidentiality. E2EUIC achieves enhanced identity privacy without forcing any change in the intermediary network. It does not require the UE to transmit its $IMSI$ at any stage of the protocol flow. We propose to replace transmission of the $IMSI$ with a Dynamic Mobile Subscriber Identity ($DMSI$). A fresh $DMSI$ is created as and when its need arises, and

its value is derived from the most recent random number received as a challenge during a successful UMTS-AKA procedure. As a result, transmission of a $DMSI$ does not compromise the permanent identity of the user. The extension can be introduced in the existing system as an optional service, with the subscriber requiring to collect a new USIM in place of his/her existing USIM, or can be introduced on a rolling basis as new USIMs are issued.

In order to enable the UE to create a $DMSI$, a new random number called Random number for Identity Confidentiality (RIC) is introduced in the security extension. The $DMSI$ is a function of this RIC, details of which is explained later in this section.

The HN maintains a pool of $RICs$ in its local database (i.e., HLR/AuC), some of which are in-use (i.e., already assigned to different UEs) and some of which are not-in-use (unassigned) at an instant of time. During every successful run of the UMTS-AKA protocol, a not-in-use RIC, randomly selected from the pool, is securely transferred to the UE. The HN uses this RIC to uniquely identify the UE for an epoch (explained later in this section) of time. The selected RIC has to be sufficiently random, such that there is no correlation with a previously selected RIC. A mapping between the selected RIC and the $IMSI$ of the UE is maintained at the HN's local database (HLR/AuC), so that the HN can uniquely identify the subscriber/UE through this mapping at a later instant (for purposes like billing, generation of AVs, etc.). Thus, whenever the UE needs to present its permanent identity, it assembles a $DMSI$ with the most recently received RIC and transmits it instead of the $IMSI$. The HN in turn, extracts the RIC from the received $DMSI$ and identifies the subscriber by referring to the RIC to $IMSI$ mapping maintained in its database (i.e., HLR/AuC).

In some exceptional situations like failure of an ongoing UMTS-AKA or due to an active attack by an adversary, the UE may not receive the next RIC (from the HN) after it has already used the most recently received RIC to create and transmit a $DMSI$. In such a situation, if the need to transmit a $DMSI$ arises again, the UE can reuse the most recently received RIC to create the next $DMSI$. This can continue, as long as the UE does not receive a fresh RIC from

the HN (during a successful UMTS-AKA). Even though such a mechanism, in the worst case, may allow an adversary to link two or more failed UMTS-AKA of the same UE, an adversary cannot gain anything from this in terms of compromised identity privacy. Moreover, it is a much better option than transmitting the $IMSI$ itself.

To securely transfer a not-in-use RIC to the UE during a run of the AKA protocol (Section 3.3), a fresh not-in-use RIC (RIC_{Fresh}) is selected at the HN. RIC_{Fresh} is then embedded into the $RAND$ part of each AV in $AV[1..M]$ (Equation 3.3.7). For embedding, the long term secret key Ki and an embedding algorithm are used. The resultant random number after embedding a RIC into a $RAND$ is called an Embedded RAND ($ERAND$). Thus, during a run of the UMTS-AKA protocol with E2EUIC extension, an $ERAND$ (which is of the same size as the $RAND$, i.e., 128 bit) is now send as a challenge to the UE instead of a $RAND$. The UE, having knowledge of the long term shared key Ki, can easily extract RIC_{Fresh} from the received $ERAND$. The rest of the AKA procedure continues in the same way as in UMTS-AKA, the sole difference being the use of $ERAND$ in all purposes where the $RAND$ was used earlier. The intermediary networks does not have to bother about this difference, as the size of $ERAND$ and $RAND$ are same, and they can continue to operate as before.

The mechanism proposed in the above paragraph cannot provide a RIC to the UE that is required for identity presentation during the first UMTS-AKA-with-E2EUIC in the USIM's life time. This is because, the first identity presentation precedes all AKAs. Thus, an alternate mechanism is used for this purpose. This mechanism is carried out before distribution of the USIM, i.e., before a subscriber procures the USIM from the mobile operator. According to this mechanism, an $ERAND$ (Say $ERAND_{First}$) that has a unique RIC (say RIC_{First}) embedded into it is stored into the USIM's flash memory. RIC_{First} is meant to be used only for the first successful authentication in the USIM's life time.

We propose the size of RIC to be of 32 bits. Choice of 32 bits for RIC is inspired by the size of $TMSI$ used in UMTS-AKA [5]. With this size, a SN is

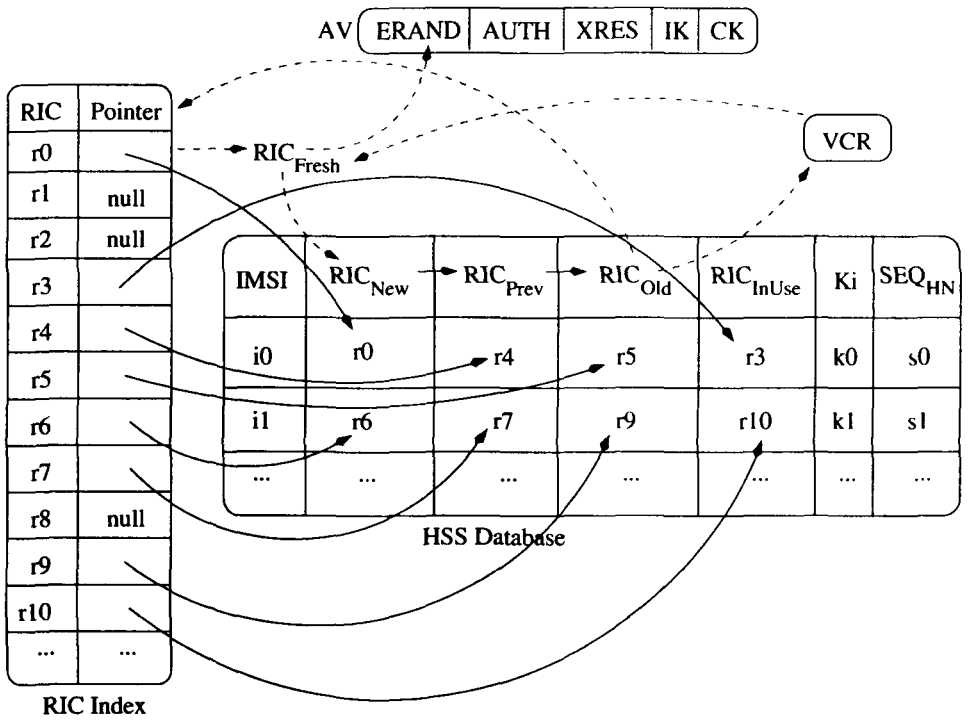

Figure 3.6: Home network's database with RIC-Index.

easily able to allocate unique $TMSIs$ to all the UEs under its service area. A 32 *bit* RIC provides a pool of $2^{32} = 4.29$ billion (approx) unique RIC values. However, size of the RIC may even be determined by the operator depending on the anticipated subscriber base of the HN (provided it is lesser than 128 *bits*). A RIC of size b *bits*, provide a pool of n unique RIC values. Where,

$$n = 2^b \tag{3.6.1}$$

The HN needs to store in its local database multiple (m) $RICs$ against the $IMSI$ of a particular UE (Figure 3.6). Multiple $RICs$ are stored in order to deal with the following exceptional situations:

- *An AV or a sequence of AVs gets lost in transit or does not get utilised*: In such situations, the most recently selected RIC that is stored at the HN's local database against the $IMSI$ of the UE will not reach the UE. Due to which, the UE will continue to consider the RIC that it received during the last successful authentication as the current RIC, using which it may create and transmit the next $DMSI$. Thus, a history of $m-1$ most recent $RICs$

generated against the UE needs to be maintained against its $IMSI$ at the HN's database (in the fields say RIC_{New}, RIC_{Prev}, RIC_{Old}, etc.); the value of m is to be decided by the mobile operator depending on the probability of AVs getting lost in transition. This ensures that a mapping between the RIC that is currently stored at the UE and the corresponding $IMSI$ is always maintained at the HN. However, like any other critical information such as the subscriber's security credentials, billing details, etc., in case of the $RICs$ maintained in the HN's database also, it is the responsibility of the operator to have a robust backup mechanism against database crash. Whenever a new RIC (i.e., RIC_{Fresh}) is generated at the HN, the oldest RIC (i.e., RIC_{Old}) is discarded (returned to the pool of not-in-use $RICs$) and the values stored in the $m-2$ other RIC fields (i.e., RIC_{New}, RIC_{Prev}, etc.) are shifted to their next older fields (i.e., RIC_{Prev} is shifted to RIC_{Old}, RIC_{New} is shifted to RIC_{Prev}, etc.). These adjustments are done to make space for RIC_{Fresh} in the HN's local database.

- *The RIC contained in the DMSI that is being used by the SN to identify the UE gets deleted from the HN's database*:
  When the subscriber enters the service area of a new SN, the very first identity that it uses to identify itself to this SN is a $DMSI$ rather than a $TMSI$ or the $IMSI$. This $DMSI$ is used by the SN to uniquely identify the UE for purposes like billing and collection of AVs. The SN uses this $DMSI$ as long as it does not receive the next $DMSI$ (during a successful authentication) from the UE or till the subscriber does not leave its service area - which ever happens earlier. When the next $DMSI$ is received from the UE, the SN discards the previous $DMSI$ and starts using the newly received $DMSI$ to uniquely identify the UE.

  If a roaming subscriber continuously stays with the SN through several authentications (requiring the SN to collect more than m ordered array of AVs (i.e., $AV[1..M]$) from the HN), a time will come when the RIC contained in the $DMSI$ that is being used by the SN to uniquely identify the UE gets removed (returned to the pool of not-in-use $RICs$) from the

HN's database. Thus, in order to deal with this situation, an additional field called RIC_{InUse} is maintained at the HN's database against the $IMSI$ of the UE. While the $m-1$ other RIC values stored against the $IMSI$ in the fields: RIC_{New}, RIC_{Prev}, RIC_{Old}, etc., keeps changing, the RIC value stored against RIC_{InUse} changes only when a new $DMSI$ is selected by the SN to uniquely identify the UE.

If s is the maximum number of subscribers that a mobile operator wants the proposed extension to handle, then

$$s = n/m \tag{3.6.2}$$

where, n (Equation 3.6.1) is the total number of possible $RICs$ in the entire pool and m is the number of $RICs$ maintained against each $IMSI$ in the HN's database (i.e., HLR/AuC). We propose the value of m to be 4. However, an operator may choose to have a different value for m depending on its anticipated subscriber base. With $m = 4$, a 32 bit RIC will enable the HSS to have at the most $2^{30} = 1.073$ billion (approx) subscribers, which is 5.73 times more than the 187.302 million (approx) subscriber base of the largest mobile operator in India as of June, 2012 [35]).

Before distribution of an USIM, it has to be initialised. During this, all the m RIC fields (maintained in the HN's database against the $IMSI$ of the USIM) are assigned randomly selected not-in-use $RICs$. Out of all the assigned RIC values, the value that is assigned against the RIC_{New} field is chosen as RIC_{First} and is later transferred to the UE as already explained earlier in this section.

In order to verify the freshness of a received $DMSI$ and to prevent replay attacks, the HSS maintains an additional field called SEQ_{HN} against every $IMSI$ in its database. SEQ_{HN} is used to store the sequence number of the most recent $DMSI$ received from the UE.

In order to quickly locate a RIC in the HN's database, a database index called *RIC-index* is maintained at the HN (Figure 3.6). The *RIC-Index* contains all the n possible $RICs$ sorted according to their values. Each entry in the *RIC-Index* contains a pointer against it, which is called an *IMSI-Pointer*. This pointer

either points to an $IMSI$ in the HN's database or is *null*, depending on whether that particular RIC is allocated to an UE or is unallocated at a particular instance of time. The collection of all the $RICs$ in the *RIC-Index* having a *null* value against it, forms the pool of not-in-use $RICs$, whereas the rest of the $RICs$ in the *RIC-Index* that points to some $IMSI$, represents the $RICs$ that are in-use. The total number of entries in the *RIC-Index* is fixed at n, irrespective of the number of $RICs$ that are currently in-use in the HN's database. Even though such an index would require more disk space (Section 6.5), compared to an index whose size grows and shrinks according to the number of $RICs$ that are in-use at a particular instance in the HN's database, it relieves the HN of computational overhead involved during frequent insertions and deletions in the index.

3.6.1 The Protocol Flow

E2EUIC, is implemented in the USIM of the UE and in the HLR/AuC of the HN. In UMTS-AKA-with-E2EUIC, a $DMSI$ is transmitted instead of the $IMSI$ (Figure 3.7). The role of the $TMSI$ remains same as in the original UMTS-AKA. A fresh $DMSI$ is created in the USIM only when its need arises, i.e., during the first AKA in the USIM's life time and when the UE receives a request for the permanent identity from the SN. The current value of the $DMSI$ depends on the most recently received RIC by the UE. A $DMSI$ is a concatenation of the Mobile Country Code (MCC), the Mobile Network Code (MNC), the most recent RIC received by the UE, and an Encrypted RIC called $ERIC$. Since $DMSI$ is calculated using short-lived RIC values, knowledge of the former does not compromise the actual identity of the UE. The protocol flow during the first UMTS-AKA-with-E2EUIC in the life time of a USIM is as follows (a list of the notations used and their brief description are presented in Table 3.1).

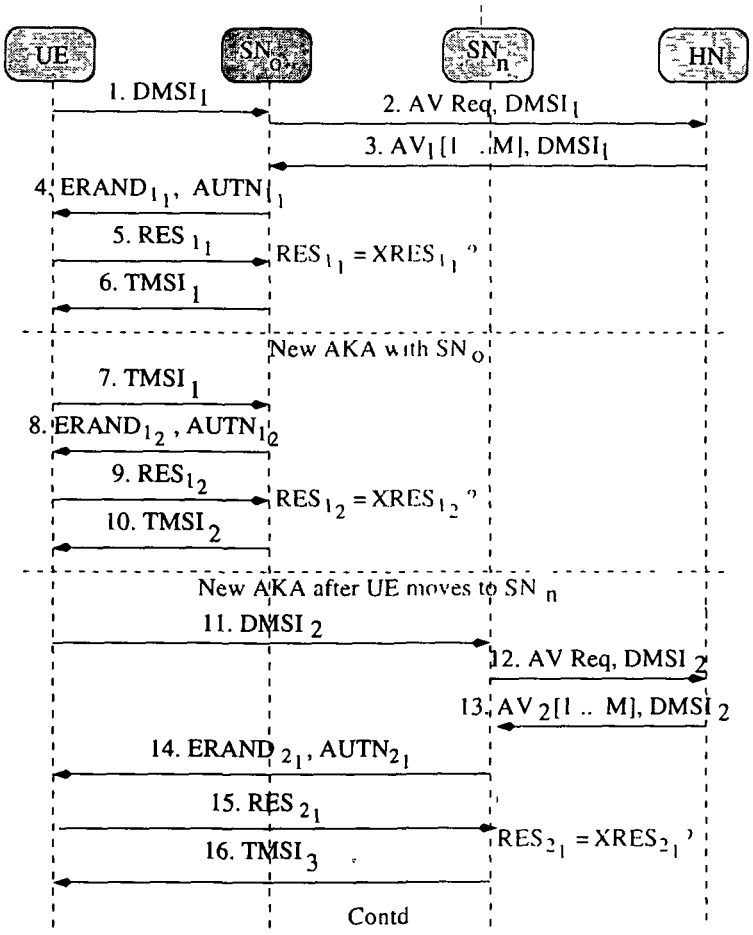

Figure 3.7: Protocol flow of UMTS-AKA-with-E2EUIC.

The First UMTS-AKA-with-E2EUIC

(1.1) The RIC (i.e., RIC_{First}) stored in $ERAND_{First}$ is extracted using an operator specific function $f_{Extract}$.

$$RIC_{First} = f_{Extract_{K_i}}(ERAND_{First}) \tag{3.6.3}$$

An example algorithm for $f_{Extract}$ is presented in Section 3.7. For identity presentation, the UE creates a $DMSI$ (say $DMSI_1$) using RIC_{First} as follows:

$$DMSI_1 = MCC||MNC||RIC_{First}||ERIC \tag{3.6.4}$$

where, $ERIC$ is created by encrypting a padded RIC (say RIC_{padded}) with the Advanced Encryption Standard (AES) algorithm, taking the long term secret key K_i as parameter. Thus,

$$ERIC = f_{Encrypt_{K_i}}(RIC_{padded}) \tag{3.6.5}$$

Table 3.1: Notations.

RIC_{Fresh}	:	A fresh not-in-use RIC.
RIC_{First}	:	A not-in-use RIC embedded in the USIM before the USIM is distributed.
RIC_{New}	:	A RIC maintained against the $IMSI$ at the HN's database.
RIC_{Prev}	:	A RIC maintained against the $IMSI$ at the HN's database.
RIC_{Old}	:	A RIC maintained against the $IMSI$ at the HN's database.
RIC_{InUse}	:	A RIC maintained at the HN's database against the $IMSI$, a copy of which is currently being used at the SN to identify the UE.
SEQ_{UE}	:	Value of a $DMSI$ counter maintained at the UE.
SEQ_{HN}	:	Sequence number maintained at the HN to check freshness of a received $DMSI$.
R	:	128-(32+b) bit random number.
RIC_{Padded}	:	$RIC\|\|SEQ_{UE}\|\|R$; where, $\|\|$ indicates concatenation.
$ERAND_{UE}$	:	Variable maintained in the USIM's flash memory to store the most recently received $ERAND$.
$DMSI_{SN}$	:	Variable maintained in the SN to uniquely identify the UE as long as it does not receive a new $DMSI$ (during a successful AKA).
TTL_{DMSI}	:	Time to live for $DMSI$.
f_{Embed}	:	Embeds a RIC into a $RAND$ to find an $ERAND$.
$f_{Extract}$	:	Used to extract an embedded RIC from an $ERAND$.
$f_{Encrypt}$	:	An AES standard encryption algorithm that encrypts RIC_{Padded} to find an $ERIC$.
$f_{Decrypt}$	:	An AES standard decryption algorithm that decrypts an $ERIC$ to find RIC_{Padded}.
f_{PRNG}	:	A pseudorandom number generator.
m	:	Total number of $RICs$ maintained at the HN's database against a particular $IMSI$.
b	:	Number of bits in a RIC.
n	:	Total number of $RICs$ in the pool of $RICs$ maintained at the HN.
s	:	Maximum number of subscribers that a mobile operator wants to accommodate.

where,

$$RIC_{padded} = RIC_{First} || SEQ_{UE} || R \tag{3.6.6}$$

SEQ_{UE} is the value of a 32 bit counter that is maintained at the UE (USIM's flash memory); whenever a new $DMSI$ is created for identity presentation, SEQ_{UE}'s value is incremented by one. R is a $128 - (32 + b)$ bit random number. The inclusion of SEQ_{UE} ensures freshness of the $DMSIs$, whereas the inclusion of R completes the block size of 128 bits that is necessary to be fed into the AES cipher. In addition, R introduces significant amount of randomness to harden cryptanalysis of the cipher text.

Finally, $DMSI_1$ is transmitted to the SN.

In this step, we selected AES algorithm for the purpose of creating ERIC, due to the following reasons:

- AES (originally called Rijndael) is fast in both software and hardware [36]
- The example algorithm set presented by 3GPP in [33] is based on Rijndael that eventually got selected as AES standard [37]; Rijndael as being then one of the five remaining AES candidates, was well studied.
- Until May 2009, the only successful published attacks against the full AES were side-channel attacks on some specific implementations [38].

Depending on availability of better functions, this choice could be replaced by any suitable 128-bit keyed function employing a 128 bit key.

(1.2) The SN temporarily stores $DMSI_1$ in its local database, as it would require this $DMSI$ to uniquely identify the UE for a period of time (till it receives the next $DMSI$ from the UE during a successful UMTS-AKA) if this run of the UMTS-AKA eventually succeeds. The SN identifies the HN of the UE by inspecting the MCC and MNC portion of $DMSI_1$. A request for a fresh set of AV is then sent along with $DMSI_1$ to the HN.

(1.3) On receiving the request, the HN separates RIC_{First} from $DMSI_1$. It then uses the *RIC-Index* to locate RIC_{First} and hence the $IMSI$ and the key Ki of the UE in the HN's database (Figure 3.6). After this, the $ERIC$ part of the $DMSI$ is decrypted using AES and the key Ki. Thus,

$$RIC_{Padded} = f_{Decrypt_{Ki}}(ERIC) \tag{3.6.7}$$

The RIC contained in RIC_{Padded} is then compared with the RIC part of $DMSI_1$ (i.e., RIC_{First}); success of this comparison ensures that a malicious agent did not create $DMSI_1$. The SEQ_{UE} part of RIC_{Padded} is then compared with the value stored against SEQ_{HN} field in the HN's database (HLR/AuC). If $SEQ_{UE} > SEQ_{HN}$, the request is proven as fresh (not a replay of any previous request). Failure of any of these two comparisons, leads to rejection of the request. If the request for AV is found to be fresh and from a genuine source (from the above two comparisons), the following are performed:

(a) SEQ_{UE} is assigned to SEQ_{HN}.

$$SEQ_{HN} = SEQ_{UE} \tag{3.6.8}$$

(b) If RIC_{First} is stored in any of the RIC fields other than RIC_{InUse} in the HN's/HSS's database, RIC_{First} is moved from its current location (say L) to RIC_{InUse} and the RIC stored earlier against RIC_{InUse} is moved to L. In other words, L and RIC_{InUse} swaps their values. For example: if RIC_{First} is found in RIC_{New} then:

$$temp = RIC_{InUse} \tag{3.6.9}$$

$$RIC_{InUse} = RIC_{New} \tag{3.6.10}$$

$$RIC_{New} = temp \tag{3.6.11}$$

This is done to ensure that a mapping between RIC_{First} and the $IMSI$ of the UE is maintained in the HN's database, as long as $DMSI_1$ is used by the SN to uniquely identify the subscriber; in other words, to ensure that RIC_{First} doesn't get removed from the

HN's database, while $DMSI_1$ is still being used by the SN to uniquely identify the subscriber. The value of RIC_{InUse} does not change till a new genuine request for authentication data along with a new $DMSI$ (with a RIC value that is different from the one which is stored in RIC_{InUse}) does not reach the HN. RIC values stored against all the $m-1$ other RIC fields, viz., RIC_{Old}, RIC_{Prev}, RIC_{New}, etc., eventually gets removed from the HN's database after generation of m $AV[1..M]s$.

(c) A fresh array of AVs (say $AV_1[1..M]$) is generated using the procedure used in UMTS-AKA (Equation 3.3.7).

After this, HN selects a fresh not-in-use RIC (say RIC_{Fresh}) from the pool of $RICs$ (*RIC-Index*). In order to select RIC_{Fresh}, a b *bit* random number (say RN) is generated using a standard Pseudo Random Number Generator (PRNG). For this, we propose to use National Institute of Standards and Technology (NIST) recommended random number generator based on ANSI X9.31 Appendix A.2.4 Using AES [39], which appears in the list of approved random number generators for Federal Information Processing Standards Publication (FIPS PUB) 140-2 [40]. With a 128 *bit* key, this PRNG generates a 128 *bit* random number, the b most significant *bits* of which is selected as RN.

$$RN = f_{PRNG}(seed) \tag{3.6.12}$$

This RN is then searched for in the *RIC-Index*. If the *IMSI-Pointer* against RN in the *RIC-Index* is found to be *null*, RN is selected as RIC_{Fresh} and the *null* value is replaced with the address of the record in the HSS's database where the $IMSI$ is stored.

$$RIC_{Fresh} = RN \tag{3.6.13}$$

$$RN.IMSI\text{-}Pointer = Address\ of\ IMSI \tag{3.6.14}$$

The oldest RIC value (i.e., RIC_{Old}) stored against the $IMSI$ is then returned to the pool of not-in-use RIC by searching for it in the *RIC-Index*

and by setting the *IMSI-Pointer* against it to *null*.

$$RIC_{Old}.IMSI\text{-}Pointer = null \tag{3.6.15}$$

In case the *IMSI-Pointer* against RN in the *RIC-Index* is not *null*, it may be inferred that there is a collision, and RN is currently in-use. For collision resolution, a b *bit* variable called Variable for Collision Resolution (VCR) is used (Figure 3.6). The VCR contains a not-in-use RIC; an indication of this fact is specified in the *RIC-Index* by setting the *IMSI-Pointer* against the value in VCR to the address of VCR. At the very outset, during initialisation of the HSS's database, a b *bit* random number (say RN_0) is stored in the VCR and the *IMSI-Pointer* against it in the *RIC-Index* is set to the address of VCR.

$$RN_0 = f_{PRNG}(seed) \tag{3.6.16}$$

$$VCR = RN_0 \tag{3.6.17}$$

$$RN_0.IMSI\text{-}Pointer = Address\ of\ VCR \tag{3.6.18}$$

Whenever there is a collision, the b *bit* value stored in the VCR is selected as RIC_{Fresh}. VCR is then searched for in the *RIC-Index* and the *IMSI-pointer* against it in the *RIC-Index* is made to point to the record in the HSS's database where the $IMSI$ is stored.

$$RIC_{Fresh} = VCR \tag{3.6.19}$$

$$VCR.IMSI\text{-}Pointer = Address\ of\ IMSI \tag{3.6.20}$$

In order to replace the RIC stored in the VCR with a fresh RIC, the oldest RIC (i.e., RIC_{Old}) stored against the $IMSI$ is copied into VCR. RIC_{Old} is then searched for in the *RIC-Index* and the *IMSI-pointer* against it is set to the address of VCR.

$$VCR = RIC_{Old} \tag{3.6.21}$$

$$RIC_{Old}.IMSI\text{-}Pointer = Address\ of\ VCR \tag{3.6.22}$$

Software heuristics for generating empirically strong random number sequences rely on entropy gathering by measuring unpredictable external

events [41]. The above procedure used to refresh the VCR introduces ample entropy to make the selection procedure of RIC sufficiently random, because it is impossible to predict which $IMSI$'s RIC_{Old} value will refresh the VCR during the next AKA at the HN. It solely depends on the call timing and usage pattern of all the active subscribers registered with the HN.

RIC_{Fresh} is then embedded into the $RAND$ part of all the M AVs of $AV_1[1..M]$ using an operator specific function f_{Embed}. We propose an example algorithm for f_{Embed} in Section 3.7. We call the resultant number after embedding RIC_{Fresh} into a $RAND$ as an $ERAND$. Thus, all the M $ERANDs$ for $AV_1[1..M]$ are derived as follows:

$$ERAND_{1_x} = f_{Embed}(RIC_{Fresh}, RAND_{1_x}) \tag{3.6.23}$$

where, $x = 1, 2, 3, ...M$. Therefore, each AV quintet of $AV_1[1..M]$ will now have an $ERAND$ in it, instead of a $RAND$.

$$AV_{1_x} = (ERAND_{1_x}, XRES_{1_x}, CK_{1_x}, IK_{1_x}, AUTN_{1_x}) \tag{3.6.24}$$

where, $x = 1, 2, 3, ...M$. From now on, an $ERAND$ is used for all purposes where a $RAND$ is used in UMTS-AKA (this will not have any impact on the protocol flow, as the size of $RAND$ and $ERAND$ are same (128 bit)). A copy of RIC_{Fresh} is also stored at the HN's database against the $IMSI$ of the subscriber. For this purpose, RIC_{Old} is replaced by RIC_{Prev}, RIC_{Prev} is replaced RIC_{New} and so on. And finally, RIC_{New} is replaced by RIC_{Fresh}. An entry in the *RIC-Index* against the *IMSI-Pointer* of RIC_{Fresh} is also made accordingly. Thus,

$$RIC_{Old} = RIC_{Prev} \tag{3.6.25}$$

$$RIC_{Prev} = RIC_{New} \tag{3.6.26}$$

$$RIC_{New} = RIC_{Fresh} \tag{3.6.27}$$

$$RIC_{Fresh}.IMSI\text{-}Pointer = address\_of(IMSI) \tag{3.6.28}$$

Finally, HN sends $AV_1[1..M]$ along with $DMSI_1$ back to the SN.

