NTNU – Trondheim
Norwegian University of
Science and Technology

Is the Session Mix-up Attack on the UMTS/LTE AKA Protocol Practical?

Ashim Bhusal

Master of Telematics - Communication Networks and Networked Services (2
Submission date: Januar 2013
Supervisor: Stig Frode Mjølsnes, ITEM

Norwegian University of Science and Technology
Department of Telematics

Is the Session Mix-up Attack on the UMTS/LTE AKA Protocol Practical?

TTM4905
Report
Master Thesis

Project Member :
Ashim Bhusal (bhusal@stud.ntnu.no)

Supervisor : Joe-Kay Tsay, NTNU ITEM
(joe.k.tsay@item.ntnu.no)
Responsible Professor : Professor Stig Frode
Mjølsnes, NTNU ITEM (stig.mjolsnes@item.ntnu.no)

NTNU
Det skapende universitet

Norwegian University of Science and Technology
Autumn 2012

Abstract

The real time implementation of theoretically proven facts is a challenging task. Not all the theoretically shown results are valid in real practice. The standards defining the policies leave lot of space for specific implementation, hence a vulnerability seen in case of some protocols defined by some specific standards may not exist in practical. The flaw may be totally or partially masked by other protocols running in conjunction with the so called vulnerable protocol. Also there is possibility that those unspecified steps in the standards are implemented by the operators. This work extends the project title: *Validation of attacks on the UMTS/LTE AKA protocol*, where the vulnerability found by Stig Frode Mjølsnes and Joe-Kay Tsay , presented in several seminars and explained in *A Vulnerability in the UMTS and LTE Authentication and Key Agreement Protocols* was theoretically proven.

The previous works had analysed the mechanisms of Authentication and Key Agreement (AKA) protocol as defined in the 3^{rd} Generation Partnership Project (3GPP) specifications. This thesis work tries to solve the questions raised by the findings of those prior works regarding AKA protocol. Here, the theoretically shown vulnerability is analysed based on information about other protocols like Internet Protocol (IP) Security (IPsec), Diameter, Signalling System 7 (SS7) and Mobile Application Part (MAP) Security (MAPsec) running along with AKA protocol. Several people from different operators were contacted so as to find the facts about which protocols are used and how these protocols are implemented.

Here the session related parameters, Session-Id in case of Long Term Evolution (LTE) and Invoke-Id and Transaction Id in case of Universal Mobile Telecommunication System (UMTS) are discussed to show if they are capable of withstanding the session mix-up vulnerability. The sessions of a user or service operations are uniquely identified by these identifiers. But the number of bits used by the identifiers and their formats have left the place to suspect if session mix-up attack is still attainable. The conclusions derived here are not complimented by the information from real implementation scenario, so, as far as it

is possible to obtain information about real implementation mechanisms from some operators, this work can be further extended to compare the strength of Session Mix-up attack against the mechanisms implemented by the operators.

Preface

This report documents the Master's thesis work entitled "Is the Session Mix-up Attack on the UMTS/LTE AKA Protocol Practical? ", which was carried out under the Faculty of Information Technology, Mathematics and Electrical Engineering, Department of Telematics(ITEM), Norwegian University of Science and Technology (NTNU), during Autumn 2012. This thesis work holds the weightage of 30 study credits and serves for the partial fulfilment of Masters of Science(MSc) degree in Telematics at NTNU. The objective of this work is to find whether theoretically proven session mix-up attack is attainable in real world scenario or not.

I would like to extend my sincere gratitude to Professor Stig Frode Mjølsnes and supervisor Joe-Kay Tsay at Department of Telematics, NTNU, for their regular and valuable guidance, feedbacks and suggestions. I would like to thank Professor Dr. Do van Thanh of NTNU/Telenor for the document he provided which worked to boost up the initial phase of this work. My thanks goes to Sebastien Decugis for his continuous support with the use and debugging of the software $freeDiameter$.

I wish to appreciate the help provided by my friends, relatives and family members throughout the period of this thesis work. As a consequence, this work has been successfully accomplished and the desired results have been obtained.

Ashim Bhusal,
Trondheim, January 2013.

Contents

List of Figures

List of Tables

Acronyms

2G Second Generation. 3, 13

3G Third Generation. 3, 9, 13

3GPP 3^{rd} Generation Partnership Project. i, 1, 3, 7–9, 14, 16, 17, 19, 23–27, 29, 40, 41, 43, 47, 48, 51, 52

4G Fourth Generation. 2, 7

AAA Authentication, Authorisation, and Accounting. 20–22

AH Authentication Header. 18, 19, 34

AIA Authentication Information Answer. 25, 26, 28, 29, 40, 43

AIR Authentication Information Request. 25, 26, 28, 29, 40, 43

AKA Authentication and Key Agreement. i, vii, 1–3, 5, 7–14, 19, 24, 26, 28, 45–47, 51, 52

AS Access Stratum. 10

AuC Authentication Centre. 3, 8, 9, 14, 31

AV Authentication Vector. 8–12, 14, 19, 26, 27, 29, 40, 42, 43, 52

AVP Attribute Value Pair. 22, 29, 30, 43, 47

BS Base Station. 7

CN Core Network. 13, 14, 17

DNS Domain Name System. 27

DTLS Datagram Transport Layer Security. 23

eNodeB Evolved NodeB. 7, 8

EPS Evolved Packet System. 2, 9, 11, 16, 51

ESP Encapsulating Security Payload. 18, 19, 34, 47

ETSI The European Telecommunications Standards Institute. 1, 25, 27, 51

EUTRAN Evolved UTRAN. 13, 29

FQDN Fully Qualified Domain Name. 22, 29

GERAN GSM/EDGE Radio Access Network. 29, 30

GSM Global System for Mobile communication. 2, 3, 7, 11, 13, 16, 53

HLR Home Location Register. 8, 9, 14, 16

HN Home Network. 1, 7–10, 12, 31, 41, 45, 47, 48

HSS Home Subscriber Server. 7–9, 12, 13, 16, 19, 25–28, 31, 40–43

IANA Internet Assigned Numbers Authority. 27, 37

IETF The Internet Engineering Task Force. 1, 3, 14, 17, 21, 23, 26, 27, 51

IKE Internet Key Exchange. 15, 18

IMSI International Mobile Subscription Identity. 10, 25

IP Internet Protocol. i, xii, 1–3, 5, 13, 14, 17–19, 23, 25, 27, 31–34, 36, 42, 46, 51

IPsec IP Security. i, 1, 3, 5, 8, 14, 17–19, 23, 31–34, 36, 37, 41–43, 45, 47, 51

ITU-T International Telecommunication Union-Telecommunication. 1, 3, 14, 16, 47, 51

KAC Key Administration Centre. 15

LTE Long Term Evolution. i, 2, 3, 5, 7–9, 11, 13, 14, 17, 23, 24, 28, 29, 31, 41–43, 45, 46, 51–53

MAC Message Authentication Code. 15

MAP Mobile Application Part. i, xii, 1, 5, 14–16, 25, 45, 47, 48, 51, 52

MAPsec MAP Security. i, 1, 3, 5, 14–16, 42, 45, 48, 51, 53

MME Mobility Management Entity. 8–10, 12, 13, 16, 19, 24–28, 31, 40–42

NAS Network Access Server. 23

nAS non Access Stratum. 10

NDS Network Domain Security. 5, 13, 14, 18, 19, 23, 28, 46

NE Network Element. 15, 16

NTNU Norwegian University of Science and Technology. 3

OS Operating System. 32, 35

PC Personal Computer. 32

PLMN Public Land Mobile Network. 15, 25

PTI Procedure transaction identity. 17, 48

RADIUS Remote Authentication Dial In User Service. 20

RFC Request for Comment. 3, 7, 17, 21, 26, 40, 51

S-GW Serving Gateway. 7, 8

SA Security Association. 15, 16, 18

SCTP Stream Control Transmission Protocol. 23, 27

SGSN Serving GPRS Support Node. 14, 16, 25, 26

SN Serving Network. 1, 7–12, 16, 26, 28, 29, 31, 41, 45, 47, 48

SPI Security Parameter Index. 18, 34

SS7 Signalling System 7. i, 5, 13, 14, 25, 42, 45, 47, 51

TCAP Transaction Capabilities Application Part. 5, 14, 47

TCP Transmission Control Protocol. 23

TLS Transport Layer Security. 23, 36, 37, 42

TS Technical Specification. 7, 17, 25, 27

UE User Equipment. 1, 7–10, 41

UMTS Universal Mobile Telecommunication System. i, 2, 3, 5, 7–9, 11, 13, 14, 16, 19, 42, 45, 47, 51–53

USIM User Subscriber Identity Module. 7–10, 12, 41

UTRAN UMTS Terrestrial Radio Access Network. 13, 29, 30

VLR Visitors Location Register. 8, 9, 14, 16

Chapter 1

Introduction

1.1 Problem Description

There are some protocols running between three entities: User Equipment (UE), Serving Network (SN) and Home Network (HN) before a requested service is granted to a user(mobile devices). The Authentication and Key Agreement (AKA) Protocol, where the users are authenticated to respective HN and the keys for subsequent communication are derived and exchanged between these entities, is one of them. The AKA protocol does not specify any mechanisms for session management. Exploring this, a session mix up attack on this protocol has been found. In this attack, an attacker has a control over all the traffic going in and out of Serving network and the attacker itself is one of the participating mobile user. The attacker uses a session of AKA established for a genuine user to authenticate itself and thus has ability to carry on subsequent communication on behalf of a genuine user. Although unspecified in AKA, it is believed that there are some specific session management or other security protocols running underneath of AKA (probably IP Security (IPsec), MAP Security (MAPsec), Radius/Diameter may be some of them) which may prevent this attack. These protocol may vary depending upon the owner of SN. If the SN belongs to the HN of the user, the protocol may be vendor specific or specific to service provider whereas if the SN is acquired by other HN than the one to which user is attached, some agreed protocol or some standard protocols are defined and used.

The 3^{rd} Generation Partnership Project (3GPP), The Internet Engineering Task Force (IETF), International Telecommunication Union-Telecommunication (ITU-T), The European Telecommunications Standards Institute (ETSI) and other bodies define and recommend the standards to be used. The attack may not be feasible in case the standards are implemented as recommended. But in non roaming case where application of these standards depend upon the operator the attack may be possible. The aim of this work is to find the protocols running below AKA for inter and intra domain communication. In case of inter domain communication, conclusions will be derived based on study of several specifications, relevant documents and recommendations from some operators. These documents and specifications are discussed in chapter 2. In the case of intra domain communication, the conclusions are highly based on the information provided by people from different service providers. The liveliness of Session Mix up Attack in real world is than tested depending upon the mechanisms mandated or recommended in relevant specifications and based on the information provided by several service provider. The other objective of the work is to construct a software simulation based on analysis of collected information, showing if the proposed session mix-up attack can be attainable in practice. Shortly, the main tasks expected in this thesis work are listed below:

- Gathering of information from people related to telecommunication operators from different location
- Finding the recommended standards from several specifications and recommendations.
- Analysing the gathered information and study the possibility of session mix-up attack based on those information.
- Construct a scenario (software based model) to implement the recent practice based on specifications and information from several people.