(1.4) On receipt, SN continues the AKA procedure by extracting the $ERAND$ and $AUTN$ part of the first unused AV of $AV_1[1..M]$ (i.e., $ERAND_{1_1}$ and $AUTN_{1_1}$). $ERAND_{1_1}$ and $AUTN_{1_1}$ are then transmitted as a challenge to the UE.

(1.5) The UE and the SN completes the remaining part of the UMTS-AKA-with-E2EUIC extension, following the same steps as in UMTS-AKA. On successful completion of the mutual authentication process, the following additional steps are carried out by the UE and the SN:

(a) The UE saves the recent $ERAND_{1_1}$ that it received from the SN in a field (say $ERAND_{UE}$) in the USIM's flash memory.

(b) The SN stores $DMSI_1$ in a variable say $DMSI_{SN}$ in its local database. The value of this variable does not change till the SN does not receive a new $DMSI$ during a successful UMTS-AKA-with-E2EUIC. The SN uses $DMSI_{SN} = DMSI_1$ to uniquely identify the UE as long as it does not receive a new $DMSI$. The same field that the SN uses to store the $IMSI$ can be used as $DMSI_{SN}$.

(c) The remaining AVs in $AV_1[1..M]$ are stored against $DMSI_{SN}$ at SN's database for future authentications.

(1.6) At the end of the AKA procedure, a pair of Cipher Key (CK) and an Integrity Key (IK) is established between the UE and the SN, following the same procedure as in UMTS-AKA (Equation 3.3.2 and Equation 3.3.3). A secure and reliable channel is then created between the UE and the SN using these two keys. After this, a Temporary Mobile Subscriber Identity (say $TMSI_1$) generated by the SN is securely communicated to the UE through this channel. The UE stores the received $TMSI_1$ in the USIM's flash memory (in a field say $TMSI_{UE}$) for identity presentation during the next authentication. A mapping between $TMSI_1$ and $DMSI_{SN}$ is also maintained in the SN's local database. If the UE uses $TMSI_1$ to identify itself in the next authentication, the $TMSI$-to-$DMSI_{SN}$ mapping helps to SN to locate/acquire an AV that is needed for the authentication.

Subsequent Authentications

During all subsequent communications and the corresponding mutual authentications involved therein, the UE may present its identity in two different ways. Both these ways are listed below in their order of preference:

(i) *By transmitting a TMSI received in the previous AKA:*

In this method, the UE transmits the most recently received $TMSI$ (i.e., the $TMSI$ stored in $TMSI_{UE}$). When the SN receives a $TMSI$, it locates the corresponding $DMSI_{SN}$ using the $TMSI$-to-$DMSI_{SN}$ mapping maintained in its database. It then initiates the authentication procedure using an unused AV stored against $DMSI_{SN}$ in its local database (if there is any). If there is no unused AV, the SN will have to acquire a fresh set of AV (i.e., $AV[1..M]$) from the HN by presenting its $DMSI_{SN}$.

Out of the two methods, this is the preferred choice for the UE, because, it reduces communication latency during authentication. Specifically, whenever there is an unused AV at the SN, the authentication happens locally between the UE and the SN, without needing the SN to communicate with the HN. The protocol flow for subsequent authentications through the transmission of $TMSI$ is as follows:

(2.1) The UE extracts $TMSI_{UE}$ from its memory (USIM's flash memory) and transmits it to the SN.

(2.2) Through this $TMSI$, the SN identifies the corresponding $DMSI_{SN}$ and hence the authentication vectors (i.e., $AV[1..M]$) that are stored against $DMSI_{SN}$. If there is no unused AV in $AV[1..M]$, SN sends a request for a fresh set of AVs along with the $DMSI_{SN}$ to the HN. In case there is an unused AV in $AV[1..M]$, the next step (i.e., step 2.3) is skipped.

(2.3) After receiving the request, the HN separates the RIC part of $DMSI_{SN}$. The *IMSI-Pointer* against this RIC leads to the record in the HN's database that contain details related with the corresponding $IMSI$ of the UE. The remaining portion of this step proceeds in the same

manner as in step 1.3.

(2.4) The remaining part of the protocol flow is same as steps 1 4 through 1.6.

(ii) *By transmitting a fresh DMSI:*

In this method, the UE transmits a fresh $DMSI$ that is created using the RIC extracted (using $f_{Extract}$) from the most recent $ERAND$ received by the UE. This method of identity presentation is performed only during the following situations:

- *The SN cannot identify the UE with its current $TMSI$:* This may happen if the $TMSI$-to-$DMSI_{SN}$ mapping is lost from the SN's database.
- *The subscriber moves from an old SN (say SN_o) to the service area of a new SN (say SN_n):* In UMTS-AKA, the first identity presentation under the service area of SN_n happens through transmission of a $TMSI$ (say $TMSI_o$) allotted to the UE by SN_o and the Location Area Identity (LAI) of SN_o. Unlike UMTS-AKA, in UMTS-AKA-with-E2EUIC, the first identity presentation under the service area of SN_n happens through transmission of a fresh $DMSI$. This, makes the following two messages of UMTS-AKA redundant:

  (a) transmission of $TMSI_o$ from SN_n to SN_o.

  (b) transmission of $IMSI$ and $TMSI_o$ from SN_o to SN_n.

  The above two messages enables SN_n to learn the $IMSI$ of the subscriber from SN_o. SN_n uses the received $IMSI$ to collect AVs from the HN. In UMTS-AKA-with-E2EUIC, SN_n can directly collect AVs from the HN (without communicating SN_o) by presenting the received $DMSI$ to the HN, thereby improving communication latency.
- *Time To Live for DMSI (TTL_{DMSI}) has expired:* The SN uses the same $DMSI$ stored in $DMSI_{SN}$ to uniquely identify the subscriber as long the subscriber continues to stay within the SN's service area or till the SN does not receive another fresh $DMSI$ from the UE, which ever happens earlier. However, this will allow a SN with malicious in-

tention to link two or more connections of the same subscriber through the value of $DMSI_{SN}$, though it will not be possible to exactly know which particular subscriber it is. Thus, in order to prevent the SN from linking several communications of the same subscriber, it becomes important to limit the lifetime of the $DMSI$ stored in $DMSI_{SN}$. For this purpose, a field called Time To Live for DMSI (TTL_{DMSI}) is introduced in the USIM's flash memory Maximum value of TTL_{DMSI} is to be decided by the operator (operator specific). Immediately after transmitting a freshly generated $DMSI$, the UE resets TTL_{DMSI} to its maximum value. The value of TTL_{DMSI} is decremented by one with the tick of every second. Next time whenever the UE has to transmit its identity, firstly the value of TTL_{DMSI} is checked. If TTL_{DMSI} is found to be greater than zero, the UE transmits a $TMSI$. Otherwise, if TTL_{DMSI} is found to be equal to zero, a fresh $DMSI$ is computed and transmitted. This forces the SN to periodically refresh $DMSI_{SN}$ even if the UE chooses to stay with the same SN for a long duration.

The E2EUIC protocol flow for subsequent authentications through the transmission of a $DMSI$ is same as that of the first mutual authentication in UMTS-AKA-with-E2EUIC (i.e., steps 1.1 through 1.6). The only difference is that the RIC contained in the most recently received $ERAND$ is used in this case by the UE to create a $DMSI$, rather than RIC_{First}.

3.6.2 Strengths

Some of the strong points of UMTS-AKA-with-E2EUIC are as follows:

- End to end user identity privacy: Knowledge of $IMSI$ is confined only to the UE and the HN; it is never transmitted at any stage of the protocol flow and at any portion of the path between the UE and the HN.
- Relaxed trust requirement: UE-to-SN as well as HN-to-SN trust relationship requirement with respect to permanent identity is relaxed. Such trust

relaxation simplifies roaming agreements between operators.

- Reduced number of message exchanges: If the extension is adopted, two protocol messages of UMTS-AKA becomes redundant. This will improve communication latency during authentication and key agreement.
- Fast database access: The *RIC-Index* makes searching through the HN's database faster.
- Minimal impact on the SN: For the extension to be adopted, most of the modifications are performed at the USIM and at the HN, very negligible amount of adjustment, of that of treating a received $DMSI$ as an $IMSI$, is required at the SN. This, makes the extension easier to adopt for the operators, since an operator has to do the necessary modifications only at the HN's database and at the USIMs of the subscribers.

3.7 Example Algorithms for f_{Embed} and $f_{Extract}$

In this section, we present an example algorithm to implement the cryptographic functions f_{Embed} and $f_{Extract}$, which otherwise is operator specific [42]. If found appropriate, an operator may choose to use this algorithm, otherwise it may have its own implementation. f_{Embed} embeds a 32 bit RIC into a 128 bit $RAND$ using the secret key Ki to produce a 128 bit $ERAND$ (Equation 3.6.23), where as $f_{Extract}$ extracts the embedded RIC from an $ERAND$ using Ki as parameter (Equation 3.6.3).

The first step of the algorithm is to generate the following two sets of 32 element integer arrays from the secret key Ki:

$$A_{pos} = \{X_1, X_2, ..., X_{32}\} \tag{3.7.1}$$

$$A_{XOR} = \{Y_1, Y_2, ..., Y_{32}\} \tag{3.7.2}$$

such that

$$X_i \neq X_j; X_i \neq Y_j; Y_i \neq Y_j$$

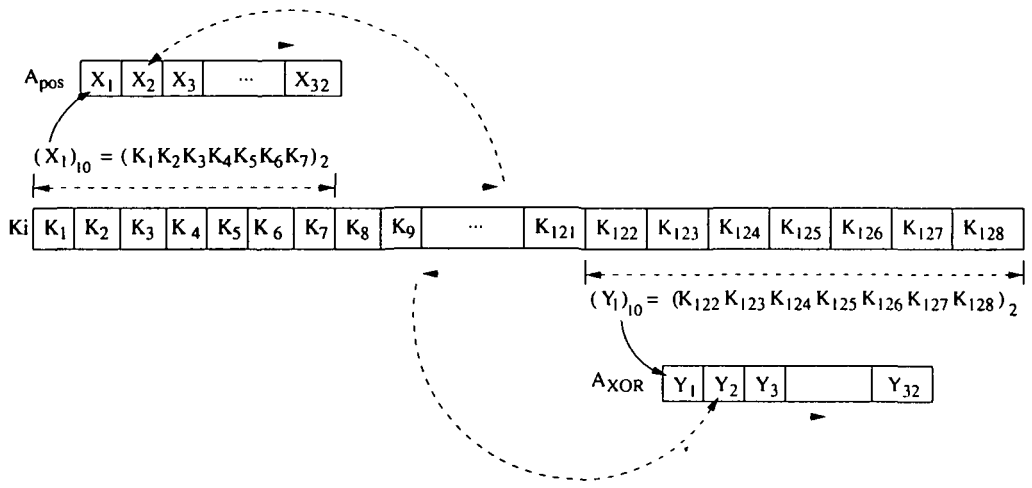

Figure 3.8: Generation of A_{pos} and A_{XOR}.

where

$$i, j \in \{1, 2, 3, ..., 32\}$$

and

$$X_i, X_j, Y_i, Y_j \in \{1, 2, ..., 128\}$$

The elements of A_{XOR} are used to locate 32 unique bit positions in the 128 bit $RAND$, the bit values in these positions are used to mask the 32 bits of RIC. The masking is achieved by performing bit-wise XOR operation between the 32 bits determined by A_{XOR} and the 32 bits of RIC. The elements of A_{pos} determine 32 other unique bit positions of $RAND$ that will be replaced by the masked bits of RIC. Finally, the resultant 128 bit number generated after inserting the masked bits is encrypted using AES cipher to find $ERAND$. A_{pos} and A_{XOR} are generated as follows:

To generate A_{pos}, the bits of $Ki(k_1, k_2, ...k_{128})$ are grouped into collection of seven, starting from left to right.

$$(k_1k_2k_3k_4k_5k_6k_7), (k_8k_9k_{10}k_{11}k_{12}k_{13}k_{14}).....\qquad (3.7.3)$$

Each group yields a decimal number in the range of 1 to 128 (Figure 3.8). 7 bit groupings are used since 2^7 provide 128 possibilities, corresponding to the 128 bit positions of RAND. At the end of the first scan, 18 such groups will be formed $((k_1k_2k_3k_4k_5k_6k_7), (k_8k_9k_{10}k_{11}k_{12}k_{13}k_{14}), ... (k_{120}k_{121}k_{122}k_{123}k_{124}k_{125}k_{126}))$ leaving the two least significant bits k_{127} and k_{128} ungrouped. Thus, there will be

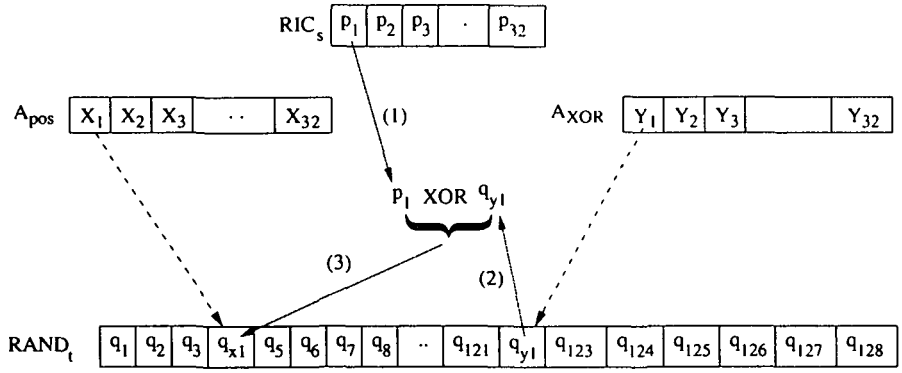

Figure 3.9: Embedding of RIC into $RAND$.

18 decimal numbers in the range of 1 to 128 after the first scan, which may or may not be unique. Only the unique numbers are used to populate A_{Pos}. In the next scan, the left out two least significant bits are grouped with the five most significant bits of Ki to form the next group of seven, successive groups are created starting from the 6th most significant bit.

$$(k_{127}k_{128}k_1k_2k_3k_4k_5), (k_6k_7k_8k_9k_{10}k_{11}k_{12}), ... \tag{3.7.4}$$

This scanning process is continued in cyclic manner, till 32 unique decimal numbers are available to populate A_{pos}.

To generate A_{XOR}, a similar process is followed, except that the scan in this case proceeds from right to left. Every time an integer is generated, it is checked with the elements of A_{pos}, as well as the filled in elements of A_{XOR} for uniqueness; only the unique ones being selected to populate A_{XOR}.

Since the key Ki is a long time shared key, A_{pos} and A_{XOR} needs to be generated only once for every UE. If a key Ki cannot be used to generate 64 unique integers by the above process, we refer it to be an unusable key. An unusable key should not be allocated to an E2EUIC enabled USIM.

In the next step, the XOR operations are performed (Figure 3.9). Let us denote a particular 32 bit RIC and a particular 128 bit $RAND$ as follows:

$$RIC_s = (p_1, p_2, .. , p_{32}) \tag{3.7.5}$$

$$RAND_t = (q_1, q_2, ..., q_{128}) \tag{3.7.6}$$

The decimal number stored against X_1 (in A_{pos}) determines the bit position in $RAND_t$ where p_1 is to be inserted. And, the decimal number stored against Y_1 (in A_{XOR}) determines the bit value in $RAND_t$ that has to be XORed with p_1 before it is inserted. For example, if $X_1 = 20$, $Y_1 = 96$ and $p_1 = 1$ then a XOR operation is performed between p_1 and the 96th bit of $RAND_t$. The resultant bit is then inserted into the 20th bit of $RAND_t$.

$$q_{X1} = q_{Y1} \oplus p_1 \tag{3.7.7}$$

where $\oplus$ is bit wise XOR operation. Thus, in order to insert all the 32 bits of RIC_s into $RAND_t$, the following is to be performed:

$$q_{X_i} = q_{Y_i} \oplus p_i \tag{3.7.8}$$

where i=1,2,3,...32. The resultant 128 bit sequence after inserting all the 32 bits of RIC_s into $RAND_t$ is called a Transformed $RAND$ (say $TRAND_t$ in this case). $TRAND_t$ may be represented as follows:

$$TRAND_t = (u_1, u_2, .. , u_{128}) \tag{3.7.9}$$

Finally, an $ERAND$ is produced by encrypting $TRAND_t$ using the block cipher AES (f_{AES}) with the secret key Ki as parameter.

$$ERAND_t = f_{AES_{K_i}}(TRAND_t) \tag{3.7.10}$$

At the UE's end, RIC_s may be extracted back from $ERAND_t$ using a similar process. First of all, $TRAND_t$ is extracted from $ERAND_t$ using f_{AES}.

$$TRAND_t = f_{AES_{K_i}}(ERAND_t) \tag{3.7.11}$$

Since, we have the property that

$$A = B \oplus C \implies C = A \ominus B \tag{3.7.12}$$

thus, each bit of RIC_s can be extracted from $TRAND_t$ as follows:

$$p_i = u_{Y_i} \oplus u_{X_i} \tag{3.7.13}$$

where i=1,2,3,...32.

Table 3.2: Percentage of unusable keys.

Iterations	Unusable keys	Time (ms)	Percentage
1	0	4	0
10	0	21	0
100	0	96	0
1000	1	376	0.100
10000	15	1282	0.150
100000	129	9476	0.129
1000000	1465	90016	0.146

A single RIC (i.e., RIC_{Fresh}) is embedded into the same bit positions (determined by A_{Pos}) of all the RANDs in $AV[1..M]$. The XOR operations are performed in order to mask the bit values in these positions, and to ensure that no two random numbers of $AV[1..M]$ have any predictable pattern in them. The final encryption through AES is carried out to further shuffle all the 128 bits so that it becomes extremely hard for an adversary to predict a RIC that is embedded into an $ERAND$.

3.7.1 Usability of a Key

If 64 unique decimal numbers in the range of 1 to 128 can be extracted from a key Ki, we call this key an usable key for the example algorithm proposed in this section. Otherwise, we call it an unusable key. Thus, in order to use the example algorithm proposed in this section, the usability of a key Ki should be properly verified before assigning it to any USIM.

We wrote a program in Java to test the usability of a series of 128 bit random numbers (keys), generated using SecureRandom class of java. Every single iteration in the program was made to generate a 128 bit random number and to check its usability, on the basis of whether 64 unique decimal numbers in the range 1 to 128 can be generated. We executed the program for varied number of iterations and recorded the number of unusable keys vis-a-vis the total number

of keys generated, and the amount of time taken for each run. Our findings are listed in Table 3.2.

As evident from the results, the number of unusable keys generated is extremely small compared to the total number of keys. In our experiments, the percentage of unusable key vis-a-vis total number of keys never exceeded 0.15%. It also illustrates that the time consumed for such evaluation of keys is not very large. Moreover, in practice the evaluation of the usability of a key will not be done in real time.

3.7.2 Test for Randomness

In UMTS-AKA-with-E2EUIC, $ERAND$ plays a pivotal role. Thus, randomness of $ERAND$ is a vital issue. While randomness of $RAND$ and RIC depends on operator specific random number generators and the call pattern of all the registered users respectively, the randomness of $ERAND$ that depends on the algorithm used to implement $f_{Embed}/f_{Extract}$, needs to be verified. In this subsection, we use a statistical test suite called the Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications provided by National Institute of Standards and Technology (NIST), to analyse the randomness of $ERANDs$ that are generated using the example algorithm proposed in this section.

The NIST Test Suite is a statistical package that provides a list of tests to test the randomness of random numbers produced by cryptographic random or pseudo-random number generators. These tests focus on a variety non-randomness that could exist in a number. For our analysis, we have used seven of these tests (selected arbitrarily), namely, the frequency (mono-bit) test, frequency test within a block, the runs test, tests for the longest-run-of-ones in a block, the linear complexity test, the approximate entropy test and the cumulative sums (cu-sums) test. Details about all these tests are presented in [43].

As stated in NIST's Special Publication 800-22 [43], a statistical test is formulated to test a specific null hypothesis (H_0). Here, the null hypothesis is that a random number being tested is random. Each test is based on a calculated test

statistic value, which is a function of the random number being tested. The test statistic is used to calculate a P-value. If the P-value for a test is equal to 1, then a random number is considered to have perfect randomness. A P-value of zero indicates that a random number is completely non-random. A significance level (α) is chosen for the tests. If P-value $\geq \alpha$, then the null hypothesis is accepted; i.e., the random number being tested is considered to be random. If P-value $< \alpha$, then the null hypothesis is rejected; i.e., the random number being tested is considered to be non-random. Typically, α is chosen in the range [0.001, 0.01].

For statistical analysis of $ERAND$, we use the strategy adopted by NIST that consists of five stages [43]:

1. *Selection of a Generator*:

   We have used Sun's SHA1 Pseudo Random Number Generator (PRNG) - 'SecureRandom.class', available in 'java.security' package to generate the random numbers used in the analysis.

2. *Binary Sequence Generation*:

   We wrote a program in Java called 'GeneratePRN.java' that generates 1000 random numbers (128-bit) and stores them in a file called 'RAND.txt'. Each of the random numbers in this file represents a RAND. We wrote another program called 'EmbedRAND.java' that generates a fresh 32 bit random number (to represent RIC) and embeds it into each 128 bit random number of 'RAND.txt' (using our proposed example algorithm for $f_{Embed}/f_{Extract}$). The resultant 1000 $ERANDs$ (128-bit) are written into another file called 'ERAND.txt'. Each of the random numbers in 'ERAND.txt' represents an $ERAND$.

3. *Execute the Statistical Test Suite*:

   Using the test suite, each of the selected seven tests were performed on the random numbers written in the file 'ERAND.txt' (by passing 'ERAND.txt' as parameter to the test suite).

4. *Empirical Results*:

   Each statistical test generates empirical results that consists of test statistics

and p-values against each of the random numbers stored in 'ERAND.txt'. An output file is generated by the test suite with the empirical results for each of the statistical tests written in it. Based on these results, a conclusion regarding quality of the sequence of random numbers generated by the proposed example algorithm can be made.

5. *Interpretation of Empirical Results*:

   NIST has adopted two approaches for interpretation of empirical results. In the event that either of these approaches fail, the corresponding null hypothesis must be rejected. Here, the null hypothesis (H_0) is that the sequence of random numbers (stored in the file 'ERAND.txt') being tested is random. In the following portion of this subsection, we carry out both these approaches to analyse randomness of the sequence of random numbers stored in 'ERAND.txt'.

   (i) *Proportion of Random Numbers Passing a Test*: The proportion of random numbers passing a statistical test is the ratio between the number of random numbers whose p-values $\geq \alpha$ and the total number of random numbers (say t) present in the sequence of random numbers being tested. For example, if a statistical test tests 1000 random numbers (i.e., $t = 1000$), with the significance level $\alpha = 0.01$, and with 996 random numbers having P-values ≥ 0.01, then the proportion of random numbers passing the test is $\frac{996}{1000} = 0.9960$. The proportion of random numbers passing a statistical test for all the seven selected tests performed on 'ERAND.txt', considering $\alpha = 0.01$, are calculated and is listed in Table 3.3.

   For a fixed significance level (α), a certain proportion of P-values generated by a particular test are expected to fail. For example, if the significance level is chosen to be 0.01 (i.e., $\alpha = 0.01$), then about 1% of the random numbers are expected to fail. Taking this into consideration, NIST has determined the following range of acceptable proportions, in

Table 3.3: Proportion of random numbers that pass a test.

Statistical Test	***P-values*** $\geq \alpha$
1. Frequency (Mono-bit) Test	995
2. Frequency Test within a Block	996
3. Runs Test	994
4. Tests for the Longest-Run-of-Ones	988
5. Linear Complexity Test	981
6. Approximate Entropy Test	995
7. Cumulative Sums (Cu-sums) Test	998

its Special Publication 800-22 [43]:

$$A = \hat{p} \pm \sqrt[3]{\frac{\hat{p}(1-\hat{p})}{t}} \tag{3.7.14}$$

where, $\hat{p} = 1 - \alpha$ and t is the sample size. If the proportion of random numbers that pass a test falls outside of this interval, then there is evidence that the sequence of random numbers being tested is non-random. For analysis of the sequence of random numbers stored in 'ERAND.txt', if $\alpha = 0.01$ and $t = 1000$, the range of acceptable proportion is:

$$\begin{aligned} A &= 0.99 \pm \sqrt[3]{\frac{0.99(0.01)}{1000}} \\ &= 0.99 \pm 0\ 0094392 \end{aligned} \tag{3.7.15}$$

Fig. 3.10 provides a graphical representation of the proportion of successful random numbers against each statistical test performed on the random numbers stored in 'ERAND.txt'. Since the proportions for all the tests lie within the range of acceptable proportions (i.e., between 0.9805608 and 0.9994392), the sequence of random numbers present in the file 'ERAND.txt' can be considered to be random.

(ii) *Uniform Distribution of P-values*: The distribution of *P-values* is examined to ensure uniformity. In this method (proposed by NIST in [43]), the interval between 0 and 1 is divided into 10 sub-intervals (C1, C2, ...C10), and the *P-values* obtained by performing a particular statistical test on a sequence of random numbers, which lie within each

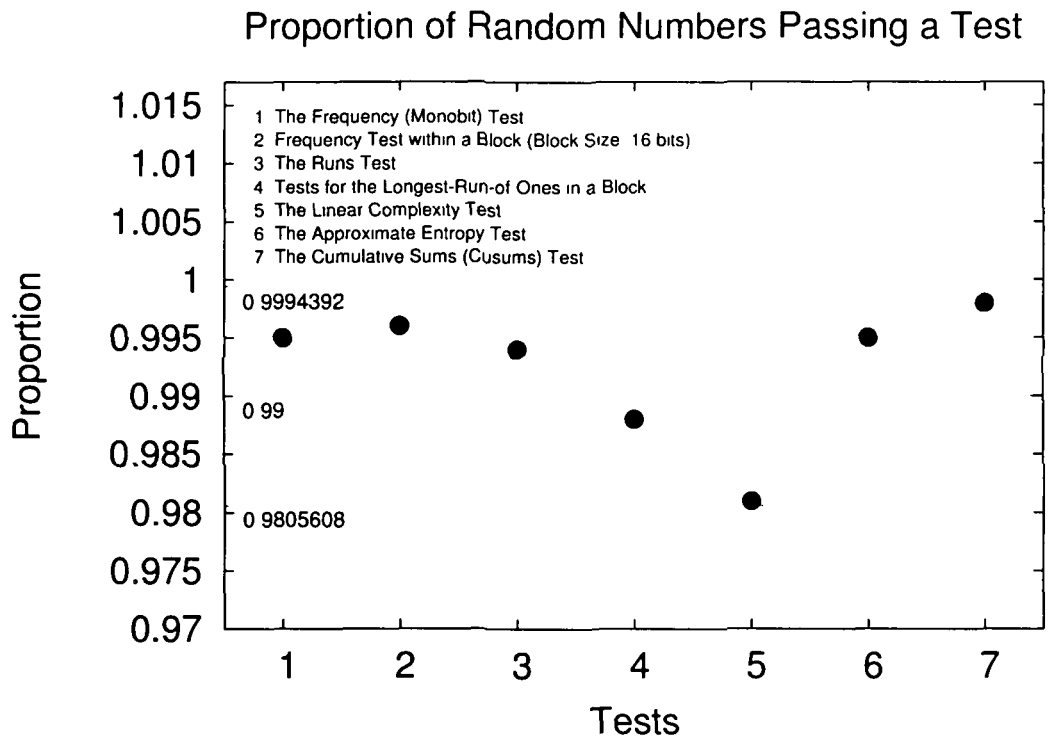

Figure 3.10: P-value plot

sub-interval are counted. Uniformity is then determined via an application of a χ^2 test and the determination of a *P-value* corresponding to the Goodness-of-Fit Distributional Test on these *P-values* (i e., a *P-value* of the *P-values*). This is accomplished by computing.