1.2 Motivation

The use of Long Term Evolution (LTE) based wireless communication system, also termed as Evolved Packet System (EPS) and Fourth Generation (4G) system is increasing. The LTE system being all Internet Protocol (IP) based system has been taken as a solution to shortcomings of legacy systems, Global System for Mobile communication (GSM) and Universal Mobile Telecommunication System (UMTS). There are various aspects in which this new system is superior to its predecessors. One of the aspect in which our study will focus is a security aspect. It is a known fact that there were no network domain security in case of Global System for Mobile communication (GSM) and also mutual authentication of UMTS were not able to protect some attacks when working in GSM environment. Irrespective of all the previously found attacks on GSM and UMTS, a protocol level session mix-up attack has been detected by Mjølsnes and Tsay. This attack aims towards Authentication and Key Agreement (AKA) protocol of UMTS and LTE. Exploiting some unspecified steps in AKA protocol the attack has theoretically been proven. The attack has been presented in several workshops and seminars [1, 2, 3] and was theoretically shown in a project carried out under Department of Telematics, Norwegian University of Science and Technology (NTNU). But the questions regarding the possibility of this session mix-up vulnerability in real world scenario were still to be solved and left as an extension to the project.

This thesis work will try to solve those unsolved questions by acquiring information from different operators, studying of several documents, specifications, standards and recommendations and developing (implementing) a real scenario based on acquired information and study. Here some other protocols below AKA are studied to find if the attack is attainable even after implementation of those protocols. A brief description of the task is discussed below.

1.3 Research Methodology

The method adopted for this thesis are theoretical studies, acquisition of data and information and experimental set ups. The theoretical studies are mainly based on the 3GPP specifications, IETF Request for Comment (RFC)s, ITU-T recommendations and other relevant articles and books. The related specifications, in many cases, do not provide complete information regarding implementation and has left the proper techniques of implementation to be dependent on the service providers. So, the information from people of different operators is required to compliment the information drawn from specifications and documents. In order to find the actual implementation technique, people working in different telecommunication operator were contacted. The expected information from those people are the answers to the following questionnaire:

- What are the communication and security layers below AKA in case of IP based network (LTE), UMTS and GSM?

- What mechanisms are used by operators underneath the protocol stack to protect the sessions between different calls?

- What communication security techniques are used between the serving and home network in the roaming situation, and within each mobile operator domain communication with the Authentication Centre (AuC)?

- Which communication security techniques are used to secure Second Generation (2G)/Third Generation (3G) user roaming in 3G/2G environment or vice-versa?

- Which of security techniques like: IPsec, MAPsec, Radius/Diameter etc. are chosen/used by operator?

The table A.1 in Appendix A contains the list of people from different countries who were contacted.

Unfortunately, the sought information was not obtained. The only information obtained was not enough to derive a conclusion. This vacuum compelled us to model a system based only on the information derived from several specifications. After developing a model, different tools to implement this model were searched. Owing to the time taken to understand the tool, unanticipated limitation of the selected tool to perform the required and expected implementation, the inability of alternate tools to cope those limitations and the time boundary to carry out this thesis work, the last objective, i.e. to construct a simulation tool, of the thesis work could not be achieved as desired. Although the desired construction of software simulation was not successful, the explanation of the protocols are assisted by the experimental set ups as described in chapter 3.

1.4 Scope of the work

This thesis work is intended to check the possibility of Session Mix-up attack in real scenario. The analysis is based on the information provided by the several specifications. This work do not perform any computational and cryptographic analysis, rather the decisions are based on protocol level analysis. The analysis of protocols are assisted by minimum implementation of them. The required software simulation are not coded and developed on our own effort but pre developed tools were searched. The derived conclusions are not tested in the real world implementation. This work attempts to find the relevant information from specialised people and standards and use the information to construct a model. Thus, developed model is tried to implement and test by use of some pre built open source tools.

1.5 Review of Report

The report contains five main chapters, references and the appendices. The summary of each chapter is provided below:

- Chapter 1: Introduction
  This chapter explains the problem in simple words and presents the motivation to carry out this work. Finally this chapter describes the structure of report and summarises the overall report.

- Chapter 2: Background Theory
  The theories which provide basic understanding of the terms, protocols and their working relevant to this thesis work are explained in this chapter.

This chapter elaborates the background literature based on several specifications, documents, books and various sources. Here, the UMTS/LTE AKA protocol, the Session Mix-up attack, Network Domain Security (NDS), LTE based protocols like NDS-IP (IPsec) and Diameter and UMTS related protocols like Mobile Application Part (MAP) of Signalling System 7 (SS7), Transaction Capabilities Application Part (TCAP) and MAPsec are explained. These explanations are the base for the modelling, implementation and discussion part of this thesis.

- Chapter 3: Lab Experiments
  This chapter explains about the tools used to obtain the implementation phase. Here the protocols Diameter and IPsec are explained with help of simple implementation. The latter part of this chapter explains about the model developed, attempts to implement the model, problems faced and some suggestions for further work.

- Chapter 4: Discussion and Evaluation of the work
  This chapter presents the discussions related to session mix-up attack and its possibility based on the theories provided in Chapter 2. The AKA protocol stacks i.e. $S6a$ application for LTE and MAP application for UMTS are illustrated. With the help of session specific parameters the findings are discussed.

- Chapter 5: Conclusions and Further Extensions
  The concluding remarks of the overall thesis work, the limitations and the future works to mitigate those limitations are discussed in this chapter.

Chapter 2

Background Theories

The wireless communication system has passed through three major phases and reached the fourth phase, 4G, also popular with name LTE. The major portion of this study covers the description and analysis based on the current phase LTE and its predecessor UMTS. The working of UMTS and LTE, where our study is concerned is pretty similar. So, some of the explanations details only one of these system, in most cases it is LTE. The LTE being the latest system is supposed to surpass the shortcomings of older and hence is given more preference in this work. Here, in depth study of the protocols carrying and securing AKA messages between HN and SN is provided. In this chapter, first a view to LTE components and interfaces between components are presented. Then, theories and various processes based on several specifications, RFC's, papers, books and recommendations are elaborated in order to clarify several related terms and overall objectives of the work.

2.1 UMTS/LTE Components and Interfaces

The basic components of LTE system and the interfaces between these components is shown in figure 2.1. The components and interfaces for UMTS system are defined in 3GPP specification 3GPP Technical Specification (TS) 09.02 [4]. The different components of the figure are obtained from several sources through internet. The figure comprises of three parts, i) the User side, ii) Serving Network side and iii) Home Network side. The user side consists of UE and User Subscriber Identity Module (USIM), which are in the hands of users. The Evolved NodeB (eNodeB) is equivalent to Base Station (BS) of GSM and Serving Gateway (S-GW) serves the purpose of routing user data, handling of inter-eNodeB handover and act as anchor for mobility. The third part HN contains Home Subscriber Server (HSS) where all the subscription data of users reside. Any service, either voice or data is the result of interaction of the entities of these three parts. The link between UE/USIM and eNodeB (the blue line) is termed as Uu interface by 3GPP. The link from eNodeB to Mobility Management Entity (MME) and S-GW, denoted by green line in figure are $S1 - MME$ and $S1 - U$ interfaces respectively. The communication in the interfaces $S1 - MME$ and $S1 - U$ is protected by IPsec. The "Authentication and Key Agreement" portion of AKA runs from USIM to MME as shown in figure. The interface $S11$ between MME and S-GW is not discussed here. Finally the link marked by red line in figure between MME and HSS is the $S6a$ interface. The AKA messages between MME and HSS are transported through this interface. The diameter protocol is implemented in this interface. The authentication data request for an user is sent to HSS by MME and the Authentication Vector (AV)s generated for respective users are sent to MME by HSS using diameter protocol. The details on $S6a$ interface and the Diameter protocol is explained in subsequent sections. Also the diameter protocol is more clarified by its implementation in further chapters. The communication between MME and HSS as shown inside dotted box "Implementation part" was supposed to be modelled and implemented.

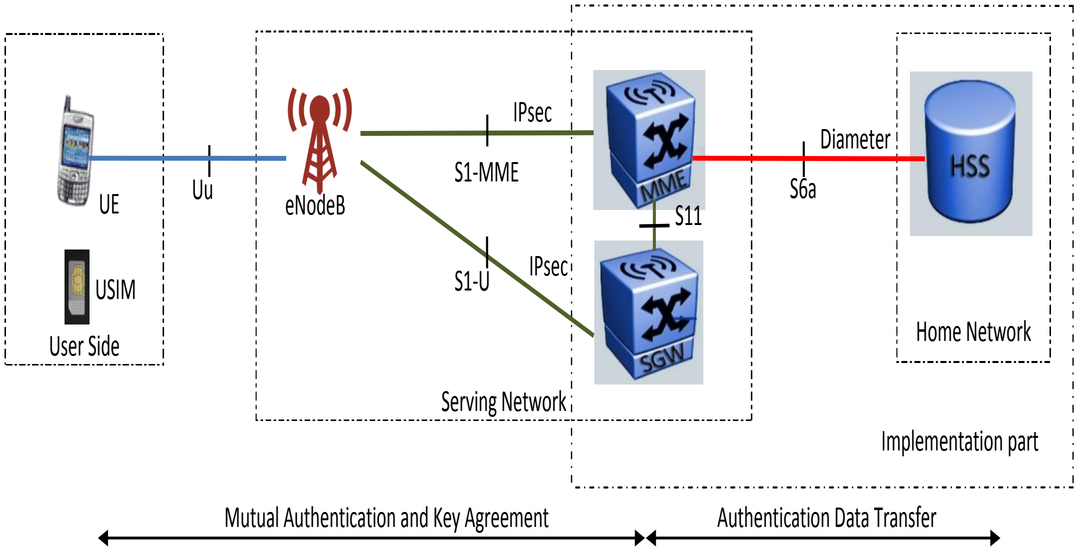


Figure 2.1: LTE Security Architecture with Interfaces.

2.2 The AKA protocol

The Authentication and Key Agreement protocol is a three parties handshake between User(UE/USIM), SN(MME or Visitors Location Register (VLR)) and HN(Home Location Register (HLR) or HSS/AuC). In this protocol the user requesting for service and the network providing the service are authenticated to each other and keys for subsequent communication are derived. The AKA process for UMTS and LTE are almost similar except that the participating entities and some exchanged parameters are termed differently. In order to generalise

AKA process for both UMTS and LTE we have used some terms to represent entities of both system in a common way. The term SN denotes both VLR of UMTS and MME of LTE; the term HN represents HLR/AuC of UMTS and HSS of LTE and $User$ is used to denote UE/USIM in our description. The above mentioned related terms may be interchangeably used in further part of this report.
The AKA protocol for the case of 3G UMTS is detailed in specification [5] and specification [6] and book [7] describes the procedure for case of LTE/EPS. The term Evolved Packet System (EPS) represent LTE system in case of 3GPP wireless communication. So these terms may be interchangeably used in further discussions of this report. The detailed explanation of the protocol is also presented in [8]. This thesis work is continuation to the work carried on [8]. The summary of AKA process follows as: the SN fetches user specific parameters from HN to which the user has subscription. Some of these parameters are transferred to $User$ where further elements are derived and transported back again to MME. The $User$ part derives further elements only after validating the freshness of the parameters and successful network authentication. Now the SN makes the authentication decision based on the comparison of the elements received from HN and $User$.
The protocol completes in two phases: 1) Distribution of AVs and 2) Authentication and key Agreement. Several terms related to AKA and their meaning listed in table 2.1 will be used in further explanation of AKA.