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - \frac{t}{10})^2}{\frac{t}{10}} \qquad (3.7\ 16)$$

where F_i is the number of *P-values* in sub-interval i, and t is the sample size. A *P-value* is calculated such that:

$$P\text{-}value_T = igamc\ (\frac{9}{2}, \frac{\chi^2}{2}) \qquad (3.7\ 17)$$

where igamc is the complementary incomplete gamma function If $p\text{-}value_T \geq 0.0001$, then the sequence of random numbers can be considered to be uniformly distributed. The value of χ^2 and $P\text{-}value_T$ (calculated from the empirical results) for each of the test performed on 'ERAND txt' is summarised in Table 3 4. Since the value of $P\text{-}value_T$ for all the tests

Table 3.4: $P\text{-}value_T$ for the statistical tests.

Statistical Test	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	χ^2	$P\text{-}value_T$
Frequency Test	134	93	86	105	107	100	96	103	100	76	20.76	0.013760
Frequency Test within a Block	93	114	129	93	89	98	110	101	93	80	18.10	0.034010
Runs Test	105	106	125	110	92	94	64	121	105	78	31.32	0.000261
Test for Longest Run of Ones	97	112	88	90	100	120	78	131	86	98	24.42	0.003684
Linear Complexity Test	107	84	85	93	77	91	109	105	130	119	25.56	0.002410
Approximate Entropy Test	113	103	79	103	100	123	86	106	86	101	15.86	0.069863
Cumulative Sums Test	80	102	94	89	138	88	81	98	128	102	33.02	0.000132

are greater than 0.0001, we consider the sequence of random numbers stored in 'ERAND.txt' to be uniformly distributed.

Since the sequence of random numbers stored in 'ERAND.txt' passes both the approaches specified by NIST for interpretation of empirical results, we can accept the null hypothesis and may conclude that the example algorithm proposed for generation of $ERANDs$ produces a sequence of $ERANDs$ that are sufficiently random for cryptographic use.

3.8 Summary

In this chapter, the AKA protocol used for access security in UMTS (i.e., UMTS-AKA) is analysed. It is found that due to the existing trust model used for roaming in UMTS, there are vulnerabilities in UMTS-AKA because of which the identity privacy of the subscriber gets compromised. To improve this situation, a security extension called E2EUIC with respect to UMTS-AKA is proposed. The extension is based on our trust model that is proposed in Chapter 2. This is a novel solution that prevents transmission of $IMSI$ across the entire network (wired as well as wireless); without necessitating any change in the intermediate network. Unlike many solutions which look at restricting $IMSI$ transmission only over radio link, E2EUIC takes a comprehensive end-to-end view of the problem. A couple of example algorithms needed for implementation of E2EUIC, which otherwise are operator specific, were also proposed. The usability of a series of randomly generated keys in these example algorithms were then analysed using programs written in Java. In addition, the randomness of random numbers that are generated by the example algorithms were verified using a statistical test suite provided by NIST.

Chapter 4

Relaxing Trust Requirement in LTE

Long Term Evolution (LTE), which has evolved from UMTS, is standardised by 3GPP for inclusion into the fourth generation of mobile networks. Although security of LTE has evolved from the security of UMTS, due to different architectural and business requirements of fourth generation systems, LTE security is substantially different and improved compared to its predecessor.

In this chapter, we analyse Evolved Packet System Authentication and Key Agreement (EPS-AKA) protocol, which is the AKA protocol used for access security in LTE. We found that the possibility of user identity privacy compromise exists in EPS-AKA as well. Therefore, we adopt our proposed security extension, i.e., E2EUIC to EPS-AKA for improved subscriber identity privacy in LTE.

4.1 Introduction

As a part of their LTE/SAE initiative for the evolution of GSM, EDGE and UMTS architecture, 3GPP has standardised a purely IP based system called the EPS. E-UTRAN is the access part of the EPS [13]. It connects with the EPC, which is the core network of the EPS. As the EPC has a flat IP based architecture, LTE uses IP protocol for both real time and data communication services. It is based on OFDMA to offer high order data rates and data volumes.

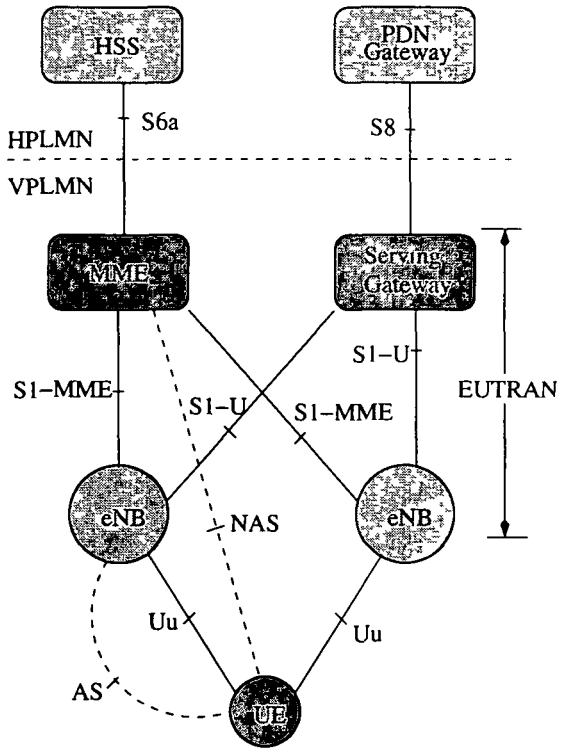

Figure 4.1 Simplified security architecture of LTE

LTE is among the most recent mobile system proposed by 3GPP that substantially enhances security as compared to UMTS [44] Although the security architecture of LTE has evolved from the security architecture of UMTS, LTE security architecture is substantially different from its predecessor Unlike UMTS, an elaborate set of keys are generated to provide explicit integrity protection and encryption security at the various interfaces in LTE

4.2 Security Architecture of LTE

Figure 4.1, depicts a simplified view of the roaming security architecture of an E-UTRAN [45]. In this, we show only the key elements associated with the EPS-AKA protocol. Each and every user is registered with a Home public Land Mobile Network (HPLMN), with their subscription and profile information stored in a Home Subscriber Server (HSS). In the Visiting Public Land Mobile Network (VPLM) the User Equipment (UE) connects with an evolved NodeB (eNB) through the Uu interface, for attach, Tracking Area Update (TAU)

and service requests. eNB is the new enhanced Base Transceiver Station (BTS) that provides the LTE air interface and performs radio resource management for the evolved access system. An eNB is connected with one or more Mobility Management Entities (MME) through the S1-MME interface. The MME is the key control node for the LTE access network and is responsible for authenticating the user by interacting with the HSS. For obtaining authentication data, the MME communicates with the HSS through the S6a interface. There are two layers of security between the UE and the E-UTRAN. The first layer is called the Access Stratum (AS) which protects the Radio Resource Control (RRC) plane signalling and the User Plane (UP) data between the UE and the eNB. The second layer is called the Non Access Stratum (NAS) which protects the control plane signalling between the UE and the MME. UE has access to packet data through the Packet Data Network Gateway (PDN-GW) via the Serving Gateway (S-GW).

4.3 EPS-AKA

EPS-AKA is the AKA protocol adopted by LTE for access security [20]. During the initial EPS-AKA (i.e., when the subscriber switches on the UE for the first time), the UE has to transmit its $IMSI$, since the UE does not have a temporary identity at this moment. For subsequent EPS-AKAs, the UE can identify itself by transmitting temporary identities.

4.3.1 The Initial EPS-AKA

The EPS-AKA procedure during the initial authentication is as follows:

1. The MME invokes the procedure by requesting an Authentication Vector (AV) from the HSS. The request shall include the $IMSI$ and the MME identity. It is recommended that the MME fetch only one authentication vector at a time as the need to perform AKA runs has been reduced in EPS through the use of a more elaborate key hierarchy, details of which is explained later.

2. Upon receipt of the request, the HSS assembles an $UMTS\text{-}AV$ (Chapter 3, Equation 3.3.7). An $UMTS\text{-}AV$ contains a random part $RAND$, an authenticaton token $AUTN$ used for authenticating the network to the UE, an expected response $XRES$, a 128-bit Integrity Key IK, and a 128-bit Cipher Key CK.

$$UMTS\text{-}AV = (RAND, AUTN, XRES, CK, IK) \tag{4.3.1}$$

The $AUTN$ contains a sequence number SQN used to indicate freshness of the AV. An $EPS\text{-}AV$ is then derived from the $UMTS\text{-}AV$ by replacing CK and IK with a Key for Access Security Management Entity (K_{ASME}). To derive K_{ASME}, a Key Derivation Function (KDF) [20] is used that take the following input parameters: CK, IK and MME identity. Thus,

$$K_{ASME} = KDF(CK, IK, MME\text{-}identity) \tag{4.3.2}$$

$$EPS\text{-}AV = (RAND, AUTN, XRES, K_{ASME}) \tag{4.3.3}$$

The HSS then sends $EPS\text{-}AV$ back to the MME.

3. After receiving $EPS\text{-}AV$, the MME extracts $RAND$ and $AUTN$ from it and sends them to the UE as a challenge. A Key Set Identifier (KSI_{ASME}) is also send along with the challenge. The purpose of the KSI_{ASME} is to make it possible for the UE and the MME to identify a native K_{ASME} without invoking the authentication procedure. This is used to allow re-use of the K_{ASME} during subsequent connection set-ups.

4. At receipt of this message, the UE uses the same procedure that is used in UMTS-AKA algorithm (Chapter 3, Section 3.6) to verify the $AUTN$. If $AUTN$ is incorrect, the UE rejects the authentication. If $AUTN$ is correct, the UE computes RES, IK and CK (using the same procedure used in UMTS-AKA algorithm). It then derives the K_{ASME} from the newly computed IK and CK. The UE then responds back to the MME with a user authentication response message that includes the computed RES.

5. Finally, MME checks whether RES is equal to $XRES$. If so, the authentication is successful. If not, the MME sends an authentication reject message to the UE.

Table 4.1: Keys derived from K_{ASME}.

Key	Details
K_{eNB}	Derived by UE and MME from K_{ASME}. After deriving, MME forwards it to the eNB.
K_{NASint}	Derived by UE and MME from K_{ASME} for integrity protection of NAS traffic.
K_{NASenc}	Derived by UE and MME from K_{ASME} for encryption of NAS traffic.
K_{UPenc}	Derived by UE and eNB from K_{eNB} for encryption of UP traffic.
K_{RRCint}	Derived by UE and eNB from K_{eNB} for integrity protection of RRC traffic.
K_{RRCenc}	Derived by UE and eNB from K_{eNB} for encryption of RRC traffic.

At the end of a successful EPS-AKA, a K_{ASME} is shared between the UE and the MME. A set of keys as explained in Table 4.1 are then generated from K_{ASME} for protection of the AS and the NAS [20]. The hierarchy of these keys is shown in Figure 4.2. These derived keys contribute to the following security contexts:

- EPS-NAS security context: This context consists of K_{ASME} and the associated KSI_{ASME}.
- EPS-AS security context: This context consists of K_{eNB}, K_{UPint}, K_{UPenc}, K_{RRCint} and K_{RRCenc}.
- EPS security context: It consists of the EPS-NAS and the EPS-AS security context.

In order to activate the EPS-NAS security context, the MME sends a NAS Security Mode Command (SMC) to the UE, and the UE replies with a NAS security mode complete message. The NAS-SMC contains the KSI_{ASME}. Similarly, to activate the EPS-AS security context, an AS-SMC is send by the eNB to

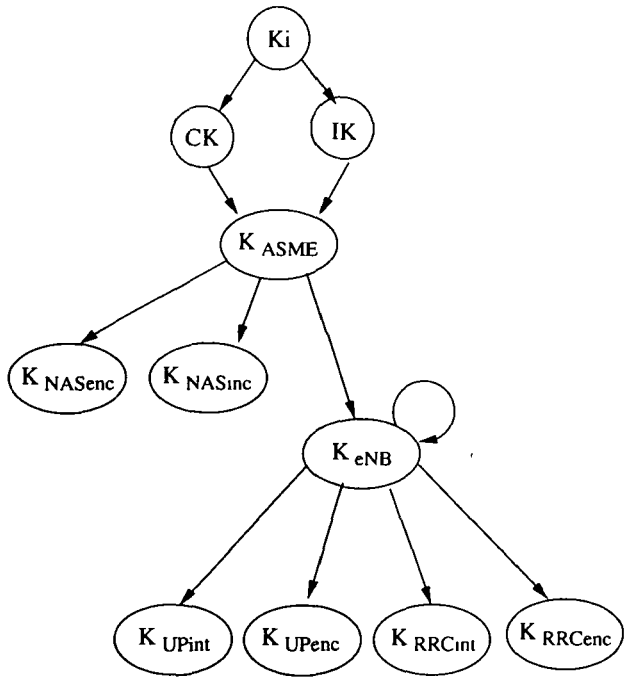

Figure 4.2: Key hierarchy.

the UE. The MME then allocates a fresh temporary identifier called Globally Unique Temporary Identity ($GUTI$) [5] to the UE by initiating a $GUTI$ reallocation procedure through the NAS. During the $GUTI$ reallocation procedure, the MME sends $GUTI$ Reallocation Command to the UE and the UE returns $GUTI$ Reallocation Complete message to the MME. A new $GUTI$ shall be sent to the UE only after a successful activation of NAS security. A mapping between the $GUTI$ and the $IMSI$ of the UE is maintained at the MME. The purpose of the $GUTI$ is to provide an unambiguous identification of the UE, so that the subscriber's permanent identity (i.e., the $IMSI$) is not revealed.

4.3.2 Subsequent EPS-AKA

For subsequent authentications (during attach requests, Tracking Area Updates (TAU) and service requests), identity presentation of the UE is accomplished by transmitting a $GUTI$ through the radio path. The KSI_{ASME} is also sent along with the request. Before transmitting, the UE integrity protects the request using NAS security. Upon receipt of the connection request, the MME identifies the corresponding K_{ASME} with the help of the received $GUTI$ and the KSI_{ASME}.

The MME then checks the integrity protection of the message. If the integrity check succeeds, the MME, depending on the MME policy, may either decide to reuse K_{ASME} that was established during a previous AKA (without invoking a fresh authentication procedure) or may decide to go for a fresh EPS-AKA that will result in the establishment of a new K_{ASME}. In order to carry out a fresh EPS-AKA, the MME locates the $IMSI$ of the UE in its local database through the $IMSI$-$GUTI$ mapping and continues in the same manner as the initial connection (discussed above). In order to reuse a K_{ASME} a fresh set of keying material is derived form the K_{ASME}. Thus, the need to perform frequent AKA runs has been reduced in EPS through the use of a more elaborate key hierarchy. In particular, connection requests can be authenticated using a stored K_{ASME} without the need to perform a fresh AKA. Several successive connections may be secured through re-derived security contexts from the current K_{ASME}.

4.4 Identity Privacy in LTE

In order to protect identity privacy of the subscriber during identity presentation, a $GUTI$ is transmitted instead of the $IMSI$. The purpose of the $GUTI$ is to provide an unambiguous identification of the UE that does not reveal the subscriber's permanent identity (i.e., the $IMSI$). In spite of this security arrangement, there are occasions when the $IMSI$ may be transmitted in clear text.

4.5 Motivation

Some of the identity privacy related vulnerabilities in EPS-AKA that motivates our work are as follows:

- During the very first attach procedure the $IMSI$ has to be transmitted in clear text (TS 33.401 [20] section 5.1.1), since no $GUTI$ is available for identity presentation at this stage.
- The $IMSI$ has to be transmitted in clear text through the radio link as

and when the MME requests for it. The MME has provision to make such a request when it cannot map the received $GUTI$ with the corresponding $IMSI$. For instance, whenever the UE visits a new MME and the new MME cannot acquire the $IMSI$ of the UE from the old MME. Such a provision, which is practicable only due to the existing trust model adopted in LTE, provides an opening for a fake MME to compromise a subscriber's $IMSI$ [22].

- The MME located in the SN, whose trustworthiness we question, has full knowledge about the $IMSIs$ of all subscribers to whom it provides services.
- The responsibility of creation and allocation of temporary identities (i.e., $GUTI$) are assigned to the MME, whose trustworthiness can itself be questioned.
- It is clearly evident that the trust model adopted in LTE is the existing trust model discussed in Chapter 2, Section 2.2. Therefore, all the identity privacy related vulnerabilities and the roaming agreements related complexities associated with the existing trust model exists in LTE as well.

4.6 EPS-AKA-with-E2EUIC

E2EUIC is a security extension for improved identity privacy and relaxed trust requirement during roaming between mobile operators. In Chapter 3, we have proposed this extension with respect to UMTS-AKA (the AKA protocol used in UMTS). In this section, we demonstrate its compatibility with EPS-AKA (the AKA protocol used in LTE) [46]. The protocol flow of EPS-AKA when E2EUIC extension is adopted to it, is presented in the following subsections. For details regarding components like RIC, $RIC\text{-}Index$, $ERAND_{First}$, RIC_{First}, RIC_{Fresh}, RIC_{New}, RIC_{Prev}, RIC_{Old}, RIC_{InUse}, f_{Embed}, $f_{Extract}$, $f_{Encrypt}$, $f_{Decrypt}$, SEQ_{UE}, R, SEQ_{HN}, TTL_{DMSI}, $ERAND_{UE}$, $DMSI_{SN}$, etc., refer to Chapter 3, Section 3.6.

4.6.1 The Initial Authentication

The protocol flow of EPS-AKA-with-E2EUIC during the initial authentication is as follows:

(1.1) For the very first connection, the UE initiates an attach procedure by transmitting an attach request to the MME. Since the UE does not have a valid temporary identity at this stage, a $DMSI$, say $DMSI_1$, calculated using RIC_{First} is transmitted along with the request.

$$DMSI_1 = MCC||MNC||RIC_{First}||ERIC \tag{4.6.1}$$

where,

$$ERIC = f_{Encrypt_{Ki}}(RIC_{padded}) \tag{4.6.2}$$

and,

$$RIC_{padded} = RIC_{First}||SEQ_{UE}||R \tag{4.6.3}$$

(1.2) The MME temporarily stores $DMSI_1$ in its local database, as it would require this $DMSI$ to uniquely identify the UE for a period of time (till it receives the next $DMSI$ from the UE during a successful EPS-AKA), if this run of the EPS-AKA eventually succeeds. The MME identifies the HN of the UE by inspecting the MCC and MNC portion of $DMSI_1$. A request for a fresh AV is then sent along with $DMSI_1$ to the HSS.

(1.3) On receiving the request, the HSS separates RIC_{First} from $DMSI_1$. It then uses the *RIC-Index* to locate RIC_{First} and hence the $IMSI$ and the key Ki of the UE in the HSS's/HN's database (Chapter 3, Figure 3.6). After this, the $ERIC$ part of the $DMSI$ is decrypted using $f_{Decrypt}$ and the key Ki.

$$RIC_{Padded} = f_{Decrypt_{Ki}}(ERIC) \tag{4.6.4}$$

The RIC contained in RIC_{Padded} is then compared with the RIC part of $DMSI_1$ (i.e., RIC_{First}), success of this comparison ensures that a malicious agent did not create $DMSI_1$. The SEQ_{UE} part of RIC_{Padded} (Equation 4.6.3) is then compared with the value stored against SEQ_{HN} field in the

HSS's database. If $SEQ_{UE} > SEQ_{HN}$, the request is proven as fresh (not a replay of any previous request). Failure of any of these two comparisons, leads to rejection of the request. If the request for AV is found to be fresh and from a genuine source (from the above two comparisons), the following are performed:

(a) SEQ_{UE} is assigned to SEQ_{HN}.

$$SEQ_{HN} = SEQ_{UE} \tag{4.6.5}$$

(b) If RIC_{First} is stored in any of the RIC fields other than RIC_{InUse} in the HN's/HSS's database, RIC_{First} is moved from its current location (say L) to RIC_{InUse} and the RIC stored earlier against RIC_{InUse} is moved to L. In other words, L and RIC_{InUse} swaps their values. For example: if RIC_{First} is found in RIC_{New} then:

$$temp = RIC_{InUse} \tag{4.6.6}$$

$$RIC_{InUse} = RIC_{New} \tag{4.6.7}$$

$$RIC_{New} = temp \tag{4.6.8}$$

(c) A fresh EPS-AV (Equation 4.3.3) (say AV_1) is then generated by the HSS.

After this, HSS selects a fresh not-in-use RIC (RIC_{Fresh}) from the pool of $RICs$ ($RIC\text{-}Index$) using the same procedure presented in Chapter 3, Section 3.6. RIC_{Fresh} is then embedded into the $RAND$ part of $EPS\text{-}AV$ using the function f_{Embed}. We call the resultant number after embedding RIC_{Fresh} into a $RAND$ as an $ERAND$. Thus,

$$ERAND_1 = f_{Embed}(RIC_{Fresh}, RAND_1) \tag{4.6.9}$$

Therefore, the $EPS\text{-}AV$ will now have an $ERAND$ in it, instead of a $RAND$.

$$EPS\text{-}AV_1 = (ERAND_1, AUTN_1, XRES_1, K_{ASME_1}) \tag{4.6.10}$$

From now on, an $ERAND$ is used for all purposes where a $RAND$ is used in EPS-AKA (this will not have any impact on the protocol flow, as the size of $RAND$ and $ERAND$ are same (128 bit)). A copy of RIC_{Fresh} is also stored at the HN's database against the $IMSI$ of the subscriber. For this purpose, RIC_{Old} is replaced by RIC_{Prev}. RIC_{Prev} is replaced RIC_{New} and so on. And finally, RIC_{New} is replaced by RIC_{Fresh}. An entry in the *RIC-Index* against the *IMSI-Pointer* of RIC_{Fresh} is also made accordingly. Thus,

$$RIC_{Old} = RIC_{Prev} \tag{4.6.11}$$

$$RIC_{Prev} = RIC_{New} \tag{4.6.12}$$

$$RIC_{New} = RIC_{Fresh} \tag{4.6.13}$$

$$RIC_{Fresh}.IMSI\text{-}Pointer = address\_of(IMSI) \tag{4.6.14}$$

Finally, HSS sends AV_1 along with $DMSI_1$ back to the MME.

(1.4) On receipt, MME continues the AKA procedure by extracting the $ERAND$ and $AUTN$ part of AV_1 (i.e., $ERAND_1$ and $AUTN_1$). $ERAND_1$ and $AUTN_1$ are then transmitted as a challenge to the UE.

(1.5) The UE and the MME completes the remaining part of the EPS-AKA-with-E2EUIC extension procedure, following the same steps as in EPS-AKA. On successful completion of the mutual authentication process, the following additional steps are carried out by the UE and the MME:

(a) The UE saves the recent $ERAND_1$ that it received from the MME in a field (say $ERAND_{UE}$) in the USIM's flash memory.

(b) The MME stores $DMSI_1$ in a variable say $DMSI_{SN}$ in its local database. The value of this variable does not change till the MME does not receive a new $DMSI$ during a successful EPS-AKA with E2EUIC. The MME uses $DMSI_{SN} = DMSI_1$ to uniquely identify the UE as long as it does not receive a new $DMSI$. The same field that the MME uses to store the $IMSI$ can be used as $DMSI_{SN}$.

(1.6) At the end of the AKA procedure, a shared K_{ASME} is established between the UE and the MME . The MME then allocates a new $GUTI$ to the UE by initiating a $GUTI$ reallocation procedure through the NAS (as explained in Section 4.3). MME stores this $GUTI$ against $DMSI_{SN}$ in its local database. The $GUTI$ is also stored in the UE's memory (say in a field called $GUTI_{UE}$) for identity presentation during the next authentication. If the UE uses $GUTI_{UE}$ to identify itself in the next authentication, the $GUTI$-to-$DMSI_{SN}$ mapping helps the SN to locate/acquire an AV for that authentication.

4.6.2 Subsequent Authentications

During all subsequent communications and the corresponding mutual authentication involved therein, the UE may present its identity in two different ways. Both these ways are listed below in their order of preference:

By Transmitting a GUTI

In this case, the UE transmits the $GUTI$ stored in $GUTI_{UE}$. Out of the two options, this one is the preferred choice, since it provides the MME with an option to reduce authentication latency by reusing the already established K_{ASME} (as explained in Section 4.3.2) for the authentication. Depending on its policy, the MME may also choose to invoke a full EPS-AKA-with-E2EUIC by sending a fresh request for AV to the HSS. The protocol flow for subsequent connections through the transmission of $GUTI$ and when the MME decides to invoke a full EPS-AKA, is as follows:

(2.1) UE extracts $GUTI_{UE}$ from its memory and transmits it to the MME.

(2.2) Through this $GUTI$, MME identifies the corresponding $DMSI$ (ie. $DMSI_{SN}$ stored in its database). MME then sends a request for a fresh AV to the HSS along with $DMSI_{SN}$.

(2.3) After receiving the request, HSS extracts the RIC part (say RIC_r) of $DMSI_{SN}$. The *IMSI-Pointer* against RIC_r leads to the record (in the

HSS's database) that contain the $IMSI$ of the UE. The remaining portion of this step proceeds in the same manner as in Section 4.6.1, Step 1.3.

(2.4) The remaining part of the protocol flow is same as Section 4.6.1, Steps 1.4 through 1.6.

By Transmitting a DMSI

The EPS-AKA-with-E2EUIC protocol flow for subsequent authentications through the transmission of a $DMSI$ is same as that of the initial authentication (i.e., Section 4.6.1, Steps 1.1 through 1.6). The only difference is that the RIC contained in the most recently received $ERAND$ (i.e., $ERAND_{UE}$) is used in this case by the UE to create a $DMSI$, rather than RIC_{First}. This option is less preferred and the UE may be forced to opt for this option when the UE roams into the area of a new MME or when the MME cannot identify the UE with its current $GUTI$. This method of identity presentation is performed only during the following situations:

- *The MME cannot identify the UE with the presented $GUTI$:* This may happen if the $GUTI$-to-$DMSI_{SN}$ mapping is lost from the MME's database.
- *The subscriber moves form an old MME (say MME_o) to the service area of a new MME (say MME_n):* In EPS-AKA, the first identity presentation under the service area of MME_n happens through transmission of a $GUTI$ (say $GUTI_o$) allotted to the UE by MME_o and the Location Area Identity (LAI) of MME_o. Unlike EPS-AKA, in EPS-AKA-with-E2EUIC the first identity presentation under the service area of MME_n happens through transmission of a fresh $DMSI$. This, makes the following two messages of EPS-AKA redundant:

  1. transmission of $GUTI_o$ from MME_n to MME_o.
  2. transmission of $IMSI$ and $GUTI_o$ from MME_o to MME_n.

  The above two messages enables MME_n to learn the $IMSI$ of the subscriber from MME_o. MME_n uses the received $IMSI$ to collect an $EPS\text{-}AV$

from the HSS. In EPS-AKA-with-E2EUIC. MME_n can directly collect an *EPS-AV* from the HSS (without communicating MME_o) by presenting the received *DMSI* to the HSS, thereby improving communication latency.

- *Time To Live for DMSI (TTL_{DMSI}) has expired.*

4.6.3 Strengths

Some of the strong points of EPS-AKA-with-E2EUIC extension are as follows:

- End to end user identity privacy: Knowledge of *IMSI* is confined only to the UE and the HSS; it is never transmitted at any stage of the protocol flow and at any portion of the path between the UE and the HSS.
- Relaxed trust requirement: MME-to-MME as well as HSS-to-MME trust relationship requirement with respect to permanent identity is relaxed. Such trust relaxation simplifies roaming agreements between operators.
- Reduced number of message exchanges: If the extension is adopted, two protocol messages of EPS-AKA becomes redundant. This will improve communication latency during authentication and key agreement.
- Fast database access: The *RIC-Index* makes searching through the HN's database faster.
- Minimal impact on the MME: For the extension to be adopted, most of the modifications are performed at the USIM and at the HSS, very negligible amount of adjustment, of that of treating a received *DMSI* as an *IMSI*, is required at the MME. This, makes the extension easier to adopt for the operators, since an operator has to do the necessary modifications only at the HSS's database and at the USIMs of the subscribers.

4.7 Summary

In this chapter, the AKA protocol used for access security in LTE (i.e., EPS-AKA) is analysed. It is found that with the use of an elaborate set of keys to

provide explicit integrity protection and encryption security at the various interfaces, access security in LTE is more advanced than that of UMTS. However, the trust model used for roaming and the security arrangement for identity privacy in LTE remains the same as that of UMTS. The only difference being the use of a $GUTI$ in place of a $TMSI$ to randomise the $IMSI$. Therefore, the need to relax trust requirement for roaming and to improve the status of identity privacy continues to exist in LTE. Towards this, we have shown how E2EUIC can be adopted to EPS-AKA for relaxed trust requirement for roaming between operators with respect to identity privacy in LTE.

Chapter 5

Relaxing Trust Requirement in 3GPP Interworking Systems

Mobile telecommunication systems that are proposed by 3GPP have wide coverage, efficient subscriber management and expertise in billing. Whereas, non 3GPP defined access networks like WLAN, WiMAX, etc., have limited coverage, but with data transfer rates much higher and cost much cheaper than that of 3GPP defined telecommunication systems. With the intention to extend the benefit of both these types of networks and to provide ubiquitous services to the subscribers, 3GPP has proposed interworking standards like Interworking between 3GPP System and WLAN (3GPP-WLAN) and Interworking between Non-3GPP Accesses and the EPS (Non3GPP-EPS). Considering the complexities, innate vulnerabilities and untrustworthiness associated with such interworking architectures, several security measures are adopted in these standards. However, a factor that continues to limit interoperability in both these interworking standards, is the trust requirement with respect to the subscriber's identity privacy. Moreover, there are occasions, when the identity privacy of the subscriber is compromised.

In this chapter, we adopt the security extension proposed in Chapter 3, Section 3.6 (i.e., E2EUIC) to the AKA protocol used in interworking standards like 3GPP-WLAN and Non3GPP-EPS to achieve relaxed trust requirement for roaming between operators with respect to identity privacy.

5.1 Introduction

3GPP System is the telecommunication system that is standardised by 3GPP and that consists of a core network and a radio access network [47]. EDGE, UMTS, etc., are examples of 3GPP Systems. On the other hand, wireless technologies like IEEE 802.11 standards fall in the category of WLAN. The combination of 3GPP Systems and WLAN technologies offer the possibility of achieving any time, anywhere services, bringing benefits of both technologies to the end users and the service providers. Thus, with the intent to extend 3GPP services and functionality to the WLAN access environment, 3GPP has proposed the specification for 3GPP-WLAN [15].

3GPP has standardised the EPS, as a part of their LTE/SAE initiative for the evolution of GSM, EDGE and UMTS architecture. EPS supports multiple access technologies, through an all-IP based core network called the EPC. In order to expand the reach of 3GPP services beyond 3GPP defined Access Networks (3GPP-ANs) like GERAN, UTRAN, E-UTRAN, etc., 3GPP has proposed the TS for Non3GPP-EPS [16]. This specification specifies description for providing IP connectivity using Non-3GPP Access Networks (Non-3GPP ANs) like WiMAX, CDMA2000, WLAN, etc., to the EPC. Thus, the EPC has opened up possibilities for interworking with all kind of access networks (3GPP/Non-3GPP), thereby promising ubiquitous services to the subscribers, irrespective of the access network he/she is attached with. Non-3GPP access can be split into two categories, viz.: trusted Non-3GPP access and untrusted Non-3GPP access. 3GPP does not specify which Non-3GPP technologies should be considered trusted and which ones to be considered untrusted; this decision is made by the operator.

For interworking standards like 3GPP-WLAN and Non3GPP-EPS, elaborate service agreements are required to be carried out in advance in order to resolve billing issues, quality of service related issues, and most importantly, trust issues. A security feature that is adopted in the security architecture of both these interworking systems is an IPsec tunnel between the subscriber's UE and the core network [48]. The tunnel provides end to end confidentiality for signalling data

and user data communication between the UE and the core network. This relaxes the need for the subscriber and the core network to trust the intermediary access networks with signalling data and user data exchanged through it. Such features relaxes trust requirements, simplifies service agreements and therefore facilitates interoperability. However, a factor that continues to limit interoperability in both these standards is the trust requirement with respect to the subscriber's identity privacy.

The AKA protocol that provides access security in 3GPP-WLAN and Non3GPP-EPS, is based on the Extensible Authentication Protocol (EAP) method [49]; it is called EAP-AKA protocol in cases where the UE is inserted with an USIM and EAP-SIM protocol in cases where the UE is inserted with a SIM [50]. The scheme used to ensure identity privacy during execution of an EAP-AKA/EAP-SIM procedure, expects the UE to have unconditional trust on the intermediary access networks with respect to its permanent identity. As a result, there are situations during an EAP-AKA/EAP-SIM procedure, when the identity privacy of the subscriber gets compromised.

5.2 Security Architecture of 3GPP-WLAN

Figure 5.1 depicts a simplified roaming security architecture of 3GPP-WLAN [15]. This architecture is used for access to IP services provided by the Visitor Public Land Mobile Network (VPLMN). In this, we show only the key elements associated with the EAP-AKA protocol. The roaming security architecture used for access to IP services provided by the Home Public Land Mobile Network (HPLMN) is presented in Appendix-A (Figure A.1).

The UE may be a laptop computer, a smart phone or an ipad that a 3GPP subscriber carries with himself/herself. It is equipped with a WLAN card, a Universal Integrated Circuit Card (UICC) (which is a smart card that contains a SIM application in GSM network and an USIM application in UMTS network) and suitable software applications. All long term security credentials, including the permanent identity (i.e., the $IMSI$), used for subscriber and network

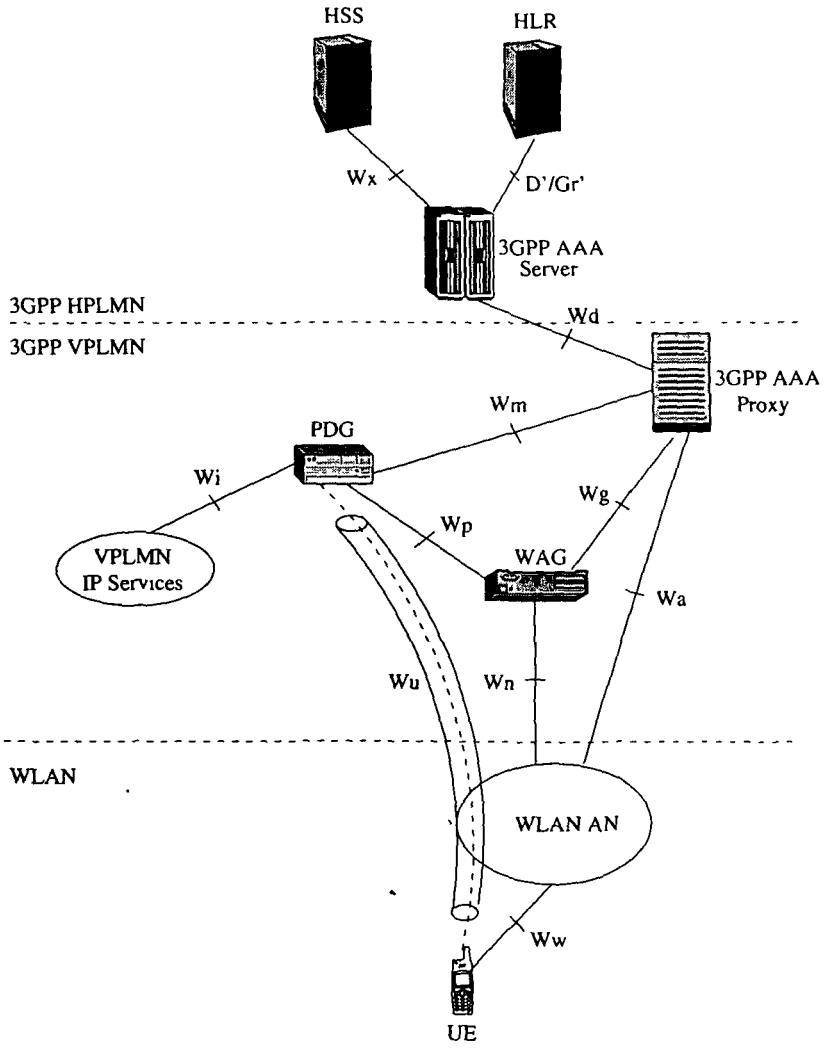

Figure 5.1: Simplified roaming security architecture of 3GPP-WLAN.

authentication shall be stored at the UICC. The 3GPP Authentication Authorisation Accounting Server (3GPP-AAA Server) is located at the HPLMN. Its primary responsibility is to authenticate the 3GPP subscriber based on authentication information retrieved from the Home Subscription Server (HSS) (through the Wx reference point) or the Home Location Register (HLR) (through the D'/Gr' reference point). The authentication signalling may pass through several AAA Proxies (through the Wd reference point). The AAA Proxies that may reside in any network between the WLAN and the 3GPP-AAA Server are used to relay AAA information between the WLAN and the 3GPP-AAA Server. The subscriber may opt for two types of IP accesses, viz.: WLAN 3GPP IP Access and WLAN Direct IP Access. The former is for access to an IP network via the

3GPP system, where as the later is for access to an IP network directly from the WLAN-AN. 3GPP IP based services are accessed via a Packet Data Gateway (PDG) located in the subscriber's HPLMN (in case of access to IP services provided by the HPLMN) or in the VPLMN (in case of IP services provided by the VPLMN). In order to protect user data packets transmitted between the UE and the PDG, a tunnel is established (through the Wu reference point). Using the information retrieved from the 3GPP-AAA Server, the PDG enforces tunnel authorisation and establishment. The WLAN Access Gateway (WAG) is a gateway via which data to/from the WLAN shall be routed to provide the UE with 3GPP IP based services. Ww is the reference point between the UE and the WLAN-AN. Wn is the reference point between the WLAN-AN and the WAG. And, Wp is the reference point between the WAG and the PDG.

5.3 Security Architecture of Non3GPP-EPS

Figure 5.2, depicts a simplified view of the roaming security architecture for Non-3GPP access to EPS. This architecture is used for access to packet switched services provided by the VPLMN. In this, we show only the key elements associated with the EAP-AKA protocol. The roaming security architecture used for access to packet switched services provided by the HPLMN is presented in Appendix-B (Figure B.1).

The 3GPP AAA Server is located at the HPLMN. Its primary responsibility is to authenticate the subscriber, based on authentication information retrieved from the HSS (through the SWx reference point). The authentication signalling may pass (through SWd reference points) via several AAA Proxies. The AAA Proxies may reside in any network between the Non-3GPP AN and the 3GPP AAA Server and are used to relay AAA information. The SWa reference point connects the untrusted Non-3GPP AN with the 3GPP AAA Server/Proxy, whereas the STa reference point connects the trusted Non-3GPP AN with the 3GPP AAA Server/Proxy. The Packet Data Network Gateway (PDN GW) provides the UE with connectivity to the external packet data networks (through

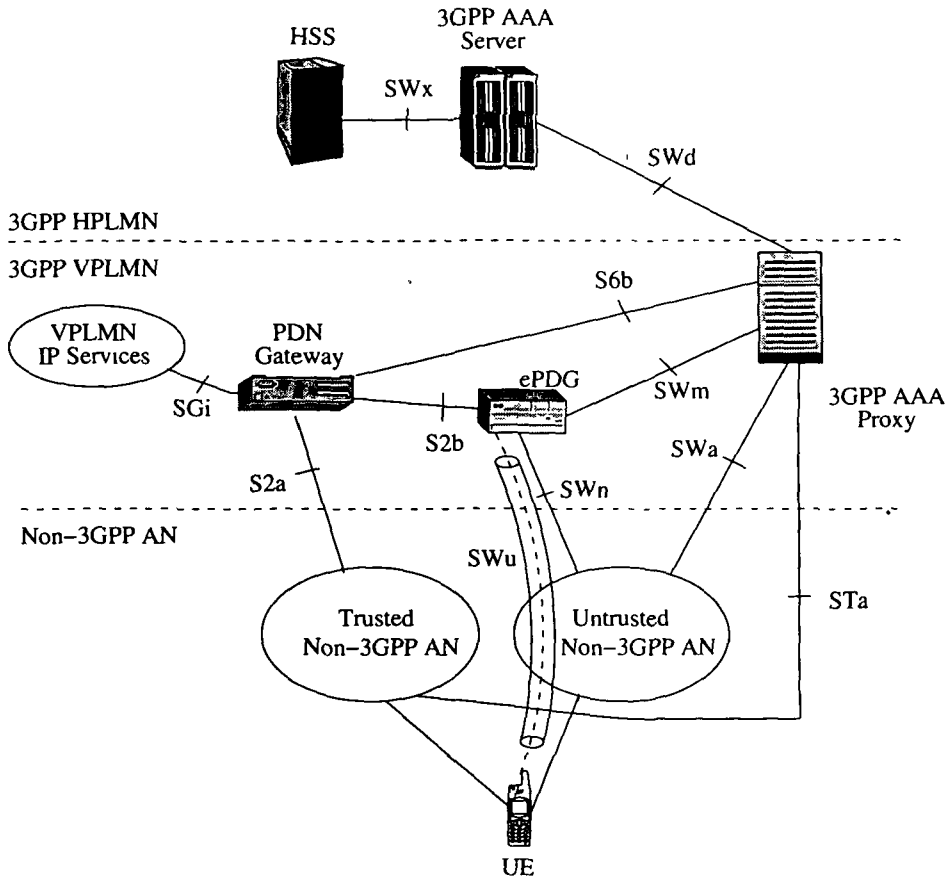

Figure 5.2: Simplified roaming security architecture of Non3GPP-EPS

the SGi reference point) by being the point of exit and entry of traffic for the UE. The Evolved Packet Data Gateway (ePDG) is a gateway with which an IPsec tunnel is established by the UE (through the SWu reference point) for untrusted Non-3GPP access to EPS. The S2a reference point provides user plane between the trusted Non-3GPP AN and the PDN GW. The S2b reference point provides the user plane between the ePDG and the PDN GW. Whereas, S6b is the reference point between the PDN GW and the 3GPP AAA Proxy. And, SWn is the reference point between the untrusted Non-3GPP AN and the ePDG.

5.4 EAP-AKA

The authentication and key agreement protocol adopted to provide access security in 3GPP-WLAN and Non3GPP-EPS Accesses, is EAP-AKA [48]. The EAP-AKA protocol that is executed between the UE and the 3GPP-AAA server

through an Access Network (AN), is described in the following paragraphs.

In order to begin the EAP-AKA procedure, the AN (i.e., the WLAN-AN or the Non-3GPP AN) sends an EAP Request/Identity message to the UE. In response, the UE sends an EAP Response/Identity message back to the AN that contains the identity of the UE in Network Access Identifier (NAI) format [5] The transmitted identity may either be a temporary identity allocated to the UE in the previous authentication or, in case of the first authentication, the $IMSI$. The message is then routed towards the proper 3GPP-AAA Server through one or more AAA proxies with the help of the realm part of the NAI.

An identity in NAI format, has the form username@realm as specified in clause 3 of RFC 2486 [51]. The NAI realm name shall be in the form of an Internet domain name as specified in RFC 1035 [52] and shall identify the user's HPLMN, based on its MCC and MNC. Details on NAI realm construction are specified in TS 23.003 [5]. The following steps show how to derive the NAI format from a given $IMSI$:

1. use the whole string of digits as the username part.
2. convert the leading digits of the $IMSI$, i.e., MNC and MCC, into a domain name.

The result will be an identity in NAI format of the following form:
<IMSI>@ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org
For example: If the $IMSI$ is 234150999999999 ($MCC = 234$, $MNC = 15$), the identity in NAI format takes the following form:
234150999999999@ims.mnc015.mcc234.3gppnetwork.org

In case the NAI received from the UE contains a temporary identity, the 3GPP AAA Server extracts the corresponding $IMSI$ form this temporary identity by using a procedure explained later in Section 5.4.1. The 3GPP-AAA Server then acquires authentication data, to be used for mutual authentication between the UE and itself, from the HSS by producing this $IMSI$. The authentication data comprises of an AV, which is based on the authentication vectors used in UMTS (Chapter 3, Section 3.3, Equation 3.3.7). It contains a random part

$RAND$, an authentication token $AUTN$ used for authenticating the network to the UE, an expected response part $XRES$, a 128-bit Integrity Key IK, and a 128-bit Cipher Key CK.

After an AV is acquired, the 3GPP-AAA Server derives new keying material, viz., Master Session Key (MSK) and Extended Master Session Key ($EMSK$), from IK and CK. Fresh temporary identities (fast re-authentication identity, pseudonym) may also be generated at this stage, using the mechanism explained in Section 5.4.1. The temporary identities are then encrypted and integrity protected with the keying material. The 3GPP-AAA server sends $RAND$, $AUTN$, a Message Authentication Code (MAC) (generated using the keying material) and the encrypted temporary identities to the AN in an EAP Request/AKA-Challenge message. $RAND$, $AUTN$, MAC and the encrypted identities are then forwarded to the UE by the AN.

The UE runs UMTS algorithm (Chapter 3, Section 3.3) on the USIM. The USIM verifies that $AUTN$ is correct and hereby authenticates the network. If $AUTN$ is incorrect, the UE rejects the authentication. If $AUTN$ is correct, the UE computes RES, IK and CK. It then derives the keying material MSK and $EMSK$ from the newly computed IK and CK, and checks the received MAC with this keying material. If encrypted temporary identities were received then the UE stores them for future authentications. The UE calculates a new MAC value that covers the EAP message. This MAC value is calculated with the newly derived keying material. The UE then sends EAP Response/AKA-Challenge containing the calculated RES and the newly calculated MAC value to the AN. The AN in turn forwards the EAP Response/AKA-Challenge packet to the 3GPP-AAA Server.

The 3GPP-AAA Server checks the received MAC and compares the received RES with $XRES$ (that was received earlier from the HSS as part of AV). If all checks are successful, the 3GPP-AAA Server sends an EAP success message to the AN through a trusted link. The keying material MSK is also included in this message for AN technology specific confidentiality and/or integrity protection; the AN stores this keying material to be used in communication with the

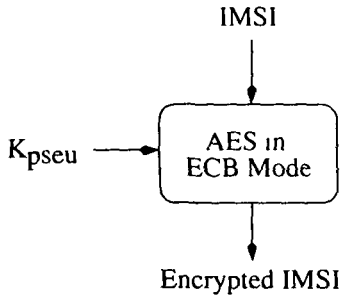

Figure 5.3: Temporary identity generation.

authenticated UE. The AN informs the UE about the successful authentication with an EAP success message. This completes the EAP-AKA procedure, at the end of which the UE and the authenticator in the AN share keying material derived during the exchange.

5.4.1 Temporary Identity Generation

In order to generate temporary identities, the 3GPP AAA Server encrypts the $IMSI$ using AES in Electronic Codebook (ECB) mode of operation (Figure 5.3). A 128-bit secret key K_{pseu} is used for the encryption. A single K_{pseu} is used for generation of temporary identities only for a given period of time determined by the operator. Once that time has expired, a new key is used. The 3GPP-AAA Server should keep a number of old keys (upto 16) for interpretation of the received temporary identities that were generated with those old keys. The number of old keys kept in the 3GPP-AAA Server is set by the operator, but it must be at least one, in order to avoid a just-generated temporary identity from becoming invalid immediately due to the expiration of the key. Each key must have a key indicator value associated with it. For identity presentation by a UE, this value is sent along with the temporary identity, so that when the 3GPP-AAA Server receives the temporary identity, it can use the corresponding key for obtaining the respective $IMSI$.

5.4.2 Security Mechanism Used in EAP-AKA for Identity Privacy

In order to ensure user identity privacy to the subscribers, the 3GPP-AAA Server generates and allocates temporary identities to the UE in a secured way (as discussed in section 5.4). For identity presentation, a temporary identity is transmitted by the UE instead of the permanent identity (i.e., *IMSI*). Two types of temporary identities, viz., re-authentication identities and pseudonyms are allocated to the UE. A re-authentication identity is used for identity presentation during a fast re-authentication [48] and a pseudonym is transmitted for identity presentation during an EAP-AKA. The UE does not interpret the temporary identities, it just stores them and uses them at the next authentication. Section 5.4.1 explains the mechanism used to generate the temporary identities.

5.5 Access Security in 3GPP-WLAN

In order to access packet switched services, a UE has to go through two different rounds of authentications with the 3GPP-AAA Server. In the first round, the user executes the EAP-AKA protocol that registers it to the WLAN-AN. In the second round, the user executes EAP-AKA within Internet Key Exchange version 2 (IKEv2) protocol [53] to establish a tunnel with the PDG, which registers it to the PLMN (3GPP core network). The tunnel is used for secured packet switched communication between the user and the core network.

5.5.1 Registration to WLAN-AN

At first, a connection is established between the UE and the WLAN-AN, using a WLAN technology specific procedure. After this, the EAP-AKA procedure (Section 5.4) is executed between the UE and the 3GPP-AAA Server, at the end of which the UE and the WLAN-AN share keying material derived during the exchange. This completes the EAP-AKA procedure required to register the UE with the WLAN-AN.

5.5.2 Tunnel Establishment

Post registration, the UE and the PDG exchange a pair of messages, to establish an IKEV2 channel, in which the PDG and UE negotiate cryptographic algorithms, exchange nonce and perform a Diffie Hellman exchange. In the remaining part of the authentication process, EAP-AKA (as explained in Section 5.4) is executed through this channel. During which, the UE sends its identity (compliant with NAI format, containing the $IMSI$ or a temporary identity) to the PDG via the IKEv2 secured channel which can only be decrypted and authenticated by the end points (i.e., the UE and the PDG). The PDG sends an authentication request message that contains the user identity to the 3GPP-AAA server. The 3GPP-AAA server fetches an AV from the HSS and initiates the authentication challenge by sending an EAP message that contains $RAND$, $AUTN$, MAC and protected identities to the PDG. The PDG in turn forwards the challenge along with its identity and a certificate to the UE. The UE checks the authentication parameters and responds to the authentication challenge. The PDG forwards this response to the 3GPP-AAA Server. When all checks are successful, the 3GPP-AAA Server sends an EAP success and a key material to the PDG. This key material shall consist of the MSK generated during the authentication process. The EAP success message is forwarded to the UE over IKEv2. This completes EAP-AKA exchange for tunnel establishment between the UE and the PDG, at the end of which, the UE and the PDG share keying material derived during that exchange.

5.6 Access Security in Non3GPP-EPS

Non-3GPP access can be split into two categories, viz.: trusted Non-3GPP access and untrusted Non-3GPP access. 3GPP does not specify which Non-3GPP technologies should be considered trusted and which ones to be considered untrusted. This decision is made by the operator.

5.6.1 Trusted Non-3GPP Access

In trusted Non-3GPP access, the UE connects with the EPC directly through the Non-3GPP AN. For access security, the UE and the 3GPP AAA Server executes the EAP-AKA protocol (Section 5.4) between them. At the end of a successful EAP-AKA, necessary key materials for secured data communication between the UE and the Non-3GPP AN is established.

5.6.2 Untrusted Non-3GPP Access

Unlike trusted Non-3GPP access, where the UE connects directly with the EPC; *in untrusted Non-3GPP access, the UE connects with the EPC via a network* entity called the Evolved Packet Data Gateway (ePDG) (SWu reference point).. The UE executes EAP-AKA using Internet Key Exchange version 2 (IKEv2) protocol [53] to establish an IPsec tunnel with the ePDG. The UE and the ePDG exchange a pair of messages to establish an IKEv2 channel in which the ePDG and UE negotiate cryptographic algorithms, exchange nonce and perform a Diffie Hellman exchange. In the remaining part of the authentication process, EAP-AKA (as explained in Section 5.4) is executed through this channel. The UE sends its identity (compliant with NAI format, containing the $IMSI$ or the temporary identity) to the ePDG via the IKEv2 secured channel which can only be decrypted and authenticated by the end points (i.e., the UE and the ePDG). The ePDG sends an authentication request message along with the user identity to the 3GPP-AAA Server. The 3GPP-AAA Server fetches the AVs from the HSS and initiates the authentication challenge by sending an EAP message that contains $RAND$, $AUTN$, MAC and encrypted temporary identities to the ePDG. The ePDG in turn forwards the challenge along with its identity and a certificate to the UE. The UE checks the authentication parameters and responds to the authentication challenge. The ePDG forwards this response to the 3GPP-AAA Server. When all checks are successful, the 3GPP-AAA Server sends an EAP success and a keying material to the ePDG. This keying material shall consist of the MSK generated during the authentication process. The EAP success message is forwarded to the UE over the secured IKEv2 channel. This

completes EAP-AKA exchange for tunnel establishment between the UE and the ePDG. After completion of the tunnel establishment and the authentication process, the UE and the ePDG share keying material that was derived during the process. The keying material is used for secured user data exchange through the tunnel during further communication between the UE and the ePDG.

5.7 Motivation

In-spite of the security measures, there are situations in EAP-AKA where the permanent identity of the subscriber is entrusted to intermediary network elements like WLAN-AN, Non3GPP-AN, PDG, ePDG, etc., and is transmitted in clear text through the radio channel, to the delight of an eavesdropper. Some of the identity privacy related vulnerabilities in EPS-AKA that motivates our work in this chapter are discussed in the following subsections.

5.7.1 Vulnerabilities During Registration to WLAN-AN and During Trusted Non-3GPP Access

Here, we discuss some of the identity privacy related vulnerabilities during registration to WLAN-AN (Section 5.5.1) in 3GPP-WLAN and during Trusted Non-3GPP Access (Section 5.6.1) in Non3GPP-EPS.

- The $IMSI$ has to be transmitted in clear text through the wireless link for identity presentation during the very first authentication.
- A subscriber, having a temporary identity (say t), may not initiate any new authentication attempt for a significant period of time. During this period, the key used to generate t could eventually get removed from the 3GPP-AAA Server (Section 5.4.1). If the user initiates an authentication attempt after this period, using t, the 3GPP-AAA Server will not be able to recognise it. In such a situation, the 3GPP-AAA server will request the UE for its other identities in the following order: 1. fast re-authentication identity, 2. pseudonym. If the 3GPP-AAA Server still does not recognise

any of these identities, it will request the UE to send its permanent identity. In response to such a request, the UE will have to transmit its $IMSI$ to the AN (for onward transmission to the 3GPP AAA Server) in clear text through the wireless link, making the $IMSI$ vulnerable to eavesdroppers.

- A corrupt AN belonging to a third party operator, may utilise the received $IMSI$ for various kind of malicious activities or may pass this identity to an unreliable party. To ensure that the subscriber's identity privacy is respected, the 3GPP operator will have to have elaborate trust agreements with the operator that owns the AN. However. such agreement complicates interoperability and limits roaming.

- A malicious/fake AN may also take advantage of the above situation by creating a spurious EAP Request/Identity message and by requesting the UE for its permanent identity; in response to which, the unsuspecting UE that does not have a mechanism to authenticate the request, will transmit its $IMSI$ in clear text through the wireless link.

5.7.2 Vulnerabilities During Tunnel Establishment

A tunnel is used for secured packet switched communication between the UE and the PDG in 3GPP-WLAN. Whereas, it is used for secured communication during untrusted Non-3GPP access between the UE and the ePDG in Non3GPP-EPS. During tunnel establishment, EAP-AKA message exchanges are performed through an IKEv2 protected channel that provides encryption and integrity protection. Thus, threats against identity privacy from passive attackers like eavesdroppers are significantly reduced. However, there exist the following risk when the $IMSI$ is send through the tunnel set-up procedure:

- The protected channel is encrypted but not authenticated at the time of receiving the user identity ($IMSI$). The IKEv2 messages, when using EAP, are authenticated at the end of the EAP exchange. So in case of a man-in-the middle attack, the attacker may pose as a genuine ePDG/PDG and

may request the UE for the $IMSI$. Although the attack would eventually fail at the time of the authentication, the attacker would have managed to see the $IMSI$ in clear text by then.

- The $IMSI$ would be visible to the ePDG/PDG, which in roaming situations will be located in the VPLMN. Such a vulnerability limits the HPLMN operator in interoperating with a VPLMN that belongs to an untrusted third party operator.

5.8 EAP-AKA-with-E2EUIC

In this section, we adopt E2EUIC to the EAP-AKA protocol, so that enhanced identity privacy and relaxed trust requirement for roaming can be achieved in interworking systems developed by 3GPP [54]. For details regarding components that are used in the extension, like RIC, $RIC\text{-}Index$, $ERAND_{First}$, RIC_{First}, RIC_{Fresh}, RIC_{New}, RIC_{Prev}, RIC_{Old}, f_{Embed}, $f_{Extract}$, $f_{Encrypt}$, $f_{Decrypt}$, SEQ_{UE}, R, SEQ_{HN}, TTL_{DMSI}, etc., Chapter 3, Section 3.6 may be referred.