Terms	Name	Purpose
$IMSI$	Permanent Subscriber Identity	identify a user on the radio path
K_0	secret key shared between the $USIM$ and the AuC	used to derive other Authentication parameters
$RAND$ SQN	Random Number Generated in AuC Sequence number	used to derive other parameters and sent to USIM as an element of AV to ensure the freshness of the vectors
$(X)MAC$	(Expected) Message Authentication Code	MAC and $XMAC$ are compared in USIM

Continued on next page ...

Table 2.1 – continued from previous page

Terms	Name	Purpose
$(X)RES$	(Expected) user response	RES and $XRES$ are compared in MME to authenticate the user
$AUTN$	Authentication Token	one of AV parameter; SQN_H and MAC are extracted from $AUTN$ in USIM
CK	Cipher Key	generated in UE; used to encrypt the messages
IK	Integrity Key	generated in UE; used for integrity protection
K_{ASME}	Key Access Security Management Entity	used in generation of Access Stratum (AS) and non Access Stratum (nAS) ciphering and integrity keys
AK	Anonymity Key	conceals the SQN but is optional
SN_{id}	Serving Network Identity	used to compute K_{ASME} and to authenticate serving network

Table 2.1: Terms used in AKA process.

Based on all above mentioned specifications, book and report, the AKA is pictured in figure 2.2 and the process is summarised as below:

- **Phase 1: Distribution of AV**

  1. SN requests user id from $User$ (this step is not always performed and not shown in figure).
  2. $User$ sends International Mobile Subscription Identity (IMSI) as user id response to SN.
  3. SN sends authentication data request to HN. Along with this message, identities of both user i.e. IMSI and serving network i.e SN_{id} are sent. SN_{id} is required to derive K_{ASME}.
  4. HN generates AVs and sends it to SN along with authentication data response. The generation of AVs is not described here.

Thus received AVs are stored in SN and used for the further authentication of the $User$. Here completes the first phase of AKA. The elements of AVs differ for GSM, UMTS and EPS. The table 2.2 lists the elements comprising AV in all three cases.

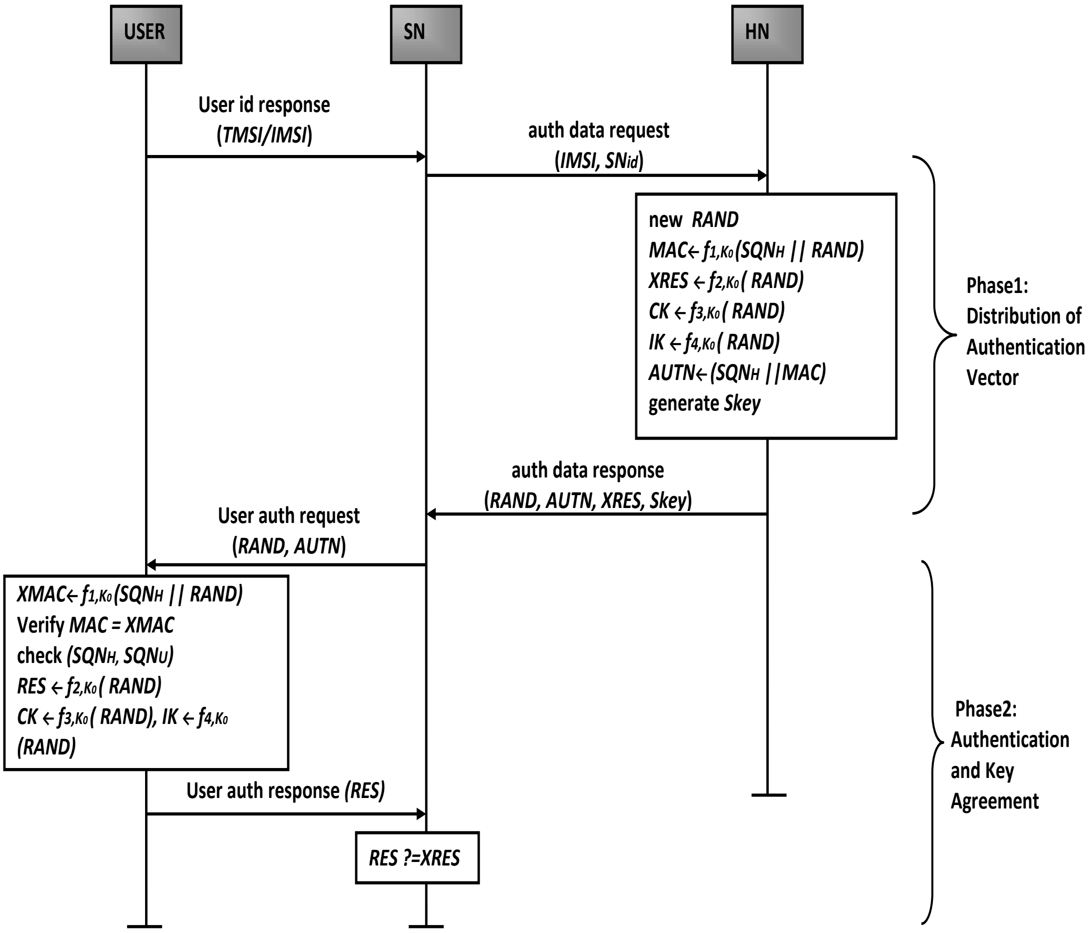


Figure 2.2:
Authentication and Key Agreement Protocol [8].

Here,
$Skey \leftarrow CK\|IK$ for UMTS and
$Skey := K_{ASME} \leftarrow KDF(SQN_H\|CK\|IK\|SNid)$ for LTE.

	GSM	UMTS	EPS
Elements of AV	authentication triplets 1. RAND 2. XRES 3. Kc	authentication quintuplets 1. RAND 2. AUTN 3. XRES 4. CK 5. IK	authentication quadruplets 1. RAND 2. AUTN 3. XRES 4. K$_{ASME}$

Table 2.2: Elements of AV for GSM, UMTS and EPS.

- **Phase 2: Authentication and Key Agreement**

  1. Now SN sends two elements ($RAND$ and $AUTN$) of AV to $User$ along with authentication request message.
  2. In $User$ side, $XMAC$ and RES are computed. Thus computed $XMAC$ is verified with the one of HSS sent by MME (contained in $AUTN$). Another verification also takes place here. That is verification of SQN_H and SQN_U. Only if both verifications pass, $User$ generates RES and sends it to SN as user authentication response.
  3. Now SN compares the RES from $User$ and $XRES$ from HN. User authentication is successful if both are equal. Thus authenticated users can only participate in further communication and serving network serves for the communication procedure.

There are several functions used to generate or compute the elements related to AKA. These functions and processes are already detailed in report [8]. The standard [5, p.23] defines these functions and their use for computation of AKA related values. In [5, p.22-25] generation of AVs and authentication function in USIM are explained with help of figures.

2.3 The Session Mix-up Attack

In the session mix-up attack two concurrent sessions of AKA in SN for two different users are swapped by the attacker. The one of the user of which session is swapped is attacker himself. Thus, after being authenticated in a session of an honest user the attacker can now carry on subsequent communication steps on behalf of the genuine user. Figure 2.3 shows the session mix up attack. In the figure, A is the attacker, S is SN, H is HN, U is a $User$ and U' is a user under control of attacker. The details on session Mix-up attack is described in document [1]. Also a brief scenario of attack is explained in [8].

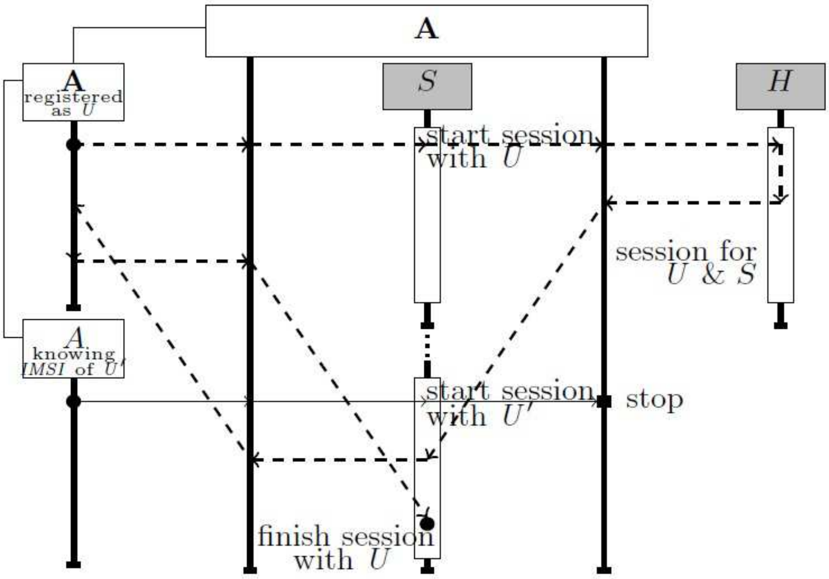


Figure 2.3: Session Mix-Up Attack [1].

As explained in [8], the attack is against $AuthenticationDataResponse$. If we look in the figure 2.2 of AKA, this message "$AuthenticationDataResponse$" is a message from HSS to MME. This message contains all the keys and parameters corresponding to the user. In the attack described above, this message generated by HSS for a genuine user is supposed to be utilised by user under control of attacker. Thus, the existence and the purpose of this attack is only fulfilled if the HSS really fails to detect the swap in session of two users. In order to check this, here we are trying to construct a real scenario of message exchange between HSS and MME. The NDS deals with the security of nodes of Core Network (CN) in case of UMTS Terrestrial Radio Access Network (UTRAN) or Evolved UTRAN (EUTRAN). Hence the next step of this work proceeds with the further study of NDS in case of UMTS and LTE.

2.4 Network Domain Security (NDS)

The GSM networks lack security in transfer of messages within the network. The specification [9] states that:*"The absence of security in SS7 networks is an identified security weakness in 2G systems"*. Also in the same specification it is mentioned that it is a goal of 3G systems to protect the CN signalling protocols. The 2G and non IP based UMTS run on SS7 signalling while LTE and IP based UMTS core use IP. So there is a need of security solutions for both SS7 and IP based protocols. The protection for SS7 based protocols are done in application layer and for IP based protocols protection is at network layer [10]. The protection of the signalling messages within network domain thus varies requiring separate procedures for UMTS (non-IP based core) and LTE (IP based core). The further discussions provided below show that in case of UMTS where SS7 signalling is used, protection of MAP termed as MAPsec is used, while for the case of LTE and IP based UMTS core native IP based protection termed as IPsec is applied. The 3GPP specifies both of these protocols with terms NDS-MAP and NDS-IP in several specifications. The overview to these protocols based on 3GPP specifications and other defining bodies like IETF for IPsec and ITU-T for MAP-SS7 is provided in sections below.

2.4.1 NDS-MAP and MAPsec

From the document [11] provided by Prof. Do Van Tanh, it is known that MAP is used to carry AVs from HLR/AuC to VLR in case of UMTS. In [12, p.74], it is further confirmed that MAP is the mobile specific part of SS7. In [7, p.40] it is mentioned that MAP protocol carries the control messages for UMTS AKA. The document [10] further states that: *"After careful analysis, it was found that one could only afford to protect the MAP protocol in this way". "..in this way"* in the statement hints towards protection in application layer. Thus, MAPsec [9] can be an ultimate choice for SS7 based networks which require protection at application layer. The major drawback of protection at application layer is that it requires modification of target protocol itself in cost of expenses and time and the procedure is to be repeated for every target protocols [10]. The statement: *"MAP is a crucial core network protocol that provides mobility management services and distributes the AV security data from the HLR/AuC to the VLR/Serving GPRS(Global Packet Radio Service) Support Node (SGSN)"* in [11], a report from Telenor[1] by the same author of [10] supports the adoption of MAP in case of UMTS AKA.