The security mechanism adopted for identity privacy in EAP-AKA is significantly different from that of UMTS-AKA and EPS-AKA. In EAP-AKA, the responsibility of generation and distribution of temporary identities are shifted from the visited/serving network to the 3GPP AAA Server located in the HPLMN. This is done to have more home control, taking into consideration the innate vulnerability associated with interworking networks. As a result, the use of RIC_{InUse} field in the HSS's database, as required according to UMTS-AKA-with-E2EUIC and EPS-AKA-with-E2EUIC (Chapter 3, Figure 3.6), becomes redundant. Figure 5.4, depicts the HSS's database for EAP-AKA-with-E2EUIC.

For successful functioning of EAP-AKA-with-E2EUIC, a fresh RIC is allocated to the UE by the HSS during every run of EAP-AKA-with-E2EUIC. In order to allocate a fresh RIC, a RIC called RIC_{Fresh} is chosen randomly from the pool of not-in-use $RICs$ at the HSS (using the same procedure as in UMTS-AKA-with-E2EUIC (Chapter 3, Section 3.6)). RIC_{Fresh} is then cryptographically embedded into the $RAND$ part of the AV (equation 3.3.7) using

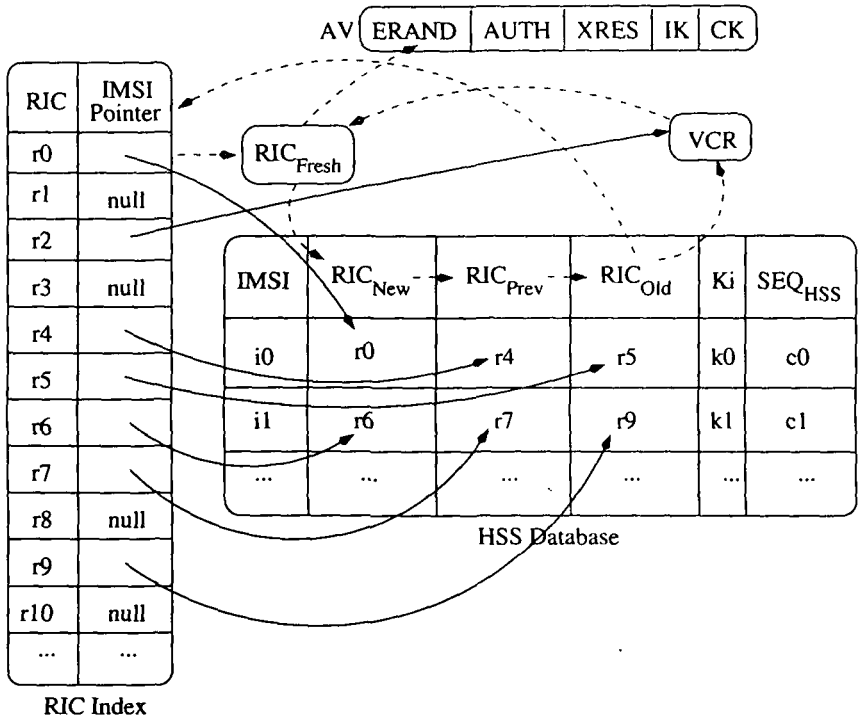

Figure 5.4: HSS's database for EAP-AKA-with-E2EUIC.

the function f_{Embed}. The resultant random number after embedding RIC into $RAND$ taking Ki as parameter is referred to as Embedded RAND ($ERAND$).

$$ERAND = f_{Embed_{Ki}}(RIC_{Fresh}, RAND) \tag{5.8.1}$$

The modified AV after embedding RIC into $RAND$ looks like the following:

$$AV = (ERAND, AUTH, XRES, IK, CK) \tag{5.8.2}$$

This $ERAND$ is now used by the 3GPP-AAA server, instead of the $RAND$ (as in the original EAP-AKA), to challenge the UE. Since the size of $RAND$ and $ERAND$ is same (i.e., 128 *bit*), the 3GPP-AAA server will not be able to perceive this change and will continue as before. The UE having knowledge of the long term shared key Ki can easily extract RIC_{Fresh} from $ERAND$.

$$RIC_{Fresh} = f_{Extract_{Ki}}(ERAND) \tag{5.8.3}$$

In EAP-AKA-with-E2EUIC, the $IMSI$ is never used for identity presentation. Instead, the UE presents a temporary identity or a $DMSI$. In either case, for successful functioning of the AKA protocol, the presented identity (i.e., the

temporary identity or the $DMSI$) has to be resolved to the corresponding $IMSI$ by the HSS. In the following subsections, we discuss the mechanism adopted at the HSS to resolve a presented identity to its corresponding $IMSI$.

5.8.1 Resolving a Temporary Identity to the Corresponding IMSI

In EAP-AKA-with-E2EUIC, the role of temporary identities (i.e., re-authentication identities and pseudonyms) remains exactly the same as EAP-AKA (as explained in Section 5.4.2). When the UE presents a temporary identity (in NAI format). The realm part of the temporary identity in NAI format helps the intermediate AAA proxy servers to guide the message to the appropriate 3GPP-AAA Server. When such a message reaches the 3GPP-AAA Server, the 3GPP-AAA Server resolves the temporary identity to its corresponding $IMSI$, using the procedure discussed in Section 5.4.1. A request for AV is then forwarded by the 3GPP-AAA Server to the HSS along with the resolved $IMSI$.

5.8.2 Resolving a DMSI to an IMSI

In order to prevent transmission of the $IMSI$ in EAP-AKA-with-E2EUIC, a $DMSI$ (in NAI format) is transmitted in place of the $IMSI$. A $DMSI$ is created using the RIC extracted from the most recent $ERAND$ that is received by the UE during a successful EAP-AKA.

$$DMSI = MCC||MNC||RIC||ERIC \tag{5.8.4}$$

where, $ERIC$ is created by encrypting a padded RIC (say RIC_{padded}) with the function $f_{Encrypt}$, taking the long term secret key Ki as parameter. Thus,

$$ERIC = f_{Encrypt_{Ki}}(RIC_{padded}) \tag{5.8.5}$$

where,

$$RIC_{padded} = RIC||SEQ_{UE}||R \tag{5.8.6}$$

The realm part of the $DMSI$ in NAI format helps the intermediate AAA proxy servers to guide it to the appropriate 3GPP-AAA Server. The 3GPP-AAA

Server forwards the $DMSI$ along with a request for AV to the HSS Thus, the onus of resolving the $DMSI$ is passed on to the HSS. On receipt of the request for AV, the HSS executes a sequence of instructions (Figure 5.5).

First and foremost, the HSS resolves the $DMSI$, which is done by locating the RIC part of the received $DMSI$ in the *RIC-Index* and by mapping it to the corresponding $IMSI$ through the *IMSI-Pointer*. The $ERIC$ part of the $DMSI$ is then decrypted using $f_{Decrypt}$ and the corresponding key Ki. Thus,

$$RIC_{padded} = f_{Decrypt_{Ki}}(ERIC) \tag{5.8.7}$$

The RIC contained in RIC_{padded} is compared with the RIC part of the $DMSI$, the success of this comparison ensures that a malicious agent did not create the $DMSI$. The SEQ_{UE} part of RIC_{padded} is then compared with the value stored against SEQ_{HN} field in the HSS's database. If $SEQ_{UE} > SEQ_{HN}$, the request is proven as a fresh request. Failure of any of these two comparisons, leads to rejection of the request. If the request for AV is found to be fresh and from a genuine source (from the above comparisons), SEQ_{UE} is copied into SEQ_{HN}

$$SEQ_{HN} = SEQ_{UE} \tag{5.8.8}$$

and a fresh AV is generated using the procedure used in EAP-AKA.

5.8.3 Embedding a RIC Into the RAND Part of AV

Whenever a RIC needs to be embedded into a $RAND$ at the HSS, a new RIC (RIC_{Fresh}) is selected from the pool of not-in-use $RICs$. In order to select RIC_{Fresh}, a b *bit* random number (say RN) is generated (using the same procedure as in UMTS-AKA-with-E2EUIC (Chapter 3, Section 3.6)).

$$RN = f_{PRNG}(seed) \tag{5.8.9}$$

This RN is then searched for in the *RIC-Index*. If the *IMSI-Pointer* against RN in the *RIC-Index* is found to be *null*, RN is selected as RIC_{Fresh} and the *null* value is replaced with the address of the record in the HSS's database where the $IMSI$ is stored.

$$RIC_{Fresh} = RN \tag{5.8.10}$$

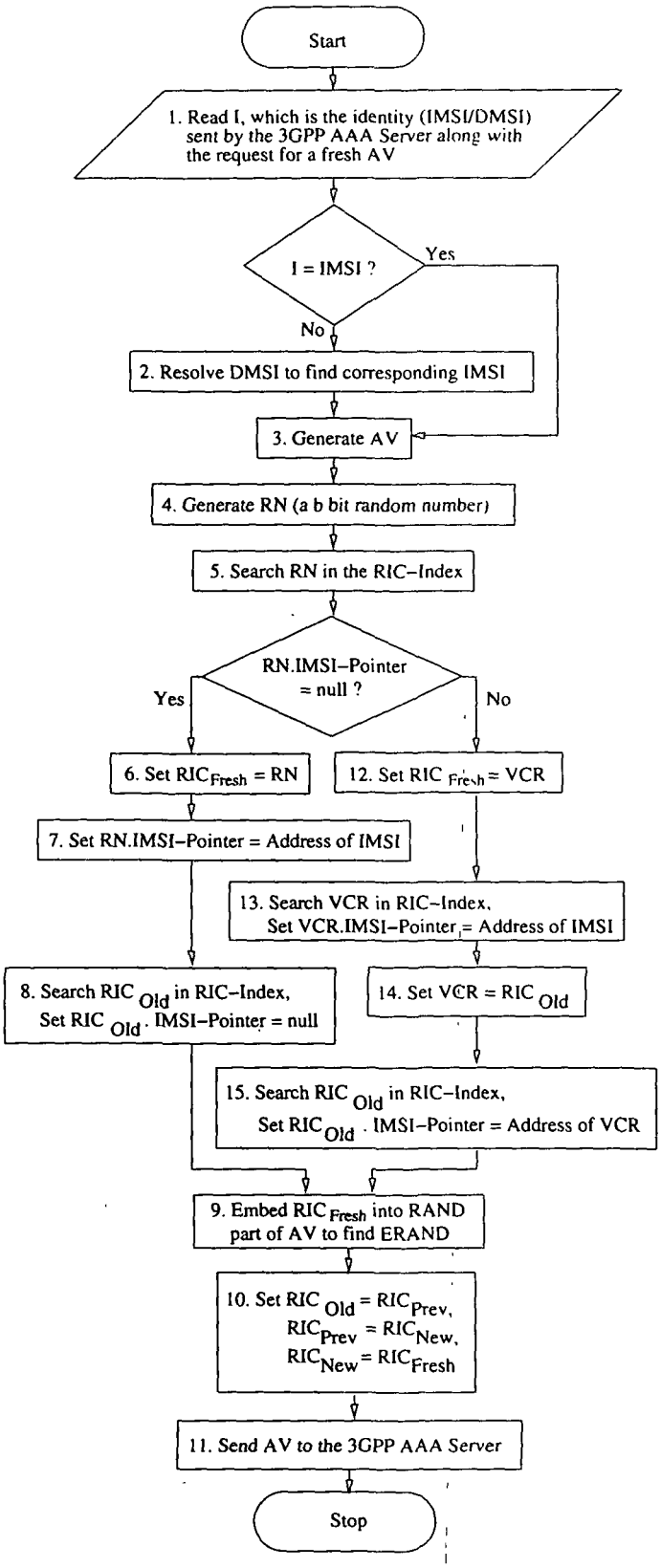

Figure 5.5: Flow of instructions at the HSS.

$$RN\ IMSI\text{-}Pointer = Address\ of\ IMSI \quad (5.8.11)$$

The oldest RIC value (i.e., RIC_{Old}) stored against the $IMSI$ is then returned to the pool of not-in-use RIC, by searching for it in the *RIC-Index* and by setting the *IMSI-Pointer* against it to *null*.

$$RIC_{Old}.IMSI\text{-}Pointer = null \quad (5.8.12)$$

In case the *IMSI-Pointer* against RN in the *RIC-Index* is not *null*, it may be inferred that there is a collision, and the value stored in the VCR is selected as RIC_{Fresh}. VCR is then searched for in the *RIC-Index* and the *IMSI-pointer* against it in the *RIC-Index* is made to point to the record in the HSS's database where the $IMSI$ is stored.

$$RIC_{Fresh} = VCR \quad (5.8.13)$$

$$VCR.IMSI\text{-}Pointer = Address\ of\ IMSI \quad (5.8.14)$$

In order to replace the RIC stored in the VCR with a fresh RIC, the oldest RIC (i.e., RIC_{Old}) stored against the $IMSI$ is copied into VCR. RIC_{Old} is then searched for in the *RIC-Index* and the *IMSI-pointer* against it is set to the address of VCR.

$$VCR = RIC_{Old} \quad (5.8.15)$$

$$RIC_{Old}.IMSI\text{-}Pointer = Address\ of\ VCR \quad (5.8.16)$$

RIC_{Fresh} is then embedded into the $RAND$ part of AV (using f_{Embed}). A copy of RIC_{Fresh} is also stored against the $IMSI$ in the HSS's database. To make space for RIC_{Fresh}, the value in RIC_{Prev} is copied into RIC_{Old} and the value in RIC_{New} is copied into RIC_{Prev}. Finally, the value in RIC_{Fresh} is copied into RIC_{New}.

$$RIC_{Old} = RIC_{Prev} \quad (5.8.17)$$

$$RIC_{Prev} = RIC_{New} \quad (5.8.18)$$

$$RIC_{New} = RIC_{Fresh} \quad (5.8.19)$$

The AV is then send to the 3GPP-AAA Server along with the $IMSI$. The 3GPP-AAA Server in turn, forwards a challenge containing $ERAND$ and $AUTH$

(extracted from AV) to the UE (through the Non-3GPP AN). The rest of the process continues in the same way as EAP-AKA. After successful authentication, the UE stores the $ERAND$ received as a challenge in its flash memory, to be used for identity presentation during subsequent authentications. The RIC embedded in $ERAND$ is extracted by the UE only when a $DMSI$ needs to be created.

5.8.4 Strengths

The strengths of EAP-AKA-with-E2EUIC may be summarised as follows:

- *End to end user identity privacy*: Knowledge of the $IMSI$ is confined only to the UE and the HPLMN; it is never transmitted at any stage of the network.

- *Relaxed trust requirement*: Since the $IMSI$ is never revealed to intermediary agents like AN, PDG, ePDG, etc., the HPLMN to AN trust relationship requirement with respect to the permanent identity (i.e. $IMSI$) is relaxed. This relaxation will facilitate interoperability.

- *No overhead at the intermediary network*: The proposed security extension has to be implemented only at the UE and the HSS, intermediary elements like like AN, PDG, ePDG, etc., can continue to maintain status quo. Thus, for an operator that adopts this extension, there is no additional cost of negotiation, implementation, computation, etc., to get the intermediary agents on board.

- *Simplifies a key security arrangement*: Since, with the proposed extension, the $IMSI$ is never transmitted at any stage of the EAP-AKA protocol, all the vulnerabilities listed in Section 5.7 are eliminated; thereby relaxing the need to trust an intermediary network element with the permanent identity of the subscriber. Thus, with respect to signalling data (that does not reveal the $IMSI$ any more), the extension removes the need to establish a tunnel between the UE and the core network. However, the need of the tunnel with respect to user data continues to exist.

- *The length of the DMSI in NAI format conforms to the maximum length expected to be handled correctly*: The maximum length of a NAI that is expected to be handled correctly by a standard equipment is 72 octets [55]. In addition, this NAI shall be transported inside the User-Name attribute of a RADIUS Access-Request, with standard length up to 63 octets [56]. Therefore, it can be assumed that the maximum length of a subscriber identity should be 63 octets (i.e., 63 characters) [50]. A *DMSI* in NAI format is of the following form:
  <DMSI>@ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org
  Since the *DMSI* is a concatenation of the *MCC* (2 octets), the *MNC* (2/3 octets), a *RIC* (4 octets) and an *ERIC* (16 octets), the maximum size of a *DMSI* in NAI format (considering each character of the relam as an octet) is 59 octets, which is less than the expected length of 63 octets.

5.9 Summary

In this chapter, the AKA protocol used for access security in Interworking networks like 3GPP-WLAN and Non3GPP-EPS (i.e., EAP-AKA) is analysed. The security mechanism adopted for identity privacy in EAP-AKA is significantly different from that of UMTS-AKA and EPS-AKA. In EAP-AKA, the responsibility of generation and distribution of temporary identities are shifted from the visited/serving network to the 3GPP AAA Server located in the HPLMN. This is done to have more home control, taking into consideration the inherent vulnerability associated with interworking networks. In addition, a security feature that is adopted in both these interworking systems is an IPsec tunnel between the subscriber's UE and the core network. The tunnel provides end to end confidentiality for signalling data and user data communication. This relaxes the need for the subscriber and the core network to trust the intermediary access networks with signalling data and user data exchanged through it. Such features relaxes trust requirements, simplifies service agreements and therefore facilitates interoperability. However, a factor that continues to limit interoperability in both these standards is the trust requirement on the intermediary elements like,

AN, PDG, ePDG, etc., with respect to the subscriber's identity privacy. To improve this situation, we have adopted E2EUIC extension to EAP-AKA. In EAP-AKA-with-E2EUIC, the need to trust the intermediary network elements with the $IMSI$ of the subscriber is eliminated. In addition, with respect to signalling data (that does not reveal the $IMSI$ any more), the extension removes the need to establish a tunnel between the UE and the core network.

Chapter 6

Performance Analysis

6.1 Introduction

In this thesis, we have proposed a security extension called E2EUIC for the AKA protocols used in mobile systems developed by 3GPP. E2EUIC improves user identity privacy and relaxes trust requirement for roaming. For the success of E2EUIC, it is important that it meets its security goals without adding much overhead (in terms of computation and space) to the existing network components. While doing so, it should also steer clear of introducing any additional vulnerability into the AKA procedure. A failure to do so would provide opportunities to adversaries, which in the first place this extension is trying to nullify. In order to validate the same, in this chapter, we analyse E2EUIC to examine its correctness, efficiency and robustness through various analysis like formal analysis, overhead analysis, complexity analysis and security analysis.

6.2 Formal Analysis

Authentication logics are one of the most widely used tools for analysing cryptographic protocols. They investigate how the beliefs of the agents/participants of a protocol evolve during exchange of messages that leads to statements like the protocol goal: "*A believes that K is a good key for communication with B*". If a protocol goal cannot be derived, it could be that some of the needed pre-

requisites are missing or it could also be that there is a failure in the protocol. Authentication logics does not explicitly find the failure but it gives the protocol designer or analyser a hint where the failure might be [57]. Thus, verifying a protocol with a formal method of authentication logic increases confidence in the protocol, even though it may not give 100% guarantee that there is no bug.

In this section, we perform a formal analysis of the proposed security extension (i.e., E2EUIC) using an enhanced BAN logic [58] called AUTLOG [59][60]. A similar analysis is performed by 3GPP in [61]. AUTLOG is based on a formal semantics and it is proven in [59] that the AUTLOG calculus is correct with respect to this formal semantics. A formal analysis using AUTLOG consists of four steps:

1. The necessary prerequisites are explicitly stated.
2. The messages are formally described.
3. The protocols goals are explicitly stated.
4. The formal calculus is applied to show how the protocols goals can be derived from the messages and the prerequisites using the rules of the calculus.

In the remaining part of this section, we carry out all the above four steps to prove that E2EUIC meets its security goals.

6.2.1 Prerequisites

The UE recognises Ki and believes that it is a good key for communication with the HN:

$$UE\ has\ Ki \tag{6.2.1}$$

$$UE\ recognises\ Ki \tag{6.2.2}$$

$$UE\ believes\ HN \overset{Ki}{\longleftrightarrow} UE \tag{6.2.3}$$

Since the UE is capable of verifying freshness of SEQ contained in the $AUTN$ part of the received challenge [62], it believes in SEQ's freshness.

$$UE\ believes\ fresh(SEQ) \tag{6.2.4}$$

The UE regards $ERAND$ as an atomic message

$$(ERAND)_{UE} \equiv ERAND \tag{6.2.5}$$

The challenge received by the UE contains a Message Authentication Code (MAC) in the $AUTN$ part of the challenge. MAC is an encryption of SEQ and $ERAND$ with key Ki. The UE believes that it has not said MAC itself.

$$UE\ believes\ \neg(UE\ said\ enc(Ki, SEQ, ERAND)) \tag{6.2.6}$$

The UE believes that the HN controls the freshness of RIC and that if the HN says $ERAND$ along with an $AUTN$ with a fresh SEQ in it, the RIC contained in the $ERAND$ is also fresh.

$$UE\ believes\ HN\ controls\ fresh(RIC) \tag{6.2.7}$$

$$UE\ believes\ (HN\ says\ (SEQ, ERAND) \longrightarrow HN\ believes\ fresh(RIC)) \tag{6.2.8}$$

$ERAND$ is an encrypted form of RIC. With knowledge of Ki, the UE can easily extract RIC from $ERAND$. Thus, UE is able to identify $ERAND$ with $enc(Ki, RIC)$.

$$(ERAND)_{UE} \equiv enc(Ki, RIC) \tag{6.2.9}$$

The UE believes that the HN has jurisdiction and belief concerning $IMSI$ as a shared secret between the UE and the HN.

$$UE\ believes\ HN\ controls\ HN \overset{IMSI}{\longleftrightarrow} UE \tag{6.2.10}$$

$$UE\ believes\ HN\ believes\ HN \overset{IMSI}{\longleftrightarrow} UE \tag{6.2.11}$$

The UE believes that the HN has jurisdiction on the fact that without access to the *RIC-Index*, RIC cannot be linked in any way with the corresponding $IMSI/MSIN$:

$$UE\ believes\ HN\ controls\ \neg(RIC \equiv IMSI) \tag{6.2.12}$$

$$UE\ believes\ (HN\ says\ ERAND \longrightarrow HN\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN)) \tag{6.2.13}$$

$$UE\ believes\ (HN\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN) \longrightarrow HN\ believes\ \neg(RIC \equiv MSIN)) \quad (6.2.14)$$

UE sees the following:

$$UE\ sees\ ERAND, \{SEQ\}_{enc(Ki,ERAND)}, enc(Ki, SEQ, ERAND) \quad (6.2.15)$$

6.2.2 Security Goals

$IMSI$ should be a shared secret between the UE and the HN (HLR/HSS). The same should not be disclosed by the UE to any third party including intermediary elements like SN, MME, WLAN AN, Non-3GPP AN, etc.

G1: $UE\ believes\ UE \overset{IMSI}{\longleftrightarrow} HN$

When ever temporary identities (i.e., TMSI, GUTI, re-authentication identities, pseudonyms, etc.) fails to protect the permanent identity, a backup mechanism is followed according to our proposed extension, so that identity privacy can still be ensured to the subscriber. According to this mechanism (E2EUIC), a $DMSI$ (created with the RIC that is extracted from the most recent $ERAND$ received by the UE) is transmitted in lieu of the $IMSI$. During every successful run of an AKA protocol with E2EUIC extension, if the UE receives a fresh RIC, it can easily protect its permanent identity.

G2: $UE\ believes\ UE\ has\ RIC$

G3: $UE\ believes\ fresh(RIC)$

It should not be possible for anyone except the HN (that has access to the RIC-Index) to map a $DMSI$ with its corresponding $IMSI$.

G4: $UE\ believes\ \neg(DMSI \equiv IMSI)$

6.2.3 Proving the Security Goals

Here, we derive the security goals from the prerequisites. The number in front of the implication arrow shows which equation lead to the implication, the symbol

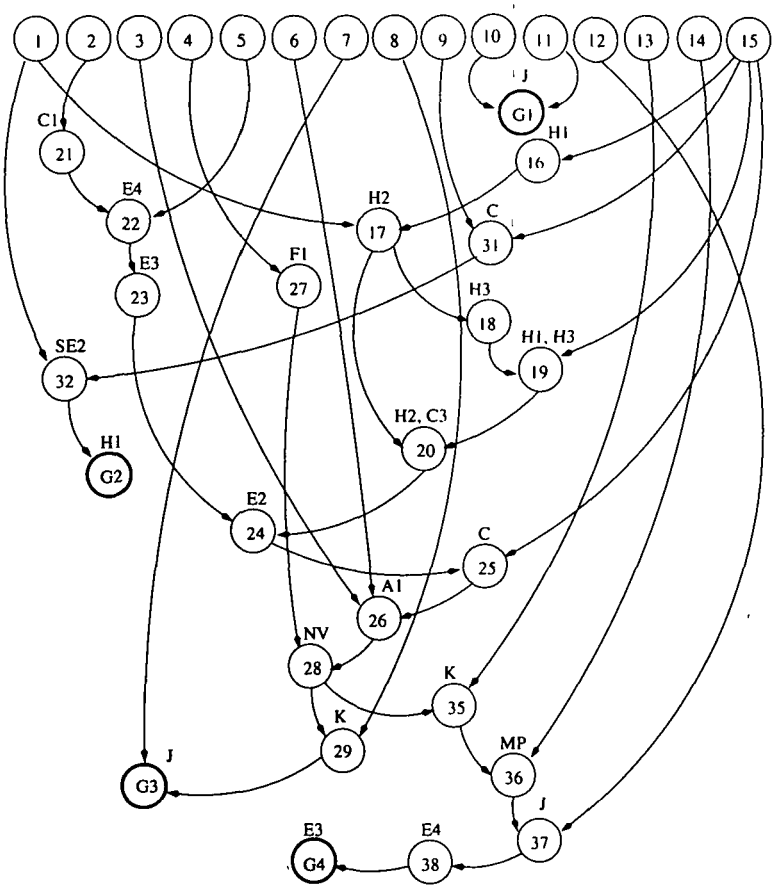

Figure 6.1: Deduction of security goals.

above the arrow refer to the rule of calculus (Appendix-C: Rules of AUTLOG Calculus) which has been used for this implication. Figure 6.1, depicts step by step deduction of the security goals. Each circle in the figure represents an equation in this section (circle 1 represents Equation 6.2.1, circle 2 represents Equation 6.2.2 and so on). The label against each circle represents the rule/rules of AUTLOG that is/are used to derive the particular implication.

$$6.2.15 \xrightarrow{H1} UE\ has\ ERAND \tag{6.2.16}$$

$$6.2.1, 6.2.16 \xrightarrow{H2} UE\ has\ (Ki, ERAND) \tag{6.2.17}$$

$$6.2.17 \xrightarrow{H3} UE\ has\ enc(Ki, ERAND) \tag{6.2.18}$$

$$6.2.15, 6.2.18 \xrightarrow{H1,H3} UE\ has\ SEQ \tag{6.2.19}$$

$$6.2.17, 6.2.19 \xrightarrow{H2,C3} (enc(Ki, SEQ, ERAND))_{UE}$$
$$\equiv enc((Ki, SEQ, ERAND)_{UE}) \tag{6.2.20}$$

$$6.2.2 \xrightarrow{C1} (Ki, SEQ, ERAND)_{UE} \equiv$$
$$((Ki)_{UE}, (SEQ)_{UE}, (ERAND)_{UE}) \quad (6.2.21)$$

$$6.2.5, 6.2.21 \xrightarrow{E4} (Ki, SEQ, ERAND)_{UE} \equiv (Ki, SEQ, ERAND) \quad (6.2.22)$$

$$6.2.22 \xrightarrow{E3} enc((Ki, SEQ, ERAND)_{UE}) \equiv$$
$$enc(Ki, SEQ, ERAND) \quad (6.2.23)$$

$$6.2.20, 6.2.23 \xrightarrow{E2} (enc(Ki, SEQ, ERAND))_{UE}$$
$$\equiv enc(Ki, SEQ, ERAND) \quad (6.2.24)$$

$$6.2.15, 6.2.24 \xrightarrow{C} UE \text{ believes } UE \text{ sees } enc(Ki, SEQ, ERAND) \quad (6.2.25)$$

$$6.2.25, 6.2.3, 6.2.6 \xrightarrow{A1} UE \text{ believes } HN \text{ said } (SEQ, ERAND) \quad (6.2.26)$$

$$6.2.4 \xrightarrow{F1} UE \text{ believes } fresh(SEQ, ERAND) \quad (6.2.27)$$

$$6.2.26, 6.2.27 \xrightarrow{NV} UE \text{ believes } HN \text{ says } (SEQ, ERAND) \quad (6.2.28)$$

$$6.2.28, 6.2.8 \xrightarrow{K} UE \text{ believes } HN \text{ believes } fresh(RIC) \quad (6.2.29)$$

$$6.2.7, 6.2.29 \xrightarrow{J} \boxed{UE \text{ believes } fresh(RIC)} \ (\mathbf{G3}) \quad (6.2.30)$$

$$6.2.15, 6.2.9 \xrightarrow{C} UE \text{ believes } UE \text{ sees } enc(Ki, RIC) \quad (6.2.31)$$

$$6.2.31, 6.2.1 \xrightarrow{SE2} UE \text{ believes } UE \text{ sees } RIC \quad (6.2.32)$$

$$6.2.32 \xrightarrow{H1} \boxed{UE \text{ believes } UE \text{ has } RIC} \ (\mathbf{G2}) \quad (6.2.33)$$

$$6.2.10, 6.2.11 \xrightarrow{J} \boxed{UE \text{ believes } (HN \xleftrightarrow{IMSI} UE)} \ (\mathbf{G1}) \quad (6.2.34)$$

$$6.2.28, 6.2.13 \xrightarrow{K} UE \text{ believes}$$
$$HN \text{ believes } \neg(f_x(Ki, ERAND) \equiv MSIN) \quad (6.2.35)$$

$$6.2.35, 6.2.14 \xrightarrow{MP} UE \text{ believes } HN \text{ believes } \neg(RIC \equiv MSIN) \quad (6.2.36)$$

$$6.2.12, 6.2.36 \xrightarrow{J} UE \text{ believes } \neg(RIC \equiv MSIN) \quad (6.2.37)$$

$$6.2.37 \xrightarrow{E4} UE \text{ believes}$$

$$\neg(MCC, MNC, RIC \equiv MCC, MNC, MSIN) \quad (6.2.38)$$

$$6.2.38 \xrightarrow{E3} \boxed{UE \text{ believes } \neg(DMSI \equiv IMSI)} \ (\mathbf{G4}) \quad (6.2.39)$$

Hence, it is proven that E2EUIC meets its security goals.

6.3 Computational Cost

In this section, we analyse the computational cost of E2EUIC, using a methodology proposed in [63]. The core idea of this methodology is to determine the amount of basic operations required for implementation of an algorithm, reducing all other operations in terms of these basic operations. For computational cost analysis of the proposed extension, all the other operations used in the extension are reduced to the following basic operations: byte-wise AND, byte-wise OR, shift (bytes) and logical comparison operation. For XOR operations, we exploit the following rule:

$$x \oplus y = x'y + y'x \quad (6.3.1)$$

Since negations are negligible compared to AND/OR logical operations, a bit-wise XOR is considered as the sum of two bit-wise ANDs and one bit-wise OR. The methodology can be used to calculate the computational overhead of some of the key computations involved in the extension as follows:

1. *Encryption/Decryption with AES*: Let $N_{EncryptAES}$ and $N_{DecryptAES}$ be the number of basic operations needed by AES to encrypt a 128 *bit* plain-text and to decrypt a 128 *bit* cipher-text, respectively, using a 128 *bit* key. Granelli et. al [63] found that 1720 byte-wise AND, 1268 byte-wise OR and 408 shift (byes) are involved in a 128/128 AES encryption, whereas, 5176 byte-wise AND, 3860 byte-wise OR and 1272 shift (byes) are involved in a 128/128 AES decryption. Thus,

$$N_{EncryptAES} = 3396 \quad (6.3.2)$$

$$N_{DecryptAES} = 10308 \tag{6.3.3}$$

2. *Encrypt RIC*: Let $N_{EncryptRIC}$ be the number of basic operations needed to encrypt a padded RIC to form an $ERIC$. 128/128 AES algorithm is used to carry out this encryption. Therefore,

$$N_{EncryptRIC} = N_{EncryptAES} = 3396 \tag{6.3.4}$$

3. *Decrypt ERIC*: Let $N_{DecryptERIC}$ be the number of basic operations needed to decrypt an $ERIC$ to find a padded RIC. 128/128 AES algorithm is used to carry out this decryption. Therefore,

$$N_{DecryptERIC} = N_{DecryptAES} = 10308 \tag{6.3.5}$$

4. *Search RIC*: Let $N_{SearchRIC}$ be the number of basic operations needed to search a RIC in the *RIC-Index*. The *RIC-Index* contains $n = 2^b$ number of entries arranged in sequential order, where b is size of the RIC in bits. A RIC can be searched using binary search in $log_2 n$ logical comparison operations. Thus, from Chapter 3, Equation 3.6.2,

$$\begin{aligned} N_{SearchRIC} &= log_2 n \\ &= log_2(m \times s) \end{aligned} \tag{6.3.6}$$

5. *Select* RIC_{Fresh}: Let $N_{SelectRIC_{Fresh}}$ be the number of basic operations needed to select a not-in-use RIC as RIC_{Fresh}. At first a b *bit* random number (RN) is generated, using a standard PRNG (in say N_{Rn} number of operations). For this purpose, the PRNG based on ANSI X9.31 Using AES can be used. In this PRNG, 256 bit-wise XOR operations (which amounts to 64 byte-wise AND and 32 byte-wise OR operations) and 3 rounds of the AES encryption algorithm are performed to generate a pseudo random number [39]. Thus,

$$\begin{aligned} N_{Rn} &= 64 + 32 + 3 \times N_{EncryptAES} \\ &= 10284 \end{aligned} \tag{6.3.7}$$

RN is then searched in *RIC-Index* in $N_{SearchRIC}$ number of operations. If the pointer against RN is *null* (with this comparison requiring 1 comparison operation), then RN is selected as RIC_{Fresh} by setting the *IMSI-Pointer* against it to the address of the concerned $IMSI$. Otherwise, the value in VCR is selected as RIC_{Fresh}. VCR is then searched in the *RIC-index* in $N_{SearchRIC}$ number of operations and the *IMSI-Pointer* against it is set to the address of the concerned $IMSI$. Thus,

$$\begin{aligned} N_{SelectRIC_{Fresh}} &= N_{Rn} + 2 \times N_{SearchRIC} + 1 \\ &= 10284 + 2 \times log_2(m \times s) + 1 \\ &= 10285 + 2 \times log_2(m \times s) \end{aligned} \tag{6.3.8}$$

6. *Embed RIC into RAND*: Let $N_{EmbedRIC}$ be the number of basic operations needed to embed a RIC into a $RAND$. Considering the example algorithm proposed in Chapter 3, Section 3.7, we found that a total of 32 bit-wise XOR operations (which amounts to 8 byte-wise AND and 4 byte-wise OR operations) and 1 round of the AES encryption algorithm are performed to embed a 32 *bit* RIC into a 128 *bit* $RAND$. Thus,

$$\begin{aligned} N_{EmbedRIC} &= 12 + N_{EncryptAES} \\ &= 3408. \end{aligned} \tag{6.3.9}$$

7. *Extract RIC from ERAND*: Let $N_{ExtractRIC}$ be the number of basic operations needed to extract the RIC that is embedded into an $ERAND$. Considering the example algorithm proposed in Chapter 3, Section 3.7, we found that a total of 32 bit-wise XOR operations (which amounts to 8 byte-wise AND and 4 byte-wise OR operations) and 1 round of the AES decryption algorithm are performed to extract the 32 *bit* RIC from a 128 *bit* $ERAND$. Thus,

$$\begin{aligned} N_{ExtractRIC} &= 8 + 4 + N_{DecryptAES} \\ &= 10320 \end{aligned} \tag{6.3.10}$$

8. *Return* RIC_{Old}: RIC_{Old} is searched in the *RIC-Index* in $N_{SearchRIC}$ operations. RIC_{Old} is then returned to the pool of not-in-use $RICs$ by setting

Table 6.1: Number of basic operations in the key computations of E2EUIC.

Key computations	Number of basic operations
1. $N_{EncryptAES}$	3396
2. $N_{DecryptAES}$	10308
3. $N_{EncryptRIC}$	3396
4. $N_{DecryptERIC}$	10308
5. $N_{SearchRIC}$	$log_2(m \times s)$
6. $N_{SelectRIC_{Fresh}}$	$10285 + 2 \times log_2(m \times s)$
7. $N_{EmbedRIC}$	3408
8. $N_{ExtractRIC}$	10320
9. $N_{ReturnRIC_{Old}}$	$log_2(m \times s)$

the *IMSI-Pointer* against RIC_{Old} in the *RIC-Index* to either *null* or to the address of VCR, depending on whether RN was selected as RIC_{Fresh} or VCR was selected as RIC_{Fresh}. Thus, if $N_{ReturnRIC_{Old}}$ is the total time taken for this purpose,

$$\begin{aligned} N_{ReturnRIC_{Old}} &= N_{SearchRIC} \\ &= log_2(m \times s) \end{aligned} \tag{6.3.11}$$

The number of basic operations required to execute the key computations involved in E2EUIC extension are summarised in Table 6.1.

6.3.1 Computational Cost at the UE

The proposed extension (i.e., E2EUIC) provides a backup mechanism that is used to identify the subscriber in situations where temporary identities fail to identify the subscriber. According to this mechanism, a $DMSI$ is transmitted in lieu of the permanent identity (i.e., the $IMSI$). The following computations are introduced at the UE, when the UE identifies itself with a $DMSI$:

1. Extract RIC from the most recently received $ERAND$ in $N_{ExtractRIC}$ number of operations.

2. Generate R, a $128 - (32 + b)$ bit random number. For this purpose, the PRNG based on ANSI X9.31 Using AES can be used. In this PRNG, 256 bit-wise XOR operations (which amounts to 64 byte-wise AND and 32 byte-wise OR operations) and 3 rounds of the AES encryption algorithm are performed to generate a pseudo random number [39]. Thus, the number of operations to generate R (say N_R) is as follows:

$$\begin{aligned} N_R &= 64 + 32 + 3 \times N_{EncryptAES} \\ &= 10284 \end{aligned} \tag{6.3.12}$$

3. Increment SEQ_{UE} in 1 operation.

4. Create $ERIC$ from the padded RIC in $N_{EncryptRIC}$ number of operations.

Thus, the computational overhead (say N_{UE}) introduced at the UE when a $DMSI$ is transmitted, can be calculated as follows:

$$\begin{aligned} N_{UE} &= N_{ExtractRIC} + N_R + 1 + N_{EncryptRIC} \\ &= 12 + N_{DecryptAES} + N_R + 1 + N_{EncryptAES} \\ &= 12 + 10308 + 10284 + 1 + 3396 \\ &= 24001 \end{aligned} \tag{6.3.13}$$

Therefore, the overall computational overhead introduced at the UE by E2EUIC extension during an AKA is as follows:

$$N_{UE} = \begin{cases} 0 & \text{-when a temporary identity is transmitted.} \\ 24001 & \text{-when a } DMSI \text{ is transmitted.} \end{cases} \tag{6.3.14}$$

6.3.2 Computational Cost at the HN

In an AKA with E2EUIC extension, whenever the HN receives a request for authentication data along with a $DMSI$, it has to firstly resolve the $DMSI$ into the corresponding $IMSI$. After this, it has to generate a fresh AV (or an array of authentication vectors (i.e., $AV[1..M]$, depending on operator's policy). It then embeds a fresh RIC (RIC_{Fresh}) into the $RAND$ part of the AV/AVs.

The following computations (executed in say $N_{Resolve}$ number of operations) are introduced at the HN when a $DMSI$ has to be resolved:

1. Search the *RIC-Index* for the RIC contained in the received $DMSI$ in $N_{SearchRIC}$ operations. This leads to the record in the HN's database (AuC/HLR) that contains the corresponding $IMSI$ and the Ki.

2. Decrypt the $ERIC$ part of the $DMSI$ to find RIC_{padded} in $N_{DecryptRIC}$ operations.

3. Compare the RIC contained in RIC_{padded} with the RIC part of the received $DMSI$ in 1 comparison operation.

4. Compare SEQ_{UE} part of RIC_{padded} with the SEQ_{HN} field maintained in the HN's database in 1 comparison operation

5. Set the RIC_{InUse} field maintained against the corresponding $IMSI$ to the RIC contained in the received $DMSI$ in a maximum of m comparison operations. As already discussed (Chapter 5, Section 5.8), this step is not required in EAP-AKA-with-E2EUIC. Thus,

$$\begin{aligned} N_{Resolve} &= N_{SearchRIC} + N_{DecryptRIC} + m + 1 + 1 \\ &= log_2(m \times s) + N_{DecryptAES} + 2 \\ &= log_2(m \times s) + m + 10310 \end{aligned} \tag{6.3.15}$$

The following computations are introduced at the HN, when RIC_{Fresh} has to be embedded into the $RAND$ part of a single AV.

1. Select RIC_{Fresh} in $N_{SelectRIC_{Fresh}}$ operations.

2. Embed RIC_{Fresh} into $RAND$ in $N_{EmbedRIC}$ operations.

3. Return the RIC_{Old} to the pool of not-in-used $RICs$ and store RIC_{Fresh} in the HN's database in $N_{ReturnRIC_{Old}}$ operations.

Thus, the following computations (executed in say N_{Embed} number of operations) are introduced when RIC_{Fresh} has to be embedded into the $RAND$ part of all the AVs of $AV[1..M]$:

$$\begin{aligned} N_{Embed} &= M \times (N_{SelectRIC_{Fresh}} + N_{EmbedRIC} + N_{ReturnRIC_{Old}}) \\ &= M \times (10285 + 2 \times log_2(m \times s) + 3408 + log_2(m \times s)) \\ &= M \times (13693 + 3 \times log_2(m \times s)) \end{aligned} \tag{6.3.16}$$

The total computational overhead (say N_{HN}) introduced at the HN by E2EUIC extension during an AKA is equal to $N_{Resolve} + N_{Embed}$. Thus,

$$\begin{aligned} N_{HN} &= N_{Resolve} + N_{Embed} \\ &= log_2(m \times s) + m + 10310 \\ &\quad + M \times (13693 + 3 \times log_2(m \times s)) \end{aligned} \tag{6.3.17}$$

If we consider the value of b to be 32 *bit* and the value of m to be 4 (as proposed in Chapter 3, Section 3.6), the value of $log_2(m \times s)$ can be derived using Chapter 3, Equation 3.6.2 and Chapter 3, Equation 3.6.1 as follows:

$$\begin{aligned} log_2(m \times s) &= log_2(n) \\ &= O(b) \\ &= 32 \end{aligned} \tag{6.3.18}$$

Thus,

$$N_{HN} = 24135 \tag{6.3.19}$$

where, we have considered M to be 1.

6.4 Time Complexity

In this section, we derive the time complexity (in terms of the growth in subscriber base s) that an operator can expect at the UE and the HN when the proposed security extension is adopted. From equation 6.3.14, we can infer that the following time complexity is introduced at the UE.

$$T_{UE} = O(1) \tag{6.4.1}$$

From Equation 6.3.17, the time complexity introduced at the HN can be derived as follows:

$$\begin{aligned} T_{HN} &= log_2(m \times s) + m + 10310 \\ &\quad + M \times (13693 + 3 \times log_2(m \times s)) \\ &= O(log_2 s), \quad m \text{ and } M \text{ being a constants.} \end{aligned} \tag{6.4.2}$$

where, s is the maximum number of subscribers that the extension is expected to handle.

6.5 Space Overhead

In order to adapt E2EUIC extension to an AKA protocol (UMTS-AKA, EPS-AKA and EAP-AKA), additional memory space is required at the UE and at the HN. In this section, we make an estimate of the amount of space required at the UE and the HN.

6.5.1 Space Overhead at the UE

In order to store the most recently received $ERAND$, the UE needs 128 *bit* of space in the UE's flash memory. And, in order to maintain a 32 *bit* $DMSI$ counter, an additional 32 *bit* are required. Thus, the amount of space (say S_{UE}) required by the extension at the UE is:

$$S_{UE} = 160 \ bit \tag{6.5.1}$$

6.5.2 Space Overhead at the HN

Against every $IMSI$ in the HN's database, $m = 4$ $RICs$ and a 32 *bit* SEQ_{HN} value are stored. Thus, every record in the HN's database will need an additional 160 *bit* space. In order to have provision for the maximum number of subscribers, i.e., s, the total amount of additional space (say S_{RIC}) needed at the HN's database is:

$$S_{RIC} = 160 \times s \ bit \tag{6.5.2}$$

In the *RIC-Index* there are n entries. Each of these entries has a 32 *bit* RIC and a 32 *bit IMSI-pointer*, with a total of 64 *bit*. Thus, the total amount of space (say $S_{RIC\text{-}Index}$) needed at the HN's memory to accommodate the *RIC-Index* is:

$$\begin{aligned} S_{RIC\text{-}Index} &= 64 \times m \times s \ bit \\ &= 256 \times s \ bit, \quad \text{considering } m \text{ as } 4. \end{aligned} \tag{6.5.3}$$

Thus, the total amount of space (say S_{Total}) required by the extension at the the HN is:

$$\begin{aligned} S_{Total} &= S_{RIC} + S_{RIC\text{-}Index} \\ &= 416 \times s \ bit \end{aligned} \tag{6.5.4}$$

where, s is the maximum number of subscribers that the extension is expected to handle.

Results of the analyses performed in Section 6.3 through Section 6.5 are summarised in Table 6.2.

6.6 Communication Overhead

E2EUIC is an extension for the existing AKA protocols used in 3GPP mobile systems. In order to adopt E2EUIC to the AKA protocols used in 3GPP mobile systems, no additional messages are needed. The same messages that are used in the existing AKA protocols are used to exchange information needed for successful functioning of E2EUIC. Thus, no communication overhead is introduced by E2EUIC to the existing AKA protocols. On the contrary, when E2EUIC is adopted to UMTS-AKA and EPS-AKA, it reduces two protocol messages, which otherwise are required when the current visited/serving network needs to acquire the $IMSI$ of a roaming subscriber from the previous visited/serving

Table 6.2: Computational cost, time complexity and space overhead of E2EUIC.

	At the UE	**At the HN**
Computational cost	No extra computational cost is introduced when a temporary identity is transmitted by the UE. However, 24001 basic operations are introduced when a DMSI is transmitted.	24135 basic operations are introduced when a DMSI is received and is to be resolved to its corresponding IMSI at the HN.
Time complexity	$O(1)$	$O(log_2 s)$
Space overhead	160 *bit*	$416 \times s$ *bit*

network (Chapter 3, Section 3.6; Chapter 4, Section 4.6).

6.7 Security Analysis

In this section, we discuss E2EUIC in terms of security threats that are perceived against it.

6.7.1 Replay Attack

A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated at a later instance, masquerading it as a regular transmission. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it [64].

In an AKA with E2EUIC, an adversary may try to launch the following two replay attacks:

1. A previously transmitted challenge (containing an $ERAND$ and an $AUTN$) is replayed again towards the UE.

2. A previously transmitted $DMSI$ is transmitted again for identity presentation to the HN (through the SN).

An attack of the first kind will fail because the UE will reject such a challenge after verifying freshness of the sequence number SQN contained in the $AUTN$ part of the challenge. The second type of attack will also not succeed because the HN will reject such a request after verifying its freshness by comparing the sequence number SEQ_{UE} contained in the presented $DMSI$ with the sequence number SEQ_{HN} maintained in its database.

6.7.2 Known Plain Text Attack

Known-plaintext attack (KPA) is an attack where the attacker has samples of both the plain text, and its encrypted version (i.e., the cipher text). The attacker uses these for cryptanalysis of the cipher text to reveal secret information such

as secret keys and code books [65]. In an AKA with E2EUIC extension, if the adversary listens to the radio link and meticulously collects several messages exchanged between the UE and the SN (MME in case of LTE and AN in case of interworking networks), samples of the following plain text and its corresponding cipher text will be available to it:

1. *A RIC (plain text) and its corresponding ERIC (cipher text)*: The attacker can obtain one such pair by extracting the RIC and the $ERIC$ part of a $DMSI$ (Equation 3.6.4).

2. *An ERAND (cipher text) and the RIC (plain text) embedded into it*: To obtain one such pair, the attacker has to pair up the RIC contained in the most recently transmitted $DMSI$ (by the UE) with the most recent $ERAND$ that was sent to the UE (by the SN).

Therefore, the method used to encrypt a plain text to obtain the respective cipher text has to be cryptographically strong to be resistant against known plain text attacks. In order to achieve this, the following measures are taken in the example algorithms that we have proposed (Section 3.7):

1. While creating a 128 bit $ERAND$ from a b bit RIC, the b bit RIC is masked, randomised and converted into a 128 bit value, before it is ciphered using AES. As stated in [66], the most efficient key-recovery attack for the current AES standard is exhaustive key search. Which means, obtaining information from given plaintext-ciphertext pairs about other plaintext-ciphertext pairs cannot be done more efficiently than by determining the key by exhaustive key search. The expected effort of exhaustive key search depends on the length of the cipher Key, which in this case is 2^{127} applications of AES.

2. For generating a 128 bit $ERIC$ from a b bit RIC, the b bit RIC is first padded with a 32 bit sequence number SEQ and a $128-32-b$ bit random number. The resultant 128 bit value is then encrypted using AES to obtain $ERIC$.

6.7.3 DoS and DDoS Attack

A Denial-of-Service attack (DoS attack) is an attempt to make a service unavailable to its intended users. Such an attack involves flooding the target device with fake communications requests, such that the target device cannot respond to legitimate request or responds so slowly as to be considered effectively unavailable [67][68]. A Distributed Denial of Service Attack (DDoS) occurs when multiple systems flood the target device with computationally intensive service requests simultaneously [69]. In this case, an attacker uses several mobile devices to simultaneously launch attack against a particular network server [70].

The following type of DDoS attack is possible against the HN in 3GPP mobile systems. Several fake UEs simultaneously initiates AKA procedures by transmitting multiple arbitrarily selected $IMSIs$ towards the HN. Such an attack will make the HN busy with the burden of generating several AVs at the same time. This would deny services to genuine subscribers or will increase response time of the HN to genuine requests. A similar attack in an AKA with E2EUIC extension would involve simultaneous transmission of multiple arbitrarily generated $DMSIs$ from several fake UEs towards the HN. This attack will not be as effective, because with no knowledge of the key Ki, it is impossible for the attacker to generate a valid $DMSI$. Even if the attacker generates spurious $DMSIs$ or replays previously transmitted $DMSIs$, such requests will be rejected straight away by the HN. This is because, the HN will detect an invalid $DMSI$ immediately after it receives it, by comparing the RIC with the $ERIC$ part of a received $DMSI$ and by verifying the freshness of the sequence number SQN_{UE} contained in the received $DMSI$. Thus, the consequences of a DDoS attack against an AKA with E2EUIC should not be worse than the already possible attack of initiating a large amount of authentication attempts presenting different forged permanent identities.

6.7.4 Fake Serving Network (Impersonation)

A fake serving network is an impersonated SN/AN that drowns the signals of a legitimate SN/AN with its own signals and presents itself to the UE as a

genuine SN/AN [21]. Such a SN can request the UE for its permanent identity, in response to which the UE has to transmit its $IMSI$ in plain text. In an AKA with E2EUIC, the subscriber is not vulnerable to such an attack because the UE in this case transmits its $DMSI$ (instead of the $IMSI$) when it receives a request for the permanent identity.

6.7.5 Corrupt Serving Network

A corrupt serving network is a genuine serving network having legitimate service agreement with the HN, but with malicious intention. Such a serving network may clandestinely share precious identity confidentiality related information entrusted to it by the UE and the HN [71]. In an AKA with E2EUIC extension (unlike UMTS-AKA, EPS-AKA and EAP-AKA), the $IMSI$ of a roaming UE is never disclosed to the SN. A dynamic and short lived $DMSI$ is transmitted in place of the $IMSI$. This protects the identity privacy of the subscriber even from SNs/ANs with malicious intention.

6.7.6 Eavesdropping

Eavesdropping is the act of secretly listening to the private conversation of others without their consent or knowledge [25]. In mobile networks proposed by 3GPP, the UE and the SN (MME in case of LTE and AN in case of interworking networks) are connected through a wireless link, which is vulnerable to eavesdroppers due to its open nature. All the mobile systems proposed by 3GPP, provides opportunities to eavesdroppers by allowing transmission of the $IMSI$ in clear text through the wireless link. This situation is improved when E2EUIC is adopted to the AKA protocol used in these networks, where a dynamic identity called $DMSI$ is transmitted instead of the $IMSI$.

6.8 Summary

In this chapter, the security extension that we have proposed for relaxed trust requirement for roaming in 3GPP mobile systems for improved subscriber identity privacy (i.e., E2EUIC) is analysed using various methods. A formal analysis of E2EUIC using an enhanced BAN logic called AUTLOG proves that E2EUIC meets its security goals. A computational cost analysis shows that E2EUIC can be adopted to the AKA protocols used in 3GPP mobile systems with the existing infrastructure. A time complexity analysis of E2EUIC provides the operator with an idea of the complexity it can expect with respect to increase in number of subscribers, if E2EUIC is to be adopted. A space overhead analysis gives an estimate of the amount of memory space needed at the UE and the HN to implement E2EUIC. A communication overhead analysis makes an assessment of the number of messages to be introduced/removed when E2EUIC is to be adopted to the existing AKA protocols. A security analysis of E2EUIC proves its robustness against threats that are perceived against it.

Chapter 7

Review of Literature and Discussion

The authentication and key agreement protocols that are used for access security in mobile systems developed by 3GPP do not ensure perfect identity privacy to the subscribers from eavesdroppers and visited/serving networks. In order to improve this situation, several works are proposed in the literature. In this chapter, we review the state of the art and some of the proposals in this area.

7.1 Introduction

The permanent identity of the subscriber is an important information that is used for purposes like identity presentation, authentication, authorisation, billing, etc. Its knowledge should be restricted to the UE and the HN. Exposure of the same to a third party, leads to compromise in identity privacy of the subscriber. An inherent problem with access security in mobile systems is that the identity of the subscriber has to be presented through the vulnerable wireless media before any kind of ciphering is possible.

7.2 Identity Privacy in 3GPP Mobile Systems

In order to find the status of the subscriber's identity privacy in mobile systems developed by 3GPP, we looked into some of the prominent technologies developed by them, viz., UMTS, LTE, 3GPP-WLAN and Non3GPP-EPS. The state of the art security architecture and the authentication and key agreement protocols used in these technologies are presented in Chapter 3 (Section 3.2, Section 3.3), Chapter 4 (Section 4.2, Section 4.3) and Chapter 5 (Section 5.2, Section 5.4). It is observed that the security mechanisms used for identity privacy in these technologies, as presented in Chapter 3 (Section 3.4), Chapter 4 (Section 4.4) and Chapter 5 (Section 5.5, Section 5.6), are based on replacing the permanent identity with temporary identities and pseudonyms. The main focus in these technologies is to protect the permanent identity from adversaries in the radio link, whereas the intermediary networks like the serving networks are treated trustworthy. As a result, in situations when the temporary identities/pseudonyms fail to identify the subscriber, the serving networks are given complete liberty to request the subscriber for its permanent identity. Such a request forces the subscriber to transmit its permanent identity in clear text through the vulnerable radio link. Thereby, providing eavesdroppers and fake/malicious serving networks with openings to compromise the subscriber's identity privacy.

7.3 Related Work

In the past few years researchers have proposed several schemes and authentication protocols for improved subscriber identity privacy in mobile networks. Some of the initial works in this area were focused on improving identity privacy of the subscriber over the radio access link between the UE and the SN. The need to protect the identity privacy of the subscriber over the entire path between the UE and the HN is now well recognised. In [72], Herzberg *et al.* pointed out that in an ideal situation no entity other than the subscriber himself and a responsible authority in the subscriber's home domain should know the real identity of the user. Even the authority in the visited network should not have any idea about

the real identity. In [73], Samfat *et al.* specified the requirement of hiding the subscriber identity from foreign authorities. In [7], Barbeau *et al.* expressed the need for concealment of the IMSI to be pushed up from the radio access network to the interface between the SN and the HN.