In SS7 signalling, the TCAP protocol, defined in ITU-T recommendations Q.771-Q.775 facilitates concurrent dialogs between same sub-systems [13]. The Transaction IDs are used to differentiate these concurrent dialogs. A specific operation invocation is identified by invoke-ID [14]. The detailed description and operation of TCAP, invoke-ID and Transaction IDs can be found in ITU-T recommendations Q.771-Q.775 ([15, 14, 16, 17, 18]). The ITU-T specifications Q.770-Q.849 deals with SS7 signalling. Although no 3GPP specifications were found mandating the use of MAP in case of UMTS, the information obtained from [11, 12, 7] suggest that MAP is used as CN protocol in case of UMTS. As the MAP in case of 3GPP bears sensitive functions like carrying session keys, authentication data etc., the MAP messages are to be protected. The 3GPP specification [9] provides the mechanisms and procedures to protect MAP protocol and the actual implementation of MAP can be found in [19].

[1] Telenor is a Norwegian mobile operator and also one of the world's major mobile operators

Based on the specification [9] and [12, p.74-76] general working of MAPsec can be defined as follows. The keys, algorithms and protection profiles for the protection of MAP are defined by establishing and negotiating Security Association (SA)s between MAP elements [9]. Now using this SA, the plaintext MAP is encrypted and the encrypted MAP messages are placed inside another MAP message along with the cryptographic checksum (i.e. Message Authentication Code (MAC)) [12]. The MAPsec SAs are distributed and negotiated in Key Administration Centre (KAC)s using Internet Key Exchange (IKE). Thus achieved protection by MAPsec provides following security services:

- Connectionless (cryptographic) data integrity of the MAP messages.
- Data origin authentication for the MAP messages.
- Replay protection for the MAP messages.
- Confidentiality (encryption) for the MAP messages (Optional).

The figure 2.4 show the entities and interfaces of MAPsec. The terms used in figure are shortly described below:

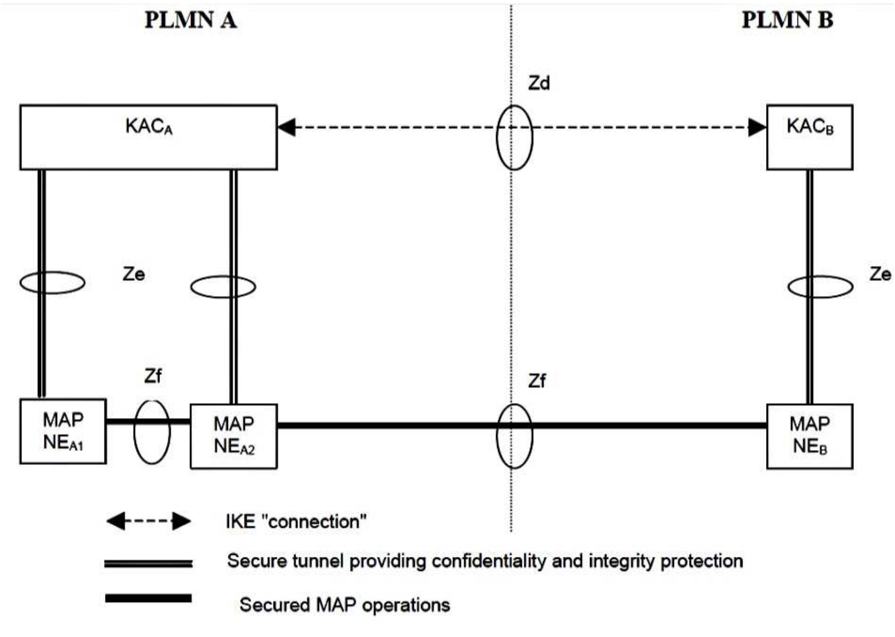
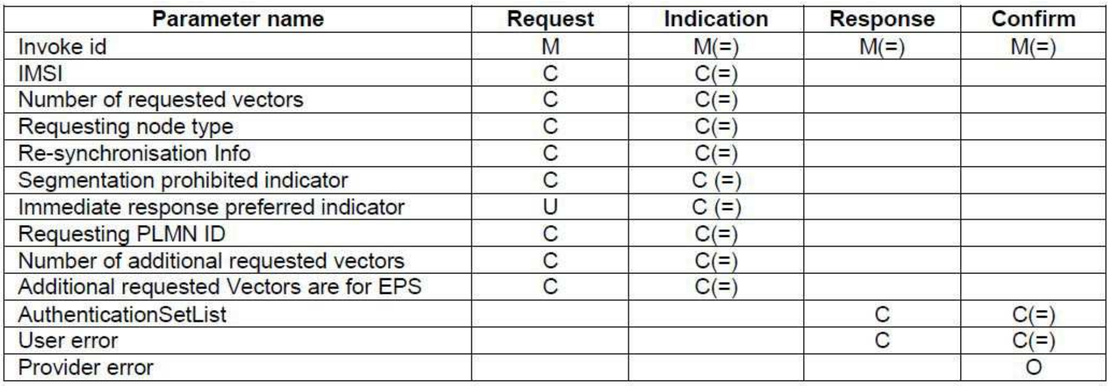


Figure 2.4: Overview of Zd, Ze and Zf interfaces [20].

- Zd-interface (KAC-KAC): to negotiate MAPsec SAs between two security domains or two Public Land Mobile Network (PLMN)s.
- Ze-Interface (KAC-NE): used to transport negotiated MAPsec SAs and relevant. security policy information from KAC to MAP-Network Element (NE) under same security domain.

- Zf-interface (NE-NE): For the MAPsec transactions within NEs of same domain or different domains. Received SAs are used to protect MAP operations between NEs.

The specification [19, p.98] lists the Authentication parameters for GSM/ UMTS. The clause 8.5 of specification [19] details MAP based Authentication Management services. VLR and HLR; SGSN and HLR; MME and HSS use service MAP_SEND_AUTHENTICATION_INFO parameters to retrieve Authentication Information from HLR or HSS. The HSS returns EPS authentication vectors in case the requester is MME and user is a UMTS user else if the node is not MME, HLR shall return authentication quintuplets for UMTS user and authentication triplets for GSM user [19]. The table 2.3 lists the parameters of MAP_SEND_AUTHENTICATION_INFO service as shown in table 8.5/2 of [19].



Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
Number of requested vectors	C	C(=)		
Requesting node type	C	C(=)		
Re-synchronisation Info	C	C(=)		
Segmentation prohibited indicator	C	C (=)		
Immediate response preferred indicator	U	C (=)		
Requesting PLMN ID	C	C(=)		
Number of additional requested vectors	C	C(=)		
Additional requested Vectors are for EPS	C	C(=)		
AuthenticationSetList			C	C(=)
User error			C	C(=)
Provider error				O

Table 2.3: MAP_SEND_AUTHENTICATION_INFO parameters [19, p.146].

The invoke id (invoke-ID), the first parameter in the list 2.3, is mandatory in all four services. In 3GPP specification, invoke-ID is defined as :"*This parameter identifies corresponding service primitives. The parameter is supplied by the MAP service-user and must be unique over each service-user/service-provider interface.*" [19, p.63]. The procedures for service invocation is described in clause 15.5.1 of [19]. This confirms that there is a unique identifier for each user service request tracked and maintained by the SN side.

The invoke-ID is detailed in ITU-T standards. The invocation, operation of invoke-ID, as defined in ITU-T standard Q.775 [18] should be different from other concurrent invocations. The concurrent invocations may be of either same operation or different operations [18, p.5]. Hence invoke-ID is an identifier of particular activation of operation. The ITU-T standard Q.775 in section 2.3.1 also states that the invoke-ID may take any value which can be mapped to integer and encoded to one octet. Thus, the invoke-ID is only one octet long (value ranging from -127 to 127 [18]). The Transaction IDs which uniquely identifies a dialogue can range from 1 to 4 octets as specified in [18, p.16]. The 3GPP specification [19, p.403] defines Transaction IDs as follow:

```
TransactionId ::= OCTET STRING (SIZE (1..2))
-- This type carries the value part of the transaction identifier
    which is used in the
-- session management messages on the access interface. The
    encoding is defined in
-- 3GPP TS 24.008
```

Thus, the length of Transaction IDs used in case of 3GPP operations is upto 2 octets. Again, the 3GPP specification [21] in clause 11.2.3.1.3 indicates that only 4 bits (5-8 bits of first octet) of standard L3(layer 3) message contains Transaction Identifier. The same specification also defines, an octet long Procedure transaction identity (PTI), in clause 11.2.3.1a. The function of both transaction ID and PTI are distinguish message flows.

The discussion above concludes that in 3GPP one octet invoke-ID is used while the length of transaction-ID may extend upto 2 octets.

2.4.2 NDS-IP and IPsec

Microsoft defines IP Security (IPsec) [22, 23] as *"Internet Protocol security (IPSec) is a framework of open standards for ensuring private, secure communications over Internet Protocol (IP) networks, through the use of cryptographic security services."* [24]. The LTE system is an all IP based system. So, the communication between nodes in CN is also IP based and native IP based protection applies to the traffic between the nodes. The security architecture for IP based network core in case of 3GPP networks is described in specification 3GPP TS 33.210 [25]. The protocol to protect native IP communication is termed as (IPsec) and is defined in IETF based RFC-4301 [22] which obsoletes RFC-2401 [26]. The IPsec provides security in the network (IP) layer [25, 12, 22]. The secured network layer not only provides protection at IP but also protects upper layers [22]. The IPsec provides integrity, data origin authentication, replay protection, confidentiality and limited protection against traffic flow analysis if confidentiality is applied [22, 27, 25]. The IPsec is mandated in IPv6 while it is also supported in IPv4 [12]. The details on IPsec and its operations are not mentioned here. However the working of IPsec is explained with a simple example of IPsec implementation later in chapter 3. There are set of IETF based RFC's detailing security of native IP based systems. The terms related to IPsec listed and described in table 2.4 below simplifies the understanding of working of IPsec.

Terms	Purpose
Security Domain	networks managed by a single administrative authority

Continued on next page ...

Table 2.4 – continued from previous page

Terms	Purpose
Security Gateway (SEG)	entities on the borders of security domains; NDS/IP traffic enters and leaves security domain via SEG
Security Associations (SAs)	establishment of shared security attributes between two network entities
Security Policy Database (SPD)	decides which security services are to be offered and in what fashion
Security Association Database (SAD)	Database containing parameters associated to the active security associations
Internet Key Exchange (IKE)	responsible for negotiation, establishment and maintenance of Security Associations

Table 2.4: Some Terms related to IPsec.