7.3.1 Proposals to Improve Identity Privacy Over the Radio Access Link

Many schemes were proposed to improve identity privacy of the subscriber over the vulnerable radio access link between the UE and the SN.

Asokan [74], proposed a solution to protect information about movements and activities of mobile entities from onlookers through limited disclosure of information. In this solution, which uses both public and shared key crypto system, the UE sends an encrypted message and its group identifier to identify itself.

Lin *et al.* [75] proposed a protocol that uses secret key cryptographic techniques in combination with public key cryptographic techniques. In this protocol, a secret authentication key is established between the subscriber and the SN for mutual authentication. During identity presentation, the identity of the subscriber is protected from eavesdropping by encrypting it with the HN's public key.

Horn *et al.* [76] proposed an efficient public key protocol for mutual authentication and key exchange designed for third generation mobile communication systems. In this protocol the signature of the subscriber is encrypted with the session key K between the UE and the SN. This is done to protect the subscriber's identity. If the signature is not encrypted, an attacker would be able to detect the identity of UE by verifying the signature.

In order to provide subscriber anonymity, Park *et al.* [77] introduced a new method of computing temporary identity. In this method, a temporary identity is initially computed by the UE and updated by both the UE and network side during every execution of the protocol. The method is based on public key

cryptosystems that include digital signature and Diffie-Hellman key exchange. The challenge-response authentication mechanism used in this method uses nonce that are uniquely generated by protocol entities.

Barbeau et al. [7] proposed three different solutions that uses one-time aliases: a coupon-based solution, a primary key infrastructure based solution and an anonymous number-based solution. Each solution, provides identity concealment against a passive attacker which can eavesdrop over the radio access link and an active attacker which can inject messages over the radio access link.

Juang *et al.* [78] proposed a scheme in which random numbers, message authentication codes and one way hash functions are used. The HN generates a secret token Wi which is distributed to the UE during an authentication and key agreement via a secure channel. For identity presentation, the UE does not transmit its IMSI in plain text through the radio link. Instead, it generates and transmits a token Pi that is generated by performing a XOR operation between Wi and the IMSI.

Forsberg *et al.* [79] proposed some countermeasures to UE tracking, like ciphering the radio resource controller messages, periodic temporary identity reallocation on one cell and use of discontinuous sequence numbers.

7.3.2 Proposals to Provide End to End Identity Privacy

Several schemes and protocols are proposed to provide identity privacy to the subscriber over the entire path between the UE and the HN. In such schemes, efforts are made to protect the permanent identity of the subscriber even from the SN.

Samfat *et al.* [73] proposed an alias based method that protects the identity of the subscriber from unauthorised third party. The design of this method is borrowed from *KryptoKnight*, an authentication and key distribution service developed by IBM research [80]. In this method public key cryptography is used for computation of aliases.

He *et al.* [81] proposed an anonymous-ID-based scheme that uses both sym-

metric and asymmetric crypto systems. In this scheme, an anonymous ID that is created by using blind signature technique is used to replace the real ID of an authorised mobile device. With this scheme, the authors have designed an architecture that is able to provide mobile subscribers with complete control over their location privacy while allowing the administration to authenticate legitimate mobile subscribers.

Koien *et al.* [82] proposed a privacy preserving 3-way authentication and key agreement protocol that uses secure multi party computation, identity based encryption and deffie hellman exchange to protect subscriber identity and location data from eavesdropping. It also provides location privacy with respect to the HN.

Godor *et al.* [83][84] proposed an algorithm that uses public key infrastructure, public key certificates and sequence numbers to prevent transmission of IMSI in clear text. Public keys are used for secured communication of the messages, certificates are used for mutual authentication, and sequence numbers are used to avoid replay attacks.

Yang *et al.* [85] proposed a novel approach of anonymous and authenticated key exchange protocols for a roaming subscriber and a visiting server to establish a random session key in such a way that the visiting server authenticates the subscriber's home server without knowing exactly who the subscriber is. A network eavesdropper cannot find out the subscriber's identity either. In addition, visited servers cannot track the roaming subscriber's movements and whereabouts even if they collude with each other. The proposed approach is generic and built upon secure two-party key establishment protocols.

Li *et al.* [86] proposed an enhanced authentication and key agreement protocol that is based on wireless public key infrastructure. This protocol is designed to overcome the existing vulnerabilities such as as disclosure of user identity, man-in-middle attack, etc., in EPS AKA which is the AKA protocol used in LTE/SAE architecture.

Varadharajan *et al.* [87] proposed three privacy preserving protocols for mobile communications. The protocols are based on hybrid scheme involving

a combination of public key and symmetric key' based crypto systems. Each subscriber is issued a subliminal identity (a form of alias) by the HN at the time of initial registration. A subscriber's subliminal identity can be updated at the end of each authentication session as a part of the protocol.

A scheme called the hybrid approach of authentication protocol is proposed by Al-Fayoumi *et al.* in [88]. In this scheme, both symmetric and asymmetric keys are used combined with hash technique. Authentication between the UE and the HN relies on the long term shared secret key Ki, where as authentication between the UE and the SN depends on a public/private key pair and a session key. In this scheme, the concepts of temporary identity and key refreshment are adopted. Temporary identities generated at the HN protects the subscribers true identity, and key refreshment can make the authentication process more secure.

Another hybrid scheme was proposed by Zhu *et al.* that uses both asymmetric and symmetric key crypto systems [89]. This scheme, which is based on hash functions and smart cards uses public key crypto systems, but the user equipment only do symmetric encryption and decryption. The scheme takes only one round of message exchange between the mobile user and the visited network, and one round of messages exchange between the visited network and the corresponding home network where keys between the mobile user and the visited network are for one time usage. In order to hide the real identity of the subscriber, a random number, a one way hash function and XOR operation is used.

Jiang *et al.* [90] proposed two sets of mutual authentication and key exchange protocols with anonymity property for roaming service by using secret-splitting principle and self-certified scheme. The proposed authentication protocols use the temporary identity of a subscriber instead of the real one. The temporary identity is prearranged and distributed by the home network in advance or temporarily generated by encrypting the real identity. In the secret splitting scheme, the HN generates a pseudonym from a secret m-bit random number and the permanent identity of the subscriber using a strong one way hash function and XOR operation. The self-certified scheme combines the advantages of certificated-based and identity-based public key crypto systems.

Sattarzadeh *et al.* [91] proposed a mechanism called improved user identity confidentiality. In this mechanism, anonymous tickets are employed as aliases for the IMSI. The IMSI is never exposed over any interface including the wired path. The TMSI plays the same role as in UMTS-AKA. It uses UMTS symmetric cryptography algorithms to ensure anonymity of tickets. A separate module called anonymous ticket manager module is introduced at the HN to handle ticket related functions.

Tang *et al.* [92] proposed a scheme that employs one-time alias technique with a secure trust delegation mechanism. It does not introduce security vulnerability to the underlying authentication scheme and is able to conceal the real identity of the subscriber with respect to both eavesdroppers and visited serving networks. The scheme achieves identity concealment without sacrificing authentication efficiency. Due to the low complexity, the scheme adds little overhead to the UE for privacy protection, and a moderate overhead is added on the fixed network part.

Pereniguez *et al.* [93] studied the problem of providing identity privacy and fast network access in the context of the next generation heterogeneous networks, where EAP is a preferred framework for carrying out the authentication process independently of the underlying technology. The authors have proposed a privacy framework that is able to maintain user anonymity during authentication and fast re-authentication processes that are based on EAP. It relies on a multi layered pseudonym model that defines a n-tuple of pseudonyms that are frequently renewed.

Yang *et al.* [94] proposed two roaming protocols for improved subscriber anonymity. The first protocol that is based on identity based signature, whereas the second one is based on group signature.

He *et al.* [95] proposed a secure and light weight authentication scheme with user anonymity. It is simple to implement for mobile user since it only performs symmetric encryption/decryption operation. Having this feature, it is more suitable for the low power and resource limited mobile devices. In addition, it requires four message exchanges between the user equipment, the serving

network and the home network. Thus, this protocol achieves both computation and communication efficiency.

He *et al.* [96] proposed a scheme that uses low cost functions such as one way hash functions and XOR operations to achieve security goals. It uses nonces instead of timestamps to avoid the clock synchronisation problem. Therefore, an additional clock synchronisation mechanism is not needed. It only requires four message exchanges between the user equipment, the serving network and the home network.

Zhou *et al.* [97] proposed a protocol that uses modular exponentiation, hash operation and XOR operation to hide the real identity of the subscriber. It is suitable for mobile devices due to its acceptable computation cost, high level security and lower interaction rounds.

Chen *et al.* [98] proposed a scheme that uses symmetric cryptographic and hash operation primitives to provide identity privacy to the subscriber. Besides, it takes only four message exchanges among the user equipment, the serving network and the home network.

He *et al.* [99] proposed a privacy preserving universal authentication protocol called Priauth. It uses group signature scheme to achieve anonymity. It requires the roaming user and the foreign server to be involved in each protocol run, and the home server can be off-line.

Feng *et al.* [100] proposed an anonymous identity authentication based on the self certified public key system.

Liu *et al.* [101] proposed a novel privacy preserving registration protocol that achieve user anonymity against home server and eavesdroppers via a robust temporary identity, local user revocation with untraceability support, and secure key establishment.

Jiang *et al.* [102] proposed an enhanced authentication scheme with privacy preservation based on a concept called quadratic residue assumption.

7.4 Discussion

In the early days of mobile communications, when researchers started recognising identity privacy as an important security issue, several schemes and protocols were proposed to improve identity privacy of the subscriber over the radio access link between the UE and the SN. Some of these schemes and protocols, as discussed in Section 7.3.1, are the ones proposed by Asokan [74], Lin *et al.* [75], Horn *et al.* [76], Park *et al.* [77], Barbeau et al. [7], Juang *et al.* [78], Forsberg *et al.* [79], etc. A basic assumption in these proposals is that the SN is trustworthy. In present day context, where multiple operators collaborate among each other to extend their services across a wider coverage area, such trust requirement is a limiting factor. Realising this, researchers are now concentrating on providing identity privacy to the subscriber over the the entire path between the UE and the HN. In these solutions, the need to protect the identity privacy of a subscriber from the SN is well recognised.

Several schemes and protocols were proposed to provide end to end identity privacy to the subscriber. In many of these proposals, asymmetric key cryptography is used, viz., the protocols proposed by Samfat *et al.* [73], Godor *et al.* [83][84], Yang *et al.* [85], Li *et al.* [86], He *et al.* [99], Feng *et al.* [100], etc. In asymmetric key cryptography, two keys are used: the public key known to the public, and the private key known only to the subscriber [103]. Asymmetric key cryptographic algorithms are slow because they are designed to work through computationally intensive mathematical functions, like factoring of large prime numbers, etc. They are almost 1000 times slower than symmetric techniques [104][105]. Considering the limitations on processing power and energy consumption at the UE, such solutions are not suitable for mobile systems.

Symmetric key cryptographic techniques are fast and suitable for the UE, whereas asymmetric key cryptographic techniques can provide identity privacy in a natural way. In order to exploit the advantages of both, in many proposals for end to end subscriber identity privacy, a hybrid approach is adopted. Where, a combination of both asymmetric key and symmetric key crypto systems are used. Some of these protocols are the ones proposed by He *et al.* [81], Varadhara-

jan *et al.* [87], Al-Fayoumi *et al.* [88], Zhu *et al.* [89]. Koien *et al.* [82], etc. In a majority of such proposals, the UE do only symmetric key encryption/decryption whereas the HN do asymmetric key encryption/decryption. In mobile systems, the HN needs to process several simultaneous request from the subscribers at the same time. Hence, computationally intensive algorithms should be avoided at the HN, because such algorithms may increase the overall processing time of the subscriber's requests [106]. Since a protocol with a hybrid approach can introduce significant amount of computational overhead at the HN, a protocol that uses computationally light cryptographic operations like symmetric key encryption/decryption operations is more desirable. Moreover, there are several criticisms against some of the much discussed hybrid protocols that were proposed recently. Wong *et al.* [107] found that Varadharajan *et al.*'s protocols are vulnerable to several attacks. Lee *et al.* [108] demonstrated that the scheme proposed by Zhu *et al.* [89] has three security weaknesses viz., it cannot achieve perfect backward secrecy, it cannot achieve mutual authentication, and it cannot protect against a forgery attack. Thus, in order to improve the shortcomings of the scheme proposed by Zhu *et al.*, Lee *et al.* proposed and enhancement. But, Wu *et al.* pointed out that Lee *et al.*'s proposed fix fails to preserve anonymity as claimed and then proposed yet another fix to address the problem [109]. However, Wu *et al.*'s fix is not without limitations. Mun *et al.* [110] found weaknesses in Wu *et al.*'s scheme such as failing to achieve anonymity and perfect forward secrecy, and disclosing of legitimate users password. Therefore, they proposed a new enhanced scheme that uses elliptic curve diffie hellman to overcome these weaknesses and improve performance. Moreover, Chang *et al.* [111] found that Lee *et al.*'s scheme cannot provide anonymity under the forgery attack and the heavy computation cost may consume battery power expeditiously for mobile devices. Therefore, Chang *et al.* proposed a novel authentication scheme to overcome these weaknesses. Youn *et al.* [112] showed that even Chang *et al.*'s scheme fails to achieve anonymity by providing four attack strategies. More recently, Zeng *et al.* demonstrated that due to an inherent design flaw in the original Zhu *et al.*'s scheme, the latter and its successors are unlikely to provide anonymity [113]. The communication and computation complexity of the proto-

col proposed by Koien *et al.*, as agreed by the author's themselves, is overall quite high. In this protocol, the UE is burdened with many asymmetric encryption and decryption operations.

Keeping the constraints of mobile computing environment in mind, several schemes and protocols were proposed recently. In these proposals, computationally light techniques that use symmetric key based crypto systems, hash functions, XOR operations, temporary identities, alias, etc., are used. Some of these protocols are the ones proposed by Jiang *et al.* [90], Sattarzadeh *et al.* [91], Tang *et al.* [92], Pereniguez *et al.* [93], He *et al.* [95], Zhou *et al.* [97], He *et al.* [96], Chen *et al.* [98], Liu *et al.* [101], Jiang *et al.* [102], etc. All these proposals claim to achieve end to end subscriber identity privacy. However, none of them are in the general lines of approach with the authentication and key agreement protocols used in mobile systems developed by 3GPP. Therefore, if any of these schemes are to be adopted, they will bring about major changes to the existing security arrangement.

For a mobile operator that already has a big subscriber base, changing over to a completely new AKA protocol is a big challenge. Moreover, the AKA protocols used in mobile systems standardised by 3GPP have already proven their efficiency with respect to other security features, through their successful real time implementations over the years. Therefore, an ideal scheme for enhanced identity privacy in a 3GPP defined mobile system would be the one that can be easily configured into the existing AKA protocol. At the same time, implementation of such a scheme should be restricted only to the home operator. Visited/serving networks that may even belong to third party operators should not be expected to participate equally. A simple yet effective solution that takes these critical issues into consideration is therefore needed. Taking cognisance of this, we proposed E2EUIC, a security extension for the AKA protocols used for access security in 3GPP mobile systems.

In order to evaluate the performance of E2EUIC, we present a performance comparison in Table 7.1. In this table, we compare the AKA protocols used in 3GPP mobile systems (viz., UMTS-AKA, EPS-AKA and EAP-AKA) when

Table 7.1: Performance comparison.

Protocols	Cryptographic operations a: asymmetric encryption/decryption s: symmetric encryption/decryption h: hash operation x: XOR operation **at the UE**	**at the SN**	**at the HN**	Number of messages	Identity privacy from SN	Impact on SN	In line with 3GPP AKA
Samfat *et al.* [73]	1a+3s+1h+3x	2a+1h+1x	3a+6s+1h+6x	4	Yes	Yes	No
Asokan *et al.* [74]	1a+4s	4s	1a+1s+1h	5	No	Yes	No
Lin *et al.* [75]	3a+1s+2h	2a+3s+2h	1a+1s+5h	4	No	Yes	No
Park *et al.* [77]	2a+3s+4h+1x	2a+3s+6h	2a+2s+4h+1x	5	No	Yes	No
Juang *et al.* [78]	6h+2x	2h	6h+2x	7	No	Yes	No
koien *et al.* [82]	5a+1h	5a+3s	2a+3s	8	Yes	Yes	No
Godor *et al.* [83]	3a	3a	3a	5	Yes	Yes	No
Yang *et al.* [85]	1a+6h+2x	1a+5h+1x	1h	3	Yes	Yes	No
Li *et al.* [86]	2a	2a	1a	5	Yes	Yes	No
Varadharajan *et al.* [87]	5s+2h	1a+4s+3h	2a+2s+2h	6	Yes	Yes	No
Al-Fayoumi *et al.* [88]	1a+3s+1h	1a+1s+1x	1a+2s	4	Yes	Yes	No
Zhu *et al.* [89]	2s+2h+3x	1a+1s+2h+1x	2a+1s+5h+3x	4	Yes	Yes	No
Sattarzadeh *et al.* [91]	6h	None	6h	6	Yes	Yes	No
Tang *et al.* [92]	1s	3s	2s	4	Yes	Yes	No
He *et al.* in [96]	5h	3h	7h	4	Yes	Yes	No
Zhou *et al.* [97]	2a+3h+2x	2h+1x	1a+5h+2x	4	Yes	Yes	No
Chen *et al.* [98]	2s+9h	2s+3h	2s+6h	4	Yes	Yes	No
Feng *et al.* [100]	3a+1s+5h+1x	6a+2h	4a+1s+3h	5	Yes	Yes	No
Lee *et al.* [108]	2s+4h+3x	2a+2s+4h+1x	2a+1s+5h+3x	4	Yes	Yes	No
UMTS-AKA	5h+1x	None	5h+1x	5	No	-	-
UMTS-AKA-with-E2EUIC	2s+5h+2x	None	2s+5h+2x	5	Yes	No	Yes
EPS-AKA	6h+2x	None	6h+2x	5	No	-	-
EPS-AKA-with-E2EUIC	2s+6h+3x	None	2s+6h+3x	5	Yes	No	Yes
EAP-AKA	8h+1x	None	4s+10h+1x	24	No	-	-
EAP-AKA-with-E2EUIC	2s+8h+2x	None	6s+10h+2x	24	Yes	No	Yes

E2EUIC is adopted to it, with some of the protocols discussed in this chapter. The protocols are compared in terms of the number/kind of cryptographic operations used, the number of message transmissions involved, identity privacy from the SN, impact on the SN, and on the basis of whether a protocol is in general line of functioning with the AKA protocols used in 3GPP mobile systems. The following cryptographic operations, listed according to their decreasing computational complexity, are used in the various proposals and are considered for the performance comparison.

1. Asymmetric encryption/decryption (a).
2. Symmetric encryption/decryption (s).
3. Hash operation (h).
4. XOR operation (x).

Asymmetric operations are computationally intensive, whereas symmetric encryption/decryption and hash operations are computationally light and are suitable for resource-limited mobile devices. XOR operation is a combination of just two AND and one OR operation and thus can be easily executed in a mobile device.

In UMTS-AKA protocol (Chapter 3, Section 3.3), 5h+1x cryptographic operations are performed at each of the UE and the HN, with no cryptographic operations at the SN. The same is applicable for EPS-AKA and EAP-AKA, as both of them are based on UMTS-AKA. However, in EPS-AKA (Chapter 4, Section 4.3), due to the need to generate an additional K_{ASME}, which involves 1h+1x operations, a total of 6h+2x cryptographic operations are performed at each of the UE and the HN. And, in EAP-AKA (Chapter 5, Section 5.4), due to the need to generate new keying materials, encrypted and integrity protected pseudonyms, message authentication codes, etc., which involves 5h+4s operations at the HN and 3h operations at the UE, a total of 8h+1x and 4s+10h+1x cryptographic operations are performed at the UE and at the HN respectively. This difference in computational complexity between the various AKA protocols

used in 3GPP mobile systems does not have any impact on the performance of E2EUIC. Across all the AKA protocols used in 3GPP mobile systems, E2EUIC can be equally adopted with a fixed cryptographic overhead of 2s+1x operations.

From the performance comparison in Table 7.1, it is clear that while several of the proposals in this area treat the SN as trustworthy by focussing only on the radio path, many do not take the limitations of the UE and the load on the HN (to process several simultaneous request) into consideration, by using computationally intensive asymmetric encryption/decryption. Whereas, none of the proposals, to the best of our knowledge, are in general lines of approach with the AKA protocols used in mobile systems developed by 3GPP. And, none of them consider the importance of keeping the SNs away from making any major changes to the existing mode of operation in a 3GPP mobile system. E2EUIC is in general lines of functioning with the AKA protocols used in 3GPP mobile systems. Hence, it can be easily adopted as an extension to the AKA protocols used in 3GPP mobile systems. Moreover, it can be implemented at the operator's level (i.e., at the HN and the SIM of the UE), without tasking the intermediary SN, with an execution and implementation cost which is feasible with the current infrastructure. In E2EUIC, computationally light symmetric encryption/decryption and XOR operations are used. When E2EUIC is adopted to an AKA protocol, no additional messages are introduced, as the information needed to be exchanged between the agents are communicated as a part of the existing messages in the AKA protocol.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

To facilitate roaming in mobile systems, so that a subscriber of one operator can use the services of another operator when inside the latter's coverage area, a-priori roaming agreements with respect to issues like billing, quality of service, security, etc., needs to be established between mobile operators. A factor that directly affects such agreements is the level of trust requirement expected from an operator. Lower is the trust requirement with respect to a particular issue, more is the convenience for an operator to come to an agreement with another operator.

The AKA protocols used to provide access security in mobile systems standardised by 3GPP does not ensure perfect identity privacy to the subscribers. There are occasions during authentication and key agreement when the permanent identity gets compromised over the vulnerable wireless link between the user equipment and the visited/serving network. Moreover, the serving network that may even belong to a third party operator, is fully trusted with the permanent identity of the subscriber. This situation can be attributed to the existing trust model adopted in the mobile systems developed by 3GPP, where the subscriber and the home network has to confer a high amount of trust on the visited/serving network (HN $\rightarrow$ SN; UE $\rightarrow$ SN). Such trust requirement not only makes identity privacy of the subscriber vulnerable, but also complicates roaming agreements.

If we relax the requirement of having to trust the visited/serving networks from the UE/HN, it then improves the subscriber's identity privacy. Moreover, it opens up an opportunity to have on-demand/on-the-fly roaming agreements between the mobile operators, instead of the current a-priori agreements. This would ensure that a subscriber will be serviceable in any location as long as there is at least one network serving that location. With more and more operators taking a plunge into the competitive mobile/cellular market, with each of them promising an ubiquitous service to the subscriber, interoperability amongst them is a key issue. Thus, the benefits of a relaxed HN $\rightarrow$ SN/UE $\rightarrow$ SN trust requirement would be difficult to ignore in the foreseeable future.

Taking the benefits of a relaxed HN $\rightarrow$ SN/UE $\rightarrow$ SN trust requirement into consideration, we have proposed a new trust model where the only trust requirement is the need for the UE to trust the HN (UE $\rightarrow$ HN); all the other trust requirements like the need for the HN to trust the SN (HN $\rightarrow$ SN) and the need for the UE to trust the SN (UE $\rightarrow$ SN) is relaxed with respect to the subscriber's permanent identity.

To implement our proposed trust model, we devised a security extension called E2EUIC for enhanced identity privacy and relaxed trust requirement for roaming agreements. E2EUIC is developed as a common solution that can be adopted to all the AKA protocols used for access security in the mobile and interworking systems proposed by 3GPP. E2EUIC takes an end to end approach, where the knowledge of the permanent identity is restricted to the UE and the HN. The permanent identity is never transmitted at any stage of the path between the subscriber and the HN. Thereby, eliminating the need to transmit the permanent identity through the vulnerable radio link and the need to trust the SN with the permanent identity. Thus, when E2EUIC extension is adopted to an AKA protocol used in a particular 3GPP mobile system, the benefits are two folds; one: it ensures an enhanced identity privacy to the subscriber, and two: it relaxes trust requirement for roaming between the operators. In order to demonstrate the adaptability of E2EUIC, we have presented its functioning with some prominent 3GPP mobile systems like UMTS, LTE, 3GPP-WLAN

and Non3GPP-EPS. We also performed a statistical analysis, a formal analysis, an overhead analysis and a security analysis of E2EUIC, to verify its logical correctness, feasibility of implementation (in terms of computation, complexity and memory space) and its robustness. Results of these analysis were found satisfactory.

8.2 Future Work

There is scope for extending the work reported in this thesis in the following directions:

- E2EUIC relaxes trust requirement for roaming with respect to identity privacy of the subscriber. In order to realise the full benefit of our proposed trust model, there is a need to relax trust requirement for roaming with respect to other security aspects like ciphering and integrity protection as well. Therefore, as a future work, one may explore the use of end to end ciphering solutions like IPSec to further relax trust requirement for roaming in 3GPP mobile systems.

- While it is important to protect the identity privacy of the subscriber from adversaries, it is equally important for mobile operators to allow law enforcement agencies to intercept identity of the subscriber for prevention of crimes. In an AKA protocol with E2EUIC, even though the permanent identity of the subscriber is concealed from adversaries using $DMSIs$, details of $DMSI$ to $IMSI$ mappings are maintained in the HN. As a future work, these mappings can be used to devise a mechanism that allows lawful interception with E2EUIC.

- Unlike the existing AKA protocols used in 3GPP mobile systems, in an AKA protocol with E2EUIC the permanent identity is not shared with the visited/serving network. Therefore as a future work, the impact of this in the existing mechanism used for billing of the roaming subscribers needs to be studied.

Bibliography

[1] Whalen, T. Mobile devices and location privacy: Where do we go from here? *IEEE Security & Privacy* **9**(6), 61–62, 2011.

[2] TWB. Mobile phone access reaches three quarters of planet's population, The World Bank Press Release. `http://www.worldbank.org /en/news/2012/07/17/mobile-phone-access-reaches-three-quarters-planets-population`, 2012.

[3] Pohjola, O. P. *et al.* Roaming dynamics in GPRS and beyond: Options and strategies, in *IEEE International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2004)*, Vienna, Austria, 99–104.

[4] Fu, Z. J. *et al.* AAA for spontaneous roaming agreements in heterogeneous wireless networks, in *Autonomic and Trusted Computing (ATC 2007)*, Hong Kong, China, 489–498.

[5] 3GPP. Numbering, addressing and identification, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23003.htm`, 2012.

[6] Liu, Z. Y. *et al.* A fast suffix matching method in network processor, in *IEEE International Conference on Computational Intelligence and Security (CIS' 08)*, Suzhou, China, 405–410.

[7] Barbeau, M. & Robert, J. M. Perfect identity concealment in UMTS over radio access links, in *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob 2005)*, Montreal, Canada, 72–77.

[8] Boman, K. *et al.* UMTS security, *Electronics & Communication Engineering Journal* **14**(5), 191–204, 2002.

[9] Niemi, V., Nyberg, K. & Wiley, J. *UMTS Security,* Wiley, United States, 2003.

[10] 3GPP. Universal terrestrial radio access network (UTRAN) overall description, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/ 25401.htm`, 2012.

[11] Smith, C. *3G Wireless Networks,* McGraw-Hill Inc., United States, 2006.

[12] Motorola. Long term evolution (LTE): A technical overview, Technical Whitepaper. `http://www.motorola.com/web/Business/Solutions/Industry Solutions/Service Providers/Wireless Operators/LTE/_Document/Static Files/6834_MotDoc_New.pdf`, 2007.

[13] 3GPP. Evolved universal terrestrial radio access network (E-UTRAN); architecture description, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/36401.htm`, 2012.

[14] 3GPP. LTE overview, Tech. Rep. `http://www.3gpp.org/LTE`, 2013.

[15] 3GPP. 3GPP system to wireless local area network (WLAN) interworking; system description, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23234.htm`, 2012.

[16] 3GPP. Architecture enhancements for non-3GPP accesses, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23402.htm`, 2012.