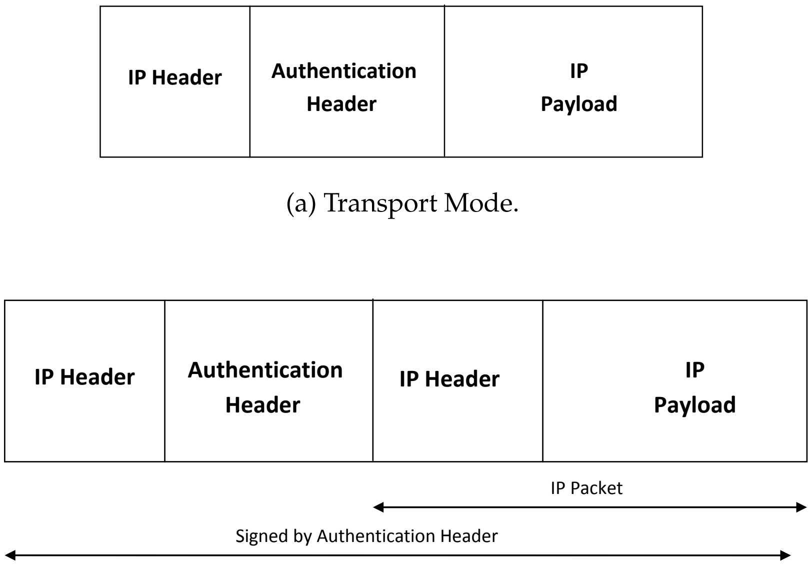
The IKE protocol [28, 29, 30] is implemented between the peers to negotiate the security parameters required to establish a secure connection [31]. Thus, established secure channel or tunnel is then used to exchange security parameters which are required to transmit user data [31]. Both the negotiation and exchange of security parameters are based on SA [32]. SA describes the relationship between two or more entities about the utilization of security services to communicate securely [32]. Each SA is identified by Security Parameter Index (SPI), IP Destination Address and security protocol identifier [25]. As, NDS/IP always use Encapsulating Security Payload (ESP) protocol, IP Destination Address and security protocol identifier are ESP SA endpoint and ESP protocol respectively.

There are two protocols: 1) Authentication Header (AH) [33] and 2) Encapsulating Security Payload (ESP) [34] used by IPsec for various operations. The Authentication Header (AH) provides connectionless integrity and data origin authentication for IP datagrams and to optionally provide protection against replays [33] while ESP provides confidentiality, data origin authentication, connectionless integrity, an anti-replay service and limited traffic flow confidentiality [34]. The IPsec can be implemented in two modes:

1. Transport Mode: This mode is the default mode of IPsec where only the payload of IP packet is encrypted [35, 27]. This mode is used for end to end communication [27]. The figure 2.5a illustrates that AH is added in between IP payload and IP header. This AH provides integrity and authentication to that packet as explained in [27]. Similarly figure 2.6 illustrates the ESP encryption of IP packet in transport mode.

2. Tunnel Mode: In this mode both the IP headers and IP payload are encapsulated by IPsec. This mode protects IP traffic between different networks. The figure 2.5b shows an IP packet encrypted by AH of IPsec in tunnel mode. Here the outer IP header contains the addresses of tunnel end points while the inner (encapsulated) IP header are used to route the traffic to final destination [27]. The figure 2.6 shows the ESP protection in tunnel mode.

The clause 11 of specification [36] mandates the use of integrity and confidentiality protection in accordance to NDS/IP (3GPP specification TS 33.210 [25]) in case of interfaces S3, S6a and S10. The S6a interface is the interface between MME and HSS. This interface carries on AKA process where the AV parameters are transported from HSS to MME. The mechanism of IPsec in case of 3GPP is provided in specification [25] and book [12] explains this mechanism in case of UMTS. The specification [25, p10.] states that: *"For NDS/IP-networks the IPsec security protocol shall always be ESP. For NDS/IP-networks it is further mandated that integrity protection/message authentication together with anti-replay protection shall always be used."*.



(a) Transport Mode.

(b) Tunnel Mode.

Figure 2.5: AH encryption of IP packet in different modes (modified from [27]).

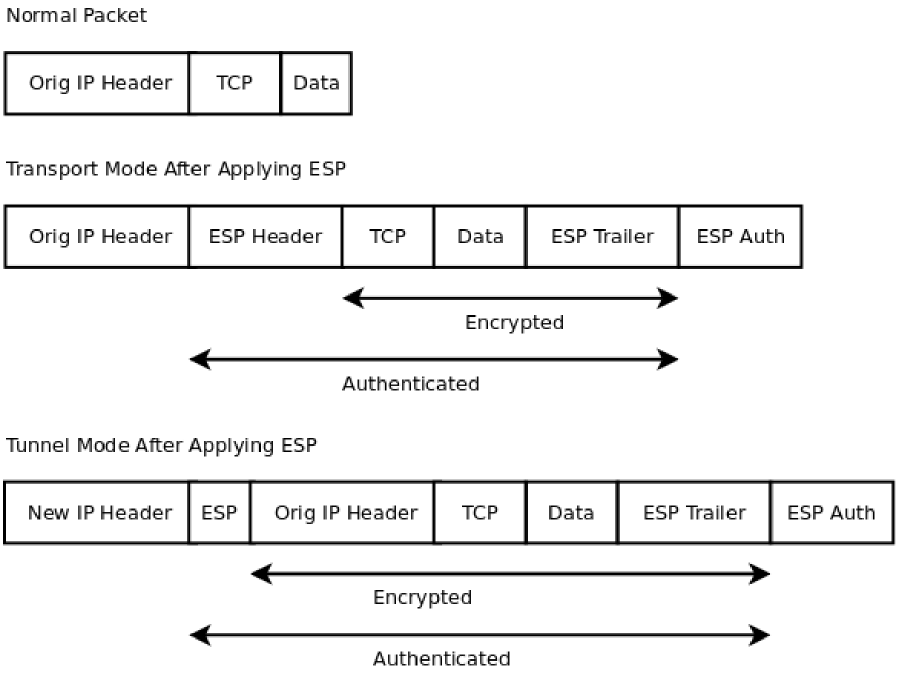


Figure 2.6: IP packet secured by ESP [37].

2.5 The Diameter Protocol

Diameter is a solution to the shortcomings of Remote Authentication Dial In User Service (RADIUS) protocol and is used in case of Authentication, Authorisation, and Accounting (AAA) purposes of network elements. Diameter in general (Diameter Base Protocol) is defined in IETF RFC 6733 [38]. This RFC 6733 [38] obsoletes RFC 3588 [39]. The major portion of the study was made before [39] was obsoleted, so some of the points described here based on RFC 3588 may contradict the latest specification RFC 6733. But, major changes in RFC 3588 included in RFC 6733 are studied and the analysis are tried to made based on latest specifications. All the diameter protocols are extensions of Diameter Base Protocol and every diameter nodes must support Diameter Base Protocol. The figure 2.7 shows the various diameter application on top of Diameter Base protocol.

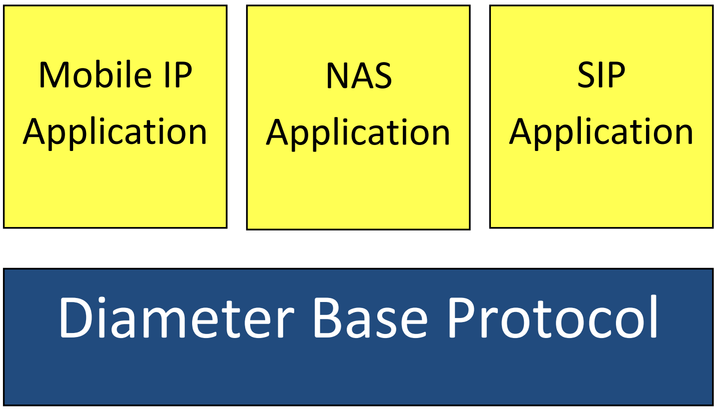


Figure 2.7: The Diameter Applications on top of Diameter Base Protocol [40].

The diameter protocol is better understood by defining some protocol related terms defined in RFC 6733 [38] and used in our case as below:

User is the entity requesting service for which Diameter Client generates authentication request to server.

Diameter Clients are diameter capable nodes attached in the edge of a network providing access control services to users in that network.

Diameter Server is a diameter node that provides Authentication, Authorisation, and Accounting (AAA) for particular realm.

Diameter Node is a host implementing diameter protocol. The clients, server and agents are diameter node.

Diameter Peers are those nodes with direct transport connection.

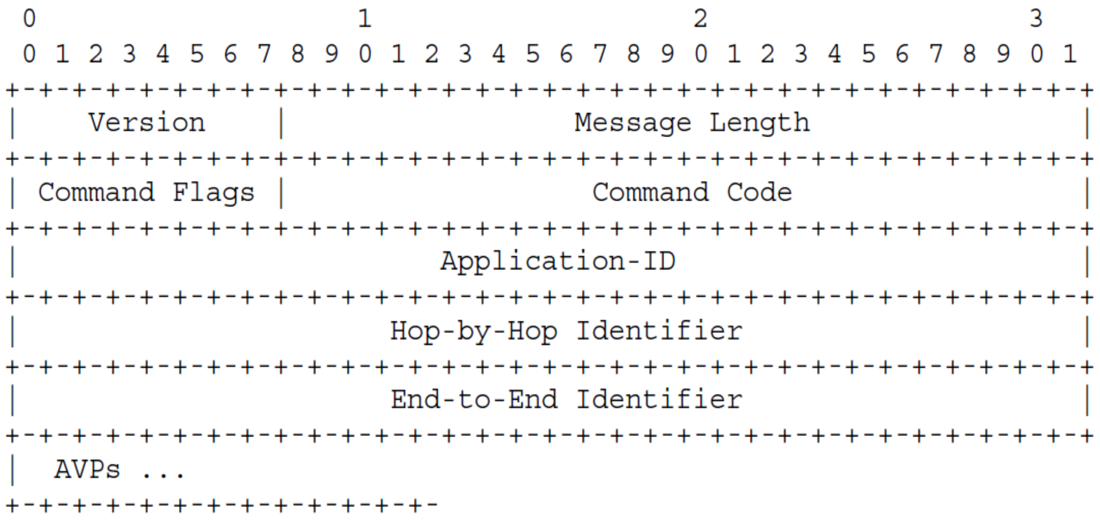
Diameter Agent provides relay, redirect, translation services.

Session keeps track of progression of a particular activity. A session ID is dedicated to a particular session.

Realm is like a domain but has a particular naming convention. The string immediately following '@' of identity is the realm. The Fully Qualified Domain Name (FQDN) gives the realm.

The **Attribute Value Pairs** contain header and encapsulate routing information as well as AAA information.

The Diameter Header format is shown in figure 2.8. The above listed terms and the terms in message headers are detailed in [38].



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                 Message Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Command Flags |                  Command Code                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Application-ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Hop-by-Hop Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      End-to-End Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AVPs ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2.8: The Diameter Header Format [38].

Some of the terms relevant to our studies are listed below. Following description is based on [38] and details on following terms can be found in [38] itself.

Command Code is of three octets. This value is used to communicate the command associated with the message. The Command Code values 16,777,214 and 16,777,215 are used for experimental use.

Application-ID is of four octets. This value identifies the application (authentication application, an accounting application, or a vendor-specific application) for which the message is applicable.

Hop-by-Hop Identifier is an unsigned 32-bit monotonically increasing integer field, with randomly generated start value. This values helps to match requests and replies. This value MUST be unique for a given connection and the value in corresponding answer MUST be same as in request. The answer message with unknown Hop-by-Hop Identifier MUST be discarded [38].

End-to-End Identifier is an unsigned 32-bit integer field. This value is used to detect duplicate messages.

2.5.1 Protection of Diameter Messages

The RFC 6733 mandates the secure transport of Diameter messages [38, p12.]. This recommendation further states Transport Layer Security (TLS)/Transmission Control Protocol (TCP) and Datagram Transport Layer Security (DTLS)/Stream Control Transmission Protocol (SCTP) as primary methods to exchange the diameter messages. The IPsec is mentioned to be the secondary method of securing Diameter messages. However, NDS-IP related 3GPP specifications mandates the use of IPsec to secure native IP based communication in [25, p9.]. The specification [25, p9.] states that: *"For native IP-based protocols, security shall be provided at the network layer. The security protocols to be used at the network layer are the IETF defined IPsec security protocols as specified in RFC-4301 [22] and in RFC-2401 [26].*. The 3GPP specification [36, p59.] mandates the use of NDS-IP for integrity and confidentiality protection of $S6a$ interface. And the 3GPP specification [41] mentions the use of Diameter protocol to exchange messages along $S6a$ interface. Hence from the 3GPP specifications [25, 36, 41] it can be concluded that Diameter messages in case of LTE are protected by IPsec security mechanisms. The mechanisms of protecting Diameter message by TLS or DTLS are not discussed here and the details can be found in [38].

2.5.2 Working of Diameter

The message sequence diagram shown in figure 2.9 shows the general steps and messages exchanged between User, Diameter Client and Diameter Server. The figure and its elaboration are based on the description provided in [40]. For the connection request from user, the client node (acting as an Network Access Server (NAS)) gathers credentials related to user and sends the access request message to the server. Now the server, checks the information received from client and authenticates the user. After checking for authentication, the server generates the response message. The content of this message is either user's access privileges or an error (reject) in cases of successful authentication and authentication failure respectively. The granular explanation for diameter message exchange between Diameter Client and Server is presented in implementation part in section 3.4.

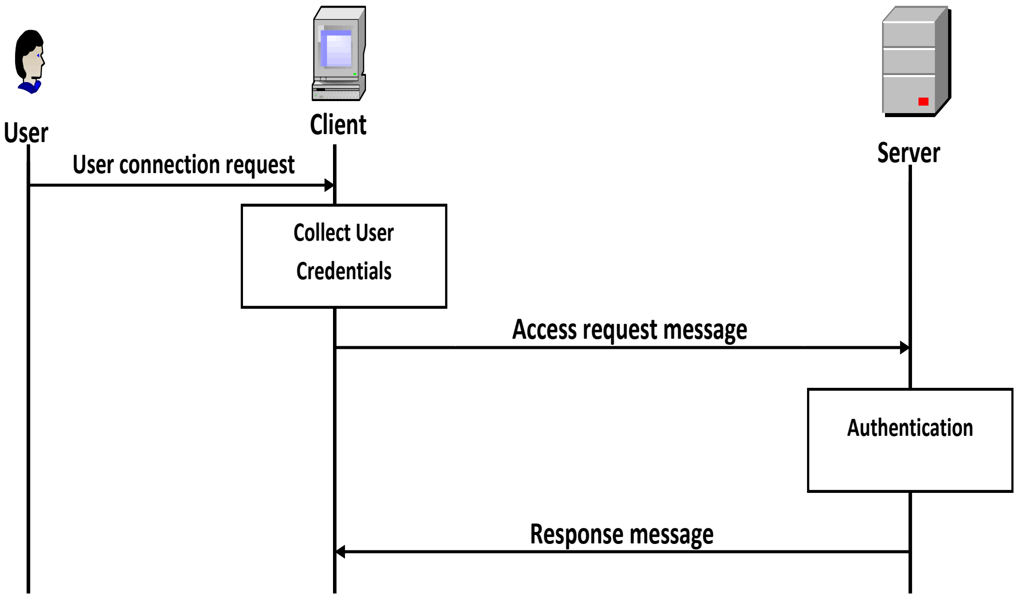


Figure 2.9: General Message Diagram for Diameter Authentication.

The connection between Diameter nodes (client and server in our case) is required before the application specific messages (in above case Access Request message and Response message) can be exchanged. In addition, the peers (Diameter nodes after connection is established) may require to discover peer and are required to exchange capabilities before the application specific operations can occur. These messages as specified in RFC-6733 [38] are : 1) Peer Connection, 2) Peer Discovery and 3) Capabilities Exchange. The figure 2.10 illustrates a Client-Server communication with these messages. The procedure for Peer Connection, Peer Discovery and Capability Exchange is elaborated in [38] and explained with help of example in section 3.4.

However the use of Diameter in case of 3GPP AKA does not exactly match the process shown in figure 2.9 as the authentication process is carried on MME which is equivalent to node $Client$ of Diameter. This requirement for 3GPP was a hurdle to implement Diameter protocol in case of LTE AKA. The effect of this hurdle is explained in section 3.6. The Diameter in case of 3GPP/LTE AKA is explained in section 2.5.3.

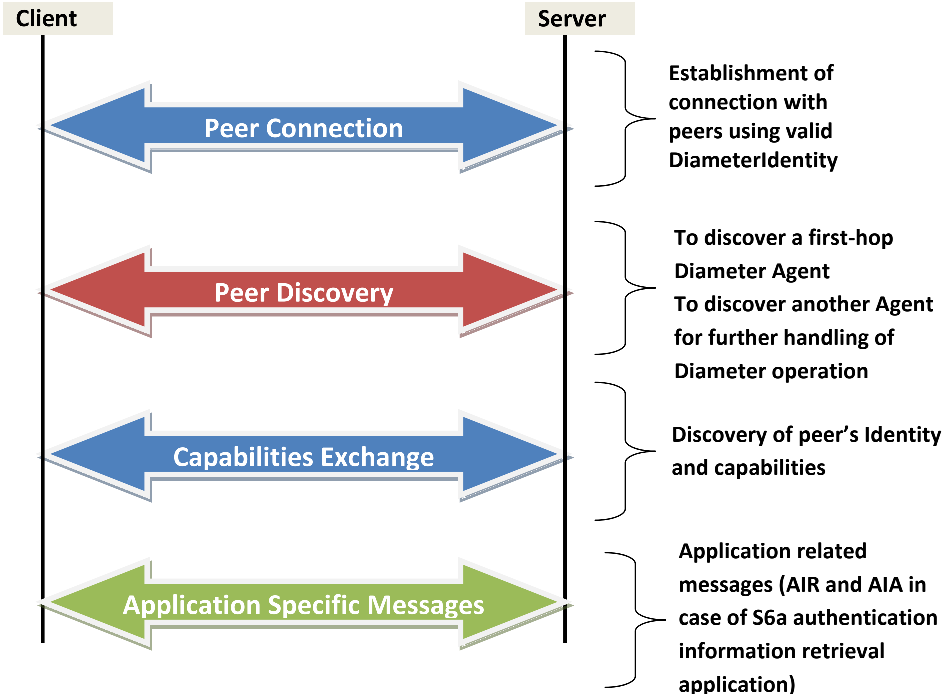

Client
Server
Peer Connection
Establishment of connection with peers using valid DiameterIdentity
Peer Discovery
To discover a first-hop Diameter Agent
To discover another Agent for further handling of Diameter operation
Capabilities Exchange
Discovery of peer's Identity and capabilities
Application Specific Messages
Application related messages (AIR and AIA in case of S6a authentication information retrieval application)

Figure 2.10: Diameter Message Exchange.

2.5.3 S6a Interface and Diameter in 3GPP

Diameter in case of IP based system is equivalent to MAP of SS7 system. The specification [41] indicates that Diameter protocol is used for the exchange of messages in $S6a$ and $S6d$ interfaces of IP based 3GPP standards and details the procedures related to MME-HSS (S6a) and SGSN-HSS (S6d) interfaces. The protocol specifications and message parameters for these interfaces are also detailed in the same document. The $S6a$ interface as defined in 3GPP specification TS 23.401 [42] is the link between MME and HSS. The same specification states that *"It($S6a$) enables transfer of subscription and authentication data for authenticating/authorizing user access to the evolved system (AAA interface) between MME and HSS"*.

The specification ETSI TS 129.272 [41] explains the use of diameter application for S6 interface. The functions of MME and HSS along with other entities are explained in specification [42]. The clause "5.2.3" of [41] explains the Authentication Procedures. As explained in the specification, the Authentication Information Retrieval procedure between MME and HSS is mapped to commands Authentication Information Request (AIR) and Authentication Information Answer (AIA) of Diameter protocol. Authentication Information from HSS is requested using AIR procedure. The IMSI and visited PLMN ID are mandated to be sent in request. The AIA procedure replies with Result Code and Authentication Information. The other elements of AIR and AIA and their mapping to 3GPP are tabled in [41] (Page 29). This result code is checked by MME. In case the check is success and AIA contains Authentication Information, the AVs received are used by MME [41] for further AKA steps. The Diameter application for AKA process is sketched in the figure 2.11. The messages Authentication Information Request (AIR) and Authentication Information Answer (AIA) of the figure 2.11 are Diameter commands and listings 2.1 and 2.2 show message format for these commands. The messages User Authentication Request and User Authentication Response are AKA messages exchanged between $User$ and SN. These messages are explained in section 2.2.

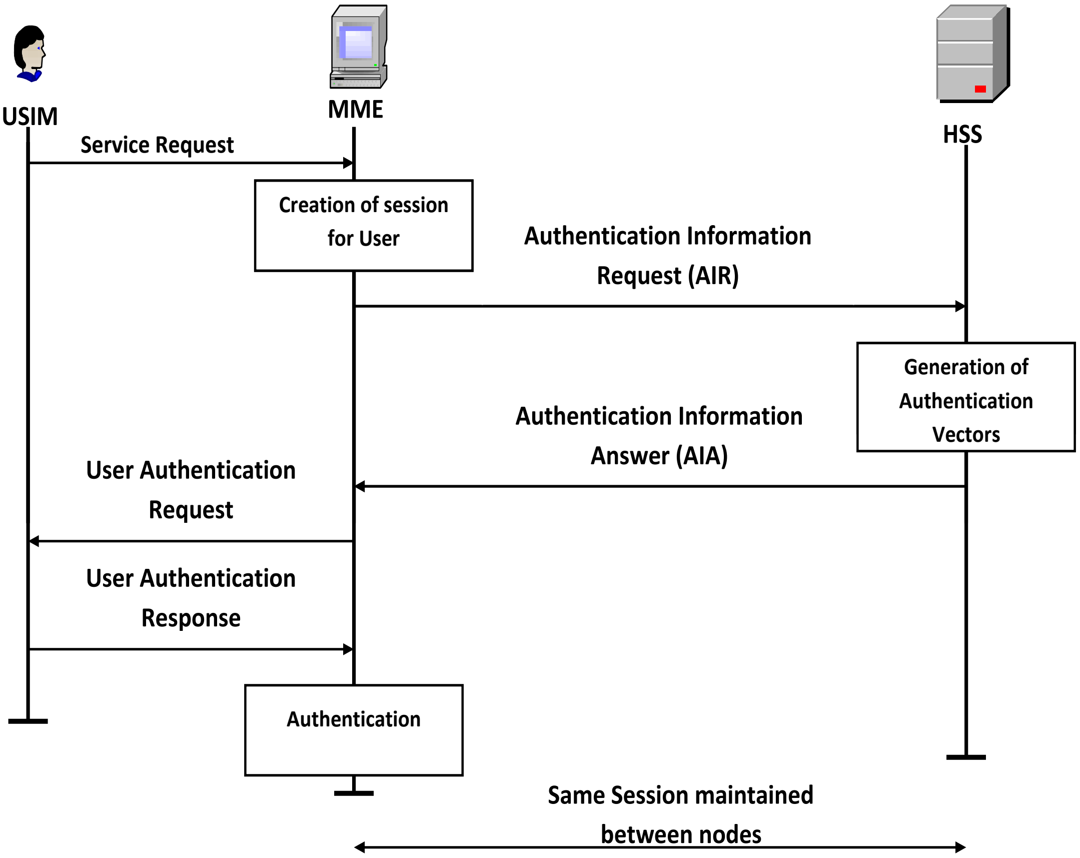

USIM
MME
HSS
Service Request
Creation of session for User
Authentication Information Request (AIR)
Generation of Authentication Vectors
Authentication Information Answer (AIA)
User Authentication Request
User Authentication Response
Authentication
Same Session maintained between nodes

Figure 2.11: Application of Diameter for AKA process.