[17] Xenakis, C. & Merakos, L. Security in third generation mobile networks, *Computer communications* **27**(7), 638–650, 2004.

[18] Sankaran, C. B. Network access security in next-generation 3GPP systems: A tutorial, *IEEE Communications Magazine* **47**(2), 84–91, 2009.

[19] 3GPP. 3G security; security architecture. Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/33102.htm`, 2012.

[20] 3GPP. 3GPP system architecture evolution (SAE); security architecture, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/33401.htm`, 2012.

[21] Zhang, M. & Fang, Y. Security analysis and enhancements of 3GPP authentication and key agreement protocol, *IEEE Transactions on Wireless Communications* **4**(2), 734–742, 2005.

[22] Meyer, U. & Wetzel, S. A man-in-the-middle attack on UMTS, in *ACM workshop on Wireless security (WiSe 2004)*, Philadelphia, USA, 90–97.

[23] Koien, G. M. An introduction to access security in UMTS, *IEEE Wireless Communications* **11**(1), 8–18, 2004.

[24] Choudhury, H. *et al.* A new trust model for improved identity privacy in cellular networks, *International Journal of Computer Applications* **56**(14), 1–8, 2012.

[25] Zhang, Y. *et al. Handbook of Research on Wireless Security*, Information Science Reference-Imprint of: IGI Publishing, Hershey, PA, 2008.

[26] Diab, W. B. & Tohme, S. VPN solution for securing voice over third generation networks, in *IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA 2008)*, Bangalore, India, 1–6.

[27] Xenakis, C. *et al.* A network-assisted mobile VPN for securing users data in UMTS, *Computer Communications* **31**(14), 3315–3327, 2008.

[28] Arkko, J. *et al.* Using IPsec to protect mobile IPv6 signaling between mobile nodes and home agents, RFC: 3776. `http://trac.tools.ietf.org/html/rfc3776`, 2004.

[29] Xenakis, C. & Merakos, L. IPsec-based end-to-end VPN deployment over UMTS, *Computer Communications* **27**(17), 1693–1708, 2004.

[30] Khan, M. *et al.* Vulnerabilities of UMTS access domain security architecture, in *International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'08)*, Phuket, Thiland, 350–355.

[31] 3GPP. Universal mobile telecommunications system (UMTS) architecture, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23101.htm`, 2012.

[32] 3GPP. Network architecture, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23002.htm`, 2012.

[33] 3GPP. 3G security; specification of the MILENAGE set: an example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5, and f5*, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/35205.htm`, 2012.

[34] Choudhury, H. *et al.* End-to-end user identity confidentiality for UMTS networks, in *IEEE International Conference on Computer Science and Information Technology (ICCSIT' 10)*, Chengdu, China, 46–50.

[35] Raj, P. Highlights on telecom subscription data as on 30th June 2012, Press Release, Telecom Regulatory Authority of India. `http://www.trai.gov.in/WriteReadData/PressRealease/Document/PR-TSD-Jun12.pdf`, 2012.

[36] Schneier, B. *et al.* The twofish team's final comments on AES selection, `http://homes.cs.washington.edu/ yoshi/papers/Misc/twofish-final.pdf`, 2000.

[37] Daemen, J. & Rijmen, V. *The Design of Rijndael: AES-the Advanced Encryption Standard,* Springer Verlag, Germany, 2002.

[38] Wikipedia. Advanced encryption standard, Article, Wikipedia the free encyclopedia. `http://en.wikipedia.org/wiki/Advanced-Encryption-Standard`, 2013.

[39] Keller, S. S. NIST-recommended random number generator based on ANSI X9. 31 Appendix A. 2.4 using the 3-key triple DES and AES algorithms, National Institute of Standards and Technology (NIST). `http://csrc.nist.gov/groups/STM/cavp/documents/rng/931rngext.pdf`, 2005.

[40] Easter, R. J. & Carolyn, F. Annex C: Approved random number generators for FIPS PUB 140-2, security requirements for cryptographic modules, NIST Federal Information Processing Standards (FIPS) Publications. `http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexc.pdf`, 2012.

[41] Seznec, A. & Sendrier, N. HAVEGE: a user-level software heuristic for generating empirically strong random numbers, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **13**(4), 334–346, 2003.

[42] Choudhury, H. *et al.* UMTS user identity confidentiality: An end-to-end solution, in *IEEE International Conference on Wireless and Optical Communications Networks (WOCN' 11)*, Paris, France, 1–6.

[43] Rukhin, A. *et al.* A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22 Revision 1a, National Institute of Standards and Technology (NIST). `http://csrc.nist.gov/rng/`, 2010.

[44] Forsberg, D. *et al. LTE Security,* Wiley, United States, 2012.

[45] 3GPP. General packet radio service (GPRS) enhancements for evolved universal terrestrial radio access network (EUTRAN), Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/23401.htm`, 2011.

[46] Choudhury, H. *et al.* Enhancing user identity privacy in LTE, in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012)*, Liverpool, UK, 949–957.

[47] 3GPP. Vocabulary for 3GPP specifications, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/21905.htm`, 2012.

[48] 3GPP. 3GPP system architecture evolution (SAE); Security aspects of non-3GPP accesses, Technical Specification. `http://www.3gpp.org /ftp/Specs/html-info/33402.htm`, 2012.

[49] Aboba, B. *et al.* Extensible authentication protocol, RFC: 3748. `http://tools.ietf.org/html/rfc3748`, 2004.

[50] 3GPP. 3G security; wireless local area network (WLAN) interworking security, Technical Specification. `http://www.3gpp.org/ ftp/Specs/html-info/33234.htm`, 2012.

[51] Aboba, B. & Beadles, M. The network access identifier, RFC: 2486. `http://tools.ietf.org/html/rfc2486.html`, 1999.

[52] Mockapetris, P. Domain names - implementation and specification, RFC: 1305. `http://www. ietf. org/rfc/rfc1035. txt`, 2004.

[53] Kaufman, C. *et al.* Internet key exchange protocol version 2 (IKEv2), RFC: 5996. `http://www.hjp.at/doc/rfc/rfc5996.html`, 2010.

[54] Choudhury, H. *et al.* Improving identity privacy in 3GPP-WLAN, in *IEEE International Conference on Computer Communications and Networks (ICCCN 2013)*, Nassau, Bahamas, Accepted.

[55] Arkko, J. *et al.* The network access identifier, RFC: 4282. `http://tools.ietf.org /html/rfc4282`, 2005.

[56] Rigney, C. *et al.* Remote authentication dial in user service (RADIUS), RFC: 2865. `http://www.hjp.at/doc/rfc/rfc2865.html`, 2000.

[57] Gritzalis, S. *et al.* Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification, *Computer Communications* **22**(8), 697–709, 1999.

[58] Burrows, M. *et al.* A logic of authentication, *Proceedings of the Royal Society A. Mathematical and Physical Sciences* **426**(1871), 233–271, 1989.

[59] Wedel, G. & Kessler, V. Formal semantics for authentication logics, in *European Symposium on Research in Computer Security (ESORICS' 96),* Rome, Italy, 219–241.

[60] Kessler, V. & Neumann, H. A sound logic for analysing electronic commerce protocols, in *European Symposium on Research in Computer Security (ESORICS' 98),* Louvain-la-Neuve, Belgium, 46–50.

[61] 3GPP. Formal analysis of the 3G authentication protocol, Technical Specification. `http://www.3gpp.org/ftp/Specs/html-info/33902.htm`, 2001.

[62] Harn, L. & Hsin, W. J. On the security of wireless network access with enhancements, in *ACM workshop on Wireless security (WiSe' 03),* San Diego, USA, 88–95.

[63] Granelli, F. & Boato, G. A novel methodology for analysis of the computational complexity of block ciphers: rijndael, camellia and shacal-2 compared, Tech. Rep., Department Of Information And Communication Technology, University Of Trento. `http://eprints.biblio.unitn.it/514/1/DIT-04-004.pdf`, 2004.

[64] Kavitha, T. & Sridharan, D. Security vulnerabilities in wireless sensor networks: A survey, *Journal of information Assurance and Security* **5**(1), 31–44, 2010.

[65] Stallings, W. *Network Security Essentials,* Prentice Hall, India, 2007.

[66] Daemen, J. & Rijmen, V. AES proposal: Rijndael, in *First Advanced Encryption Standard (AES) Conference (AES1 1998),* Ventura, California, 343–348.

[67] Carl, G. *et al.* Denial-of-service attack-detection techniques, *IEEE Internet Computing* **10**(1), 82–89, 2006.

[68] Patrick, P. C. *et al.* On the detection of signaling DoS attacks on 3G/WiMax wireless networks, *Computer Networks* **53**(15), 2601–2616, 2009.

[69] Douligeris, C. & Mitrokotsa, A. DDoS attacks and defense mechanisms: classification and state-of-the-art, *Computer Networks* **44**(5), 643–666, 2004.

[70] Rekhis, S. *et al.* Detection and reaction against DDoS attacks in cellular networks, in *International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008)*, Damascus, Syria, 1–6.

[71] Zhang, M. Adaptive protocol for entity authentication and key agreement in mobile networks, in *ACM International Conference on Information Security and Cryptology (ICISC' 03)*, Seoul, Korea, 88–95.

[72] Herzberg, A. *et al.* On travelling incognito, in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1994)*, Santa Cruz, CA, USA, 205–211.

[73] Samfat, D. *et al.* Anonymity and untraceability in mobile networks, in *ACM International Conference on Mobile Computing and Networking (MobiCom' 95)*, Berkeley, California, USA, 26–36.

[74] Asokan, N. Anonymity in a mobile computing environment, in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1994)*, Santa Cruz, CA, 200–204.

[75] Lin, H. Y. & Harn, L. Authentication protocols for personal communication systems, *ACM SIGCOMM Computer Communication Review* **25**(4), 256–261, 1995.

[76] Horn, G. & Preneel, B. Authentication and payment in future mobile systems, in *European Symposium on Research in Computer Security (ESORICS' 98)*, Louvain-la-Neuve, Belgium, 277–293.

[77] Park, J. *et al.* Wireless authentication protocol preserving user anonymity, in *Symposium on Cryptography and Information Security (SCIS 2001)*, Oiso, Japan, 159–164.

[78] Juang, W. S. & Wu, J. L. Efficient 3GPP authentication and key agreement with robust user privacy protection, in *IEEE Wireless Communications and Networking Conference (WCNC 2007)*, Kowloon, Hong Kong, 2720–2725.

[79] Forsberg, D. *et al.* Enhancing security and privacy in 3GPP E-UTRAN radio interface, in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, 1–5.

[80] Molva, R. *et al.* KryptoKnight authentication and key distribution system, in *European Symposium on Research in Computer Security (ESORICS 92)*, Toulouse, France, 155–174.

[81] He, Q. *et al.* The quest for personal control over mobile location privacy, *IEEE Communications Magazine* **42**(5), 130–136, 2004.

[82] Køien, G. & Oleshchuk, V. Location privacy for cellular systems; Analysis and solution, in *International Workshop on Privacy Enhancing Technologies (PET 2005)*, Cavtat, Croatia, 40–58.

[83] Godor, G. *et al.* Novel authentication algorithm of future networks, in *IEEE International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL' 06)*, Mauritius, 80–80.

[84] Gódor, G. & Imre, S. Novel authentication algorithm–public key based cryptography in mobile phone systems, *International Journal of Computer Science and Network Security* **6**(2B), 126–134, 2006.

[85] Yang, G. *et al.* Anonymous and authenticated key exchange for roaming networks, *IEEE Transactions on Wireless Communications* **6**(9), 3461–3472, 2007.

[86] Li, X. & Wang, Y. Security enhanced authentication and key agreement protocol for LTE/SAE network, in *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM' 11)*, Wuhan, China, 1–4.

[87] Varadharajan, V. & Mu, Y. Preserving privacy in mobile communications: A hybrid method, in *IEEE International Conference on Personal Wireless Communications (ICPWC 1997)*, Mumbai, India, 532–536.

[88] Al-Fayoumi, M. *et al.* A new hybrid approach of symmetric/asymmetric authentication protocol for future mobile networks, in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007)*, New York, USA, 29–29.

[89] Zhu, J. & Ma, J. A new authentication scheme with anonymity for wireless environments, *IEEE Transactions on Consumer Electronics* **50**(1), 231–235, 2004.

[90] Jiang, Y. *et al.* Mutual authentication and key exchange protocols for roaming services in wireless mobile networks, *IEEE Transactions on Wireless Communications* **5**(9), 2569–2577, 2006.

[91] Sattarzadeh, B. *et al.* Improved user identity confidentiality for UMTS mobile networks, in *IEEE European Conference on Universal Multiservice Networks (ECUMN '07)*, Toulouse, France, 401–409.

[92] Tang, C. & Wu, D. O. Mobile privacy in wireless networks-revisited, *IEEE Transactions on Wireless Communications* **7**(3), 1035–1042, 2008.

[93] Pereniguez, F. *et al.* Privacy-enhanced fast re-authentication for EAP-based next generation network, *Computer Communications* **33**(14), 1682–1694, 2010.

[94] Yang, G. *et al.* Universal authentication protocols for anonymous wireless communications, *IEEE Transactions on Wireless Communications* **9**(1), 168–174, 2010.

[95] He, D. *et al.* A strong user authentication scheme with smart cards for wireless communications, *Computer Communications* **34**(3), 367–374, 2011.

[96] He, D. *et al.* Design and validation of an efficient authentication scheme with anonymity for roaming service in global mobility networks, *Wireless Personal Communications* **61**(2), 465–476, 2011.

[97] Zhou, T. & Xu, J. Provable secure authentication protocol with anonymity for roaming service in global mobility networks, *Computer Networks* **55**(1), 205–213, 2011.

[98] Chen, C. *et al.* Lightweight and provably secure user authentication with anonymity for the global mobility network, *International Journal of Communication Systems* **24**(3), 347–362, 2011.

[99] He, D. *et al.* Privacy-preserving universal authentication protocol for wireless communications, *IEEE Transactions on Wireless Communications* **10**(2), 431–436, 2011.

[100] Feng, T. *et al.* Anonymous identity authentication scheme in wireless roaming communication, in *IEEE International Conference on Computing Technology and Information Management (ICCM 2012),* Berlin, Germany, 124–129.

[101] Liu, H. & Liang, M. Privacy preserving registration protocol for mobile network, *International Journal of Communication Systems.* `http://onlinelibrary.wiley.com/doi/10.1002/dac.2426/full`, 2012.

[102] Jiang, Q. *et al.* An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks, *Wireless Personal Communications* **68**(4), 1–15, 2012.

[103] Schneier, B. & Sutherland, P. *Applied cryptography: Protocols, algorithms, and source code in C,* John Wiley & Sons, Inc., United States, 1995.

[104] Edney, J. & Arbaugh, W. *Real 802.11 security: Wi-Fi protected access and 802.11 i,* Addison-Wesley Professional, United States, 2004.

[105] Hardjono, T. & Dondeti, L. R. *Security in wireless LANs and MANs,* Artech House, USA, 2005.

[106] Choudhury, H. *et al.* Desirable features of an identity privacy ensuring solution for UMTS, in *National Workshop on Network Security (NWNS 2013),* Tezpur, India, 143–157.

[107] Wong, D. S. Security analysis of two anonymous authentication protocols for distributed wireless networks, in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom 2005),* Kauai Island, HI, USA, 284–288.

[108] Lee, C. C. *et al.* Security enhancement on a new authentication scheme with anonymity for wireless environments, *IEEE Transactions on Industrial Electronics* **53**(5), 1683–1687, 2006.

[109] Wu, C. C. *et al.* A secure authentication scheme with anonymity for wireless communications, *IEEE Communications Letters* **12**(10), 722–723, 2008.

[110] Mun, H. *et al.* Enhanced secure anonymous authentication scheme for roaming service in global mobility networks, *Mathematical and Computer Modelling* **55**(1), 214–222, 2012.

[111] Chang, C. C. *et al.* Enhanced authentication scheme with anonymity for roaming service in global mobility networks, *Computer Communications* **32**(4), 611–618, 2009.

[112] Youn, T. *et al.* Weaknesses in an anonymous authentication scheme for roaming service in global mobility networks, *IEEE Communications Letters* **13**(7), 471–473, 2009.

[113] Zeng, P. *et al.* On the anonymity of some authentication schemes for wireless communications, *IEEE Communications Letters* **13**(3), 170–171, 2009.

Appendix A

HPLMN IP Services in 3GPP-WLAN

In this appendix, we present a simplified version of the roaming security architecture for access to IP services provided by the HPLMN in 3GPP-WLAN. The functionality of the components used in this architecture are same as the components used in the roaming architecture for access to IP services provided by the VPLMN in 3GPP-WLAN (Chapter 5, Figure 5.1). Here, the difference is that the PDG, with which the UE establishes a tunnel, is located in the HPLMN instead of the VPLMN.

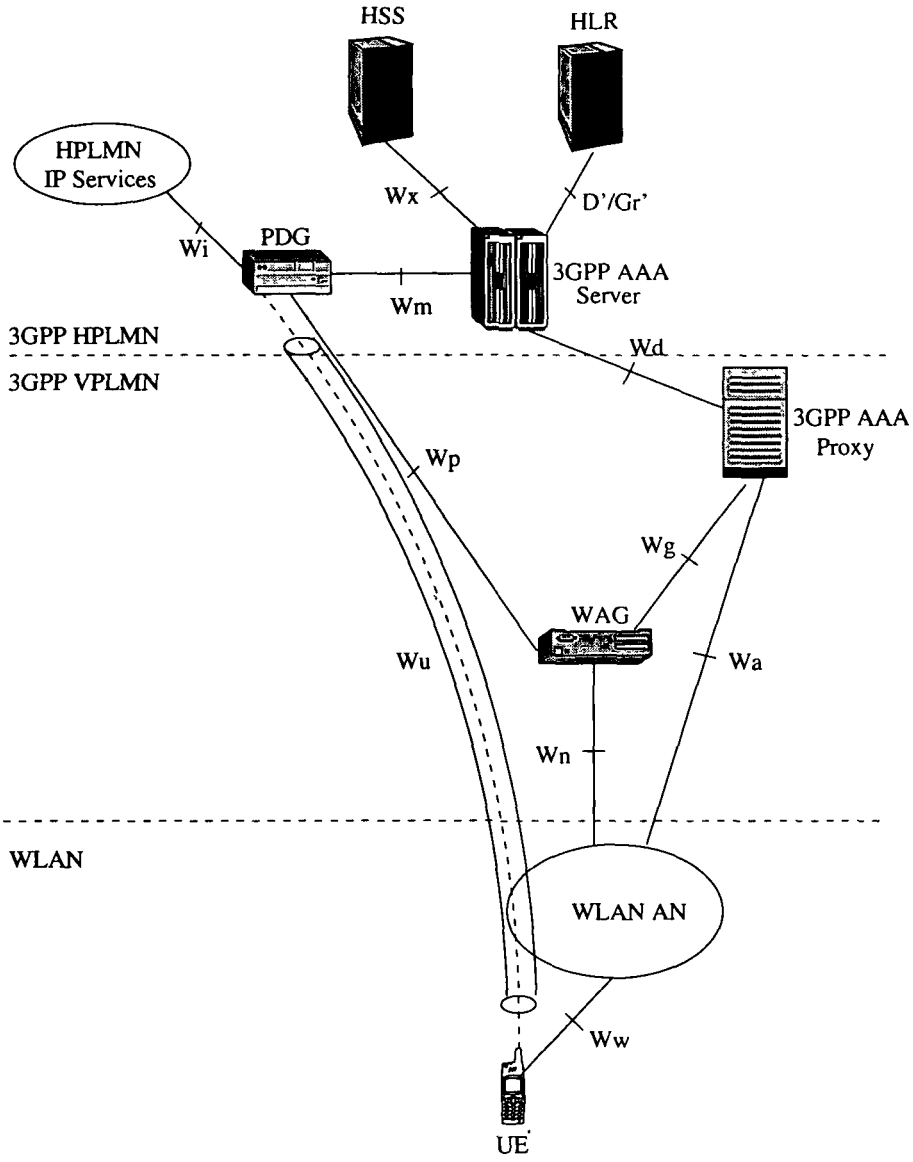

Figure A.1: Simplified roaming security architecture for access to IP services provided by HPLMN in 3GPP-WLAN.

Appendix B

HPLMN IP Services in Non3GPP-EPS

In this appendix, we present a simplified version of the roaming security architecture for access to IP services provided by the HPLMN in Non3GPP-EPS. The functionality of the components used in this architecture are same as the components used in the roaming architectures for access to IP services provided by the VPLMN in Non3GPP-EPS (Chapter 5, Figure 5.2). Here, the difference is that the PDN Gateway is located in the HPLMN instead of the VPLMN. And, an additional component called Serving Gateway (SGW) is used in the VPLMN. The role of SGW is to route user data packets.

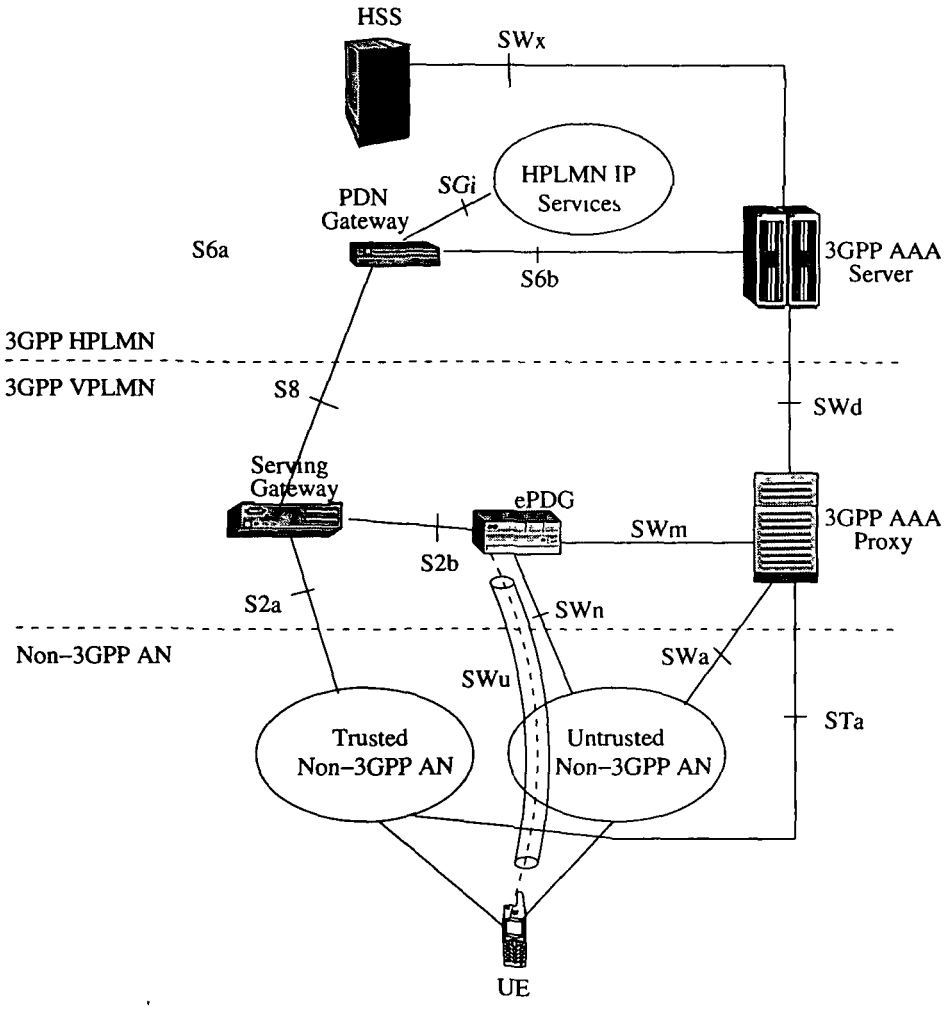

Figure B.1: Simplified roaming security architecture for access to IP services provided by HPLMN in Non3GPP-EPS.

Appendix C

Rules of AUTLOG

In this appendix, we present some of the rules of AUTLOG calculus [59] which are used in the formal analysis of Chapter 6, Section 6.2. The symbols that are used in defining these rules along with their usage are as follows (more details about these symbols can be found in [59]):

- P, Q, R represents agents, who communicate with each other.
- M represents messages which could be one of the following: names of agents, key components, computed messages, list of messages and derived keys.
- M_P represents a message in view of P or in other words a message that is localised towards P. Here, P does not necessarily understand M_P. For example, P receives a list including a cryptogram and a hash value: P *sees* $(\{X\}_K, h(M))$, where P cannot decrypt the ciphertext but knows M.
- X, X_i, Y, Y_i, Z represents messages which could be one of the following: names of agents, key components, computed messages, list of messages, derived keys and localised messages (M_P).
- K represents a public key scheme that consist of a public component K^+ and a private component K^-.

- K^{-1} represents an inverse key which is the corresponding component for public key schemes and equal to K in symmetric case.
- h denotes all hash functions including message authentication codes.
- σ denotes signatures without message recovery.
- $enc(K, M) = \{M\}_K$ denotes encryption of a message M with K and a signature with message recovery (i.e., cleartext M can be derived from $\{M\}_K$ under knowledge of the inverse key K^{-1}).
- F represents any function in $\{\sigma, h, enc\}$
- H is a one way function out of $\{\sigma, h\}$
- φ and ψ represents a formulae of the following kind:
  - $P \overset{K}{\leftrightarrow} Q$: K is a shared key/secret between P and Q.
  - $fresh(M)$: Message M has been created in the current protocol run.
  - $M \equiv N$: M is equivalent to N.
  - P *sees* M: Agent P was able to read M as a submessage of a received message.
  - P *said* M: Agent P has sent the message M and has been conscious of sending it at that time.
  - P *says* M: Agent P has sent the message M knowingly and recently.
  - P *has* M: P knows message M and can use it for further computations.
  - P *recognises* M: Either P has a reason to believe that M is not a random string but willingly constructed or that M is a random string already known to P.
  - P *controls* φ: P is able to decide whether φ is correct or not.
  - P *believes* φ: P has strong evidence that φ is correct.
  - $\neg\varphi$: Negation.

– $\varphi \wedge \psi$: Conjunction.

Following are some of the rules of AUTLOG calculus that are used in the formal analysis of E2EUIC (the characters written in bold, preceding each of the rules ($\mathbf{MP}, \mathbf{K}, \mathbf{J}, \mathbf{H1}$, etc.) are the names of the rules as stated in [59]):

MP: If φ and $(\varphi \rightarrow \psi)$ then ψ

K: P *believes* $\varphi \wedge P$ *believes*$(\varphi \rightarrow \psi) \rightarrow P$ *believes* ψ

J: $(P$ *controls* $\varphi \wedge P$ *believes* $\varphi) \rightarrow \varphi$

H1: P *sees* $X \rightarrow P$ *has* X

H2: P *has* $X_1 \wedge ... \wedge P$ *has* $X_n \rightarrow P$ *has* $(X_1, ..., X_n)$

H3: P *has* $X \rightarrow P$ *has* $F(X)$

F1: $freshX_i \rightarrow fresh((X_1, ..., X_n))$

SE2: P *sees* $enc(K, X) \wedge P$ *has* $K^{-1} \rightarrow P$ *sees* X

NV: P *said* $X \rightarrow fresh(X) \rightarrow P$ *says* X

A1: R *sees* $F(K, X) \wedge P \overset{K}{\leftrightarrow} Q \wedge \neg P$ *said* $F(K, X) \rightarrow Q$ *said* (K, X)

C: P *sees* $M \wedge M_P \equiv Y \rightarrow P$ *believes* P *sees* Y

C1: P *recognizes* $X_i \rightarrow (X_1, ..., X_n)_P \equiv ((X_1)_P, ..., (X_n)_P)$

C3: P *has* $M \rightarrow H(M)_P \equiv H(M_P)$

E2: $X \equiv Y \wedge Y \equiv Z \rightarrow X \equiv Z$

E3: $X \equiv Y \rightarrow F(X) \equiv F(Y)$

E4: $X_1 \equiv Y_1 \wedge ... \wedge X_n \equiv Y_n \rightarrow (X_1, ...X_n) \equiv (Y_1, ..., Y_n)$