The clause 7 of [41] states that a part from exceptions defined by this specification, diameter in case of 3GPP is used as per Diameter Base Protocol specified in IETF RFC 3588 [39] (now RFC 6733 [38]). The figure 2.12 shows the implementation of S6 interface. In the figure, it is seen that the link between MME and HSS is S6a interface. The other interface SGSN-HSS, S6d has similar implementation as S6a interface but is beyond our interest. The figure 4.1 shows the protocol stack for S6a interface. The figure shows that the AKA messages, specifically authentication parameters are carried out by Diameter Protocol. In addition the SCTP is used as transport protocol and IP is used in network layer. The control-plane for S6a interface as described in specification [42] is shown in figure 2.13. The specification ETSI TS 129 272 [41] in clause 7.1.8 specifies the diameter application for S6a interface. The clause defines S6a interface to be IETF vendor specific Diameter application where vendor is 3GPP with vendor identifier 10415 [2] as assigned by Internet Assigned Numbers Authority (IANA) [3].

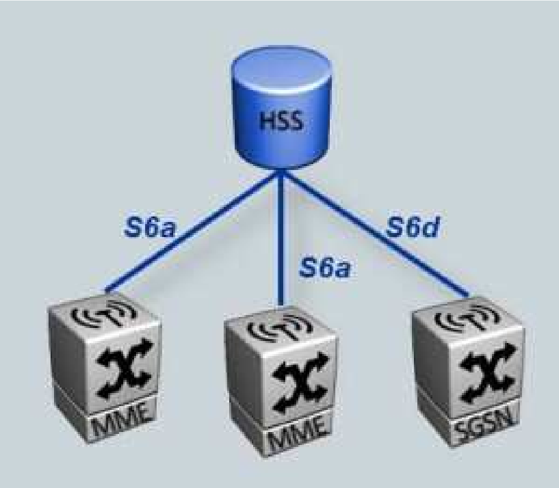


Figure 2.12: The S6 Interface [43].

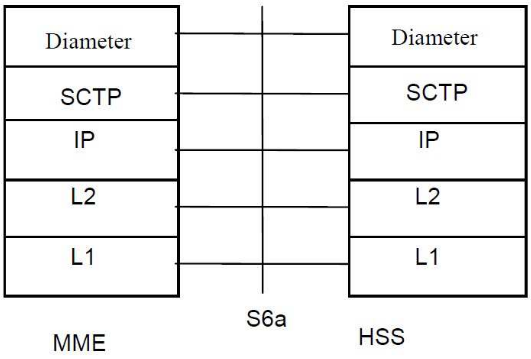


Figure 2.13: The S6a Control Plane [42].

As this work is meant to analyse the Authentication procedure between MME and HSS for the exchange of AV parameters, in depth study of the diameter protocol in case of authentication is made. The latter objectives of Diameter protocol Authorisation and Accounting are not dealt here. Our study is focused

[2] The list of private enterprise numbers can be found in http://www.iana.org/assignments/enterprise-numbers

[3] The IANA is responsible for the global coordination of the DNS Root, IP addressing, and other Internet protocol resources.

towards the NDS security and particularly towards exchange of AKA messages. From earlier description, it is known that AKA messages in case of LTE are exchanged between MME of SN and HSS along S6a interface. Hence details are focused for authentication procedures between MME and HSS Diameter based $S6a$ interface. The specification [41] in clause 5.2.3 describes the authentication procedures. The clause defines two commands: 1) Authentication Information Request and 2) Authentication Information Answer used in case of Authentication Information Retrieval. The message format for these commands are provided by same specification in clauses 7.2.5 and 7.2.6. The message format for AIR and AIA commands are shown in listings Message format 2.1 and 2.2 respectively.

Message format 2.1: Message format for AIR Command

```
< Authentication-Information-Request> ::= < Diameter Header: 318,
   REQ, PXY, 16777251 >
< Session-Id >
[ Vendor-Specific-Application-Id ]
{ Auth-Session-State }
{ Origin-Host }
{ Origin-Realm }
[ Destination-Host ]
{ Destination-Realm }
{ User-Name }
*[Supported-Features]
[ Requested-EUTRAN-Authentication-Info ]
[ Requested-UTRAN-GERAN-Authentication-Info ]
{ Visited-PLMN-Id }
*[ AVP ]
*[ Proxy-Info ]
*[ Route-Record ]
```

Message format 2.2: Message format for AIA Command

```
< Authentication-Information-Answer> ::= < Diameter Header: 318,
   PXY, 16777251 >
< Session-Id >
[ Vendor-Specific-Application-Id ]
[ Result-Code ]
[ Experimental-Result ]
[ Error-Diagnostic ]
{ Auth-Session-State }
{ Origin-Host }
{ Origin-Realm }
* [Supported-Features]
[ Authentication-Info ]
*[ AVP ]
*[ Failed-AVP ]
*[ Proxy-Info ]
*[ Route-Record ]
```

In line 2 of both the Message formats, there is Session-Id. This Session-Id is defined in Diameter Base Protocol ([38, p116]). The Session-Id Attribute Value Pair (AVP) identifies a specific session. It is created by the nodes which initiates the session. Most of the cases it is Client and here in case of 3GPP/LTE it is SN. The RFC 6733 [38] explains Session-Id as one of the content of request command issued by Client to Server whenever a user requests access to the network. The statement: *"The Session-Id is a means for the clients and servers to correlate a Diameter message with a user session."*, stated in [38, p98] shows that there a particular session maintained between User, Client and Server for a particular service request. This Id initiated by Client identifies different users and different services. The format of the Session-Id AVP as mentioned in [38, p117] is $< DiameterIdentity >; < high32bits >; < low32bits > [; < optionalvalue >]$. Here, the $DiameterIdentity$ is an FQDN name of a Diameter node and identifies that node and $optional\ value$ is implementation specific.

In lines 11 and 12 of Message format 2.1, authentication information for EUTRAN and UTRAN/GERAN(GSM/EDGE Radio Access Network) are requested. The AVP format for Requested-EUTRAN-Authentication-info is given below:

AVP format 2.3: AVP format for Requested-EUTRAN-Authentication-info

```
Requested- EUTRAN-Authentication-Info ::= <AVP header: 1408 10415>
[ Number-Of-Requested-Vectors]
[ Immediate-Response-Preferred ]
[ Re-synchronization-Info ]
*[AVP]
```

The AIA command answers AIR with the authentication information. This answer contains the Authentication Vector for requested user. The AVP format for Authentication-info is shown in AVP format 2.4 and AVP format 2.5 contains the AVP format for EUTRAN vector.

AVP format 2.4: AVP format for Authentication-Info

```
*[ E-UTRAN-Vector ]
*[UTRAN-Vector]
*[GERAN-Vector]
*[AVP]
```

AVP format 2.5: EUTRAN Authentication vector parameters

```
[ Item-Number ]
{ RAND }
{ XRES }
{ AUTN }
{ KASME }
*[AVP]
```

Similarly, AVP formats for the UTRAN and GSM/EDGE Radio Access Network (GERAN) are shown in sections 7.3.12, 7.3.19 and 7.3.20 of [41].

Chapter 3

Lab Experiments

The last objective of this thesis work is to *"Try to construct a realistic scenario and software simulation that shows that a mix-up attack can be attainable in practice"*, based on the collected information. The attempts were made to simulate Diameter based LTE nodes i.e. MME and HSS; implement $S6a$ interface and analyse the session mix-up attack with reference to the working of Diameter based $S6a$ interface. In order to attain this goal several tools for Diameter implementation were searched. The search criteria was further narrowed by imposing following requirements of the tools:

- The tools should be an open source and be available as a free software.
- The software should be easy to install, configure and handle.
- There should be certain learning outcomes while using the software.
- Already available source codes and applications were on prime focus as it is hard to develop new ones within limited time and also there are lot of parameters and processes which were to be considered for development of new Diameter application.

This chapter explains the set up of the software tools, their configurations, limitations of the tools, working of tools with examples, the difficulties faced to attain the desired goal and further suggestions regarding requirements of the tools or codes to achieve the full flexed goal in future.

3.1 Tools Used

The work comprises the communication between two diameter nodes: one client acting as MME of SN and another, the server acting as HSS/AuC of HN. Also a short study on IPsec, the security protocol for native IP based communication, is made in this work. Thus, in order to address these issues and requirements, the following tools were chosen:

- Personal Computer (PC)'s: 1
  - Operating System (OS): Ubuntu 11.10 32-bit.
  - IPv4: 192.168.1.229.
  - root of terminal: ashim@ubuntu:
  - Purpose: Used as a server and host of VirtualBox for Client.
- VirtualBox: 1
  - Purpose: To run client machine running on Linux (Ubuntu).
- Ubuntu running in VirtualBox: ip: root of terminal: client@client-VirtualBox:
  - OS: Ubuntu 11.10 32 bit.
  - IP: 192.168.1.196
  - Purpose: For the role of Client.
- IPsec-tools: This tool is used to implement IPsec communication between server and client.
- freeDiameter: This tool is used to configure client and server as Diameter nodes and to analyse Diameter based communication between these nodes.
- Wireshark: This tool is to sniff and analyse the network characteristics and to verify different protocols are running between different machines.

3.2 Ubuntu based Commands

The implementation work is carried out under Linux system using machines running on Ubuntu OS. In Linux based systems, several commands are used to install, configure and execute different programs. The list of such commands used during the procedure of installations, configuration and execution and their purposes are tabulated in table 3.1. These installations , configurations and execution of the software programs are described in further sections.

Commands	Purpose
sudo apt-get install wireshark	to install Wireshark
sudo wireshark	to run Wireshark

Continued on next page

Table 3.1 – continued from previous page

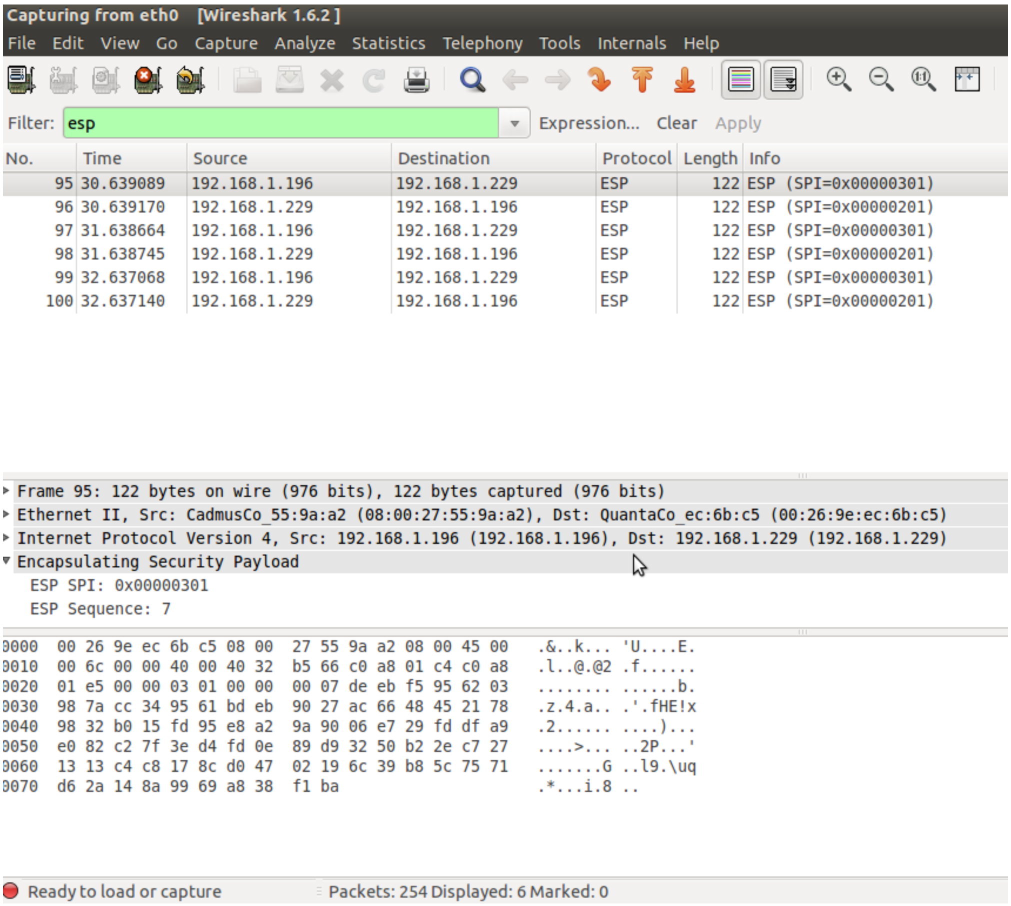
Commands	Purpose
sudo apt-get install tcpdump	to install tcpdump
sudo tcpdump -vv src or dst *ipadd_of_source_or_destination*	to capture packets to or from *ipadd_of_source_or_destination* ; *ipadd_of_source_or_destination* = IP address
sudo apt-get install ipsec-tools	to install ipsec-tools
sudo gedit /etc/ipsec-tools.conf	to edit the configuration file ipsec-tools.conf
• sudo setkey -f /etc/ipsec-tools.conf • sudo /etc/init.d/setkey start • sudo /etc/init.d/setkey stop • sudo /etc/init.d/setkey restart	• to start the IPsec • to start the IPsec • to stop the IPsec • to restart the IPsec
freeDiameterd -c /home/client/thesis/freediameter/fDbuild/conf/freediameter.conf	to start freeDiameter daemon with freediameter.conf file in specified path

Table 3.1: Ubuntu based Commands.

3.3 IPsec Tools

The IPsec can be configured in two different modes. i.e. Transport mode and Tunnel mode in different ways. The configuration can be made such that keys are manually assigned or automatically assigned. Here we configure IPsec in two nodes: server and client in transport mode. The specification states that IPsec must be configured in tunnel mode in case of inter-domain communication and transport mode may be used in case of intra-domain. But for the configuration in tunnel mode, two different public IPs were required. So, as this thesis deals with the security mechanisms provided by IPsec, labour was not made on networking part. Instead, two nodes were configured in transport mode with manually assigned keys and analysis was continued. The stepwise description for installations and configurations of IPsec in case of Linux based machines is detailed in [44, 45].

The IP traffic between server and client, after starting IPsec in both, will have additional Encapsulating Security Payload (ESP) field from IPsec. This can be seen when such traffic is monitored either by tcpdump or Wireshark tools. The IP traffic for a ping message from client(192.168.1.196) to server(192.168.1.229) is captured by Wireshark which is shown in screenshot 3.1. In the screenshot the ESP fields in addition to normal IP fields can be observed. Also, the SPI associated with the request and answer can be seen in the figure. Similarly the IPsec between two machines can be configured enabling both the flavours: ESP and AH. This can be done by removing the comment (removing the # sign) from lines 9,11,20 and 24 from B.2 and lines 8, 10, 19 and 23 from B.5. In this case the secured IP packet contains two additional fields of AH and ESP. The screenshot for this case is attached in figure C.1 in Appendix C. The capture of tcpdump is shown in listing 3.1 which also verifies IPsec ESP communication between two machines in transport mode.


Capturing from eth0 [Wireshark 1.6.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

Filter: esp

No.	Time	Source	Destination	Protocol	Length	Info
95	30.639089	192.168.1.196	192.168.1.229	ESP	122	ESP (SPI=0x00000301)
96	30.639170	192.168.1.229	192.168.1.196	ESP	122	ESP (SPI=0x00000201)
97	31.638664	192.168.1.196	192.168.1.229	ESP	122	ESP (SPI=0x00000301)
98	31.638745	192.168.1.229	192.168.1.196	ESP	122	ESP (SPI=0x00000201)
99	32.637068	192.168.1.196	192.168.1.229	ESP	122	ESP (SPI=0x00000301)
100	32.637140	192.168.1.229	192.168.1.196	ESP	122	ESP (SPI=0x00000201)

```
▸ Frame 95: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
▸ Ethernet II, Src: CadmusCo_55:9a:a2 (08:00:27:55:9a:a2), Dst: QuantaCo_ec:6b:c5 (00:26:9e:ec:6b:c5)
▸ Internet Protocol Version 4, Src: 192.168.1.196 (192.168.1.196), Dst: 192.168.1.229 (192.168.1.229)
▾ Encapsulating Security Payload
    ESP SPI: 0x00000301
    ESP Sequence: 7
```

```
0000  00 26 9e ec 6b c5 08 00  27 55 9a a2 08 00 45 00   .&..k... 'U....E.
0010  00 6c 00 00 40 00 40 32  b5 66 c0 a8 01 c4 c0 a8   .l..@.@2 .f......
0020  01 e5 00 00 03 01 00 00  00 07 de eb f5 95 62 03   ........ ......b.
0030  98 7a cc 34 95 61 bd eb  90 27 ac 66 48 45 21 78   .z.4.a.. .'.fHE!x
0040  98 32 b0 15 fd 95 e8 a2  9a 90 06 e7 29 fd df a9   .2...... ....)...
0050  e0 82 c2 7f 3e d4 fd 0e  89 d9 32 50 b2 2e c7 27   ....>... ..2P...'
0060  13 13 c4 c8 17 8c d0 47  02 19 6c 39 b8 5c 75 71   .......G ..l9.\uq
0070  d6 2a 14 8a 99 69 a8 38  f1 ba                     .*...i.8 ..
```

● Ready to load or capture   Packets: 254 Displayed: 6 Marked: 0

Figure 3.1: IP traffic secured with IPsec ESP.

Outputs 3.1: Traffic captured by tcpdump.

```
client@client-VirtualBox:~$ sudo tcpdump -vv src or dst
    192.168.1.196
```

```

16:02:30.370100 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ESP (50), length 108)
    client-VirtualBox.local > ubuntu.local: ESP(spi=0x00000301,seq=0x4), length 88
16:02:30.370262 IP (tos 0x0, ttl 64, id 63043, offset 0, flags [none], proto ESP (50), length 108)
    ubuntu.local > client-VirtualBox.local: ESP(spi=0x00000201,seq=0x4), length 88
16:02:31.369138 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ESP (50), length 108)
    client-VirtualBox.local > ubuntu.local: ESP(spi=0x00000301,seq=0x5), length 88
```

3.4 freeDiameter

The freeDiameter [46] is an open source software for the implementation of Diameter protocol. Using this software, machines can be configured as Diameter nodes and thus the machines can exchange diameter messages. The software is built in language C and implements diameter protocol as specified in RFC 3588 [39]. This software can be installed in various OSs including Linux based ubuntu system. The machines where this software is installed in our case are ubuntu based machines. This software has some dependencies which are required to install and configure properly for the successful implementation of freeDiameter itself. The website of the freeDiameter software [46] provides all the information about its complete installations, configurations and implementations for various purposes. This software is easy to install and understand and also provides a way to run various extensions to diameter base protocol. This software does not accommodate all the applications of diameter. Although the software for various applications can be developed, it requires lot of effort, time and experts. Owing to the above limitations, it was not possible to build applications supporting all the features required for our analysis and running under freeDiameter.

The freeDiameter software, after installations, contains configuration files freeDiameter.conf, several extensions(.fdx), extension specific configuration files (.conf), daemon freeDiameterd to start/stop the software. The content of typical configuration files is attached in Appendix B. The freeDiameter.conf file is the main file where configurations are made. A diameter node (server or client) contains a freeDiameter.conf each. The paths for various extensions and their conf files, its certificate, peers, the transport mode etc. are provided in freeDiameter.conf file. Only the nodes which are declared as peers in freeDiameter.conf files and have valid certificates can communicate with each other. The texts in sample configuration file B.1 in Appendix B explains more about the content of configuration file.

The freeDiameter was installed and configured in two machines. The successful configurations was checked by loading a test application and sending message from client to the server. The freeDiameter daemon (freeDiameterd) was started on both the machines (with no IPsec started on both machines); a signal was sent from client to server. Based on the observed output, the working of Diameter protocol can be clarified in following steps (The descriptions are based on output observed in client side. The server client was started after server so, the output of freeDiameter daemon in server side i.e. C.1 shows that no peer is connected.):

1. **Peer Connection:** The client and server use DiameterIdentity of each others to connect to each other. The line 16 of Configuration files B.3 and line 14 of Configuration files B.6 in Appendix B shows the declaration of peers in server and client respectively. The declaration:

   ConnectPeer = "client-VirtualBox.ed.ntnu.no" {ConnectTo = "192.168.1.196"; NO_TLS; };

   in server side specifies "client-VirtualBox.ed.ntnu.no" with IP="192.168.1.196" as a peer which can be connected without TLS negotiation. The term "client-VirtualBox.ed.ntnu.no" is the DiameterIdentity of client. There is similar declaration in Client side declaring server as its peer.

2. **Peer Discovery:** Here only two nodes are used. There are no agents, relays or transition nodes in between server and client. So, the peer is discovered as per static configuration as mentioned in step 1.
   The line 45 of C.2 shows that connection is established to the server.

3. **Capabilities Exchange:** After the connection between peers is established (line 45 of C.2), Capability Exchange Messages are exchanged and the peer's identity, protocol version number, identifier of supported diameter application, security mechanisms etc. are discovered. This can be seen in lines 50-62 in C.2.
   After the capabilities are exchanged, both peers are now in STATE_OPEN state (line 62 of C.2). Now these peers can carry on subsequent processes. The values of common codes, flags etc. for capability exchange messages in C.2 complies with the values defined in [38, p.62].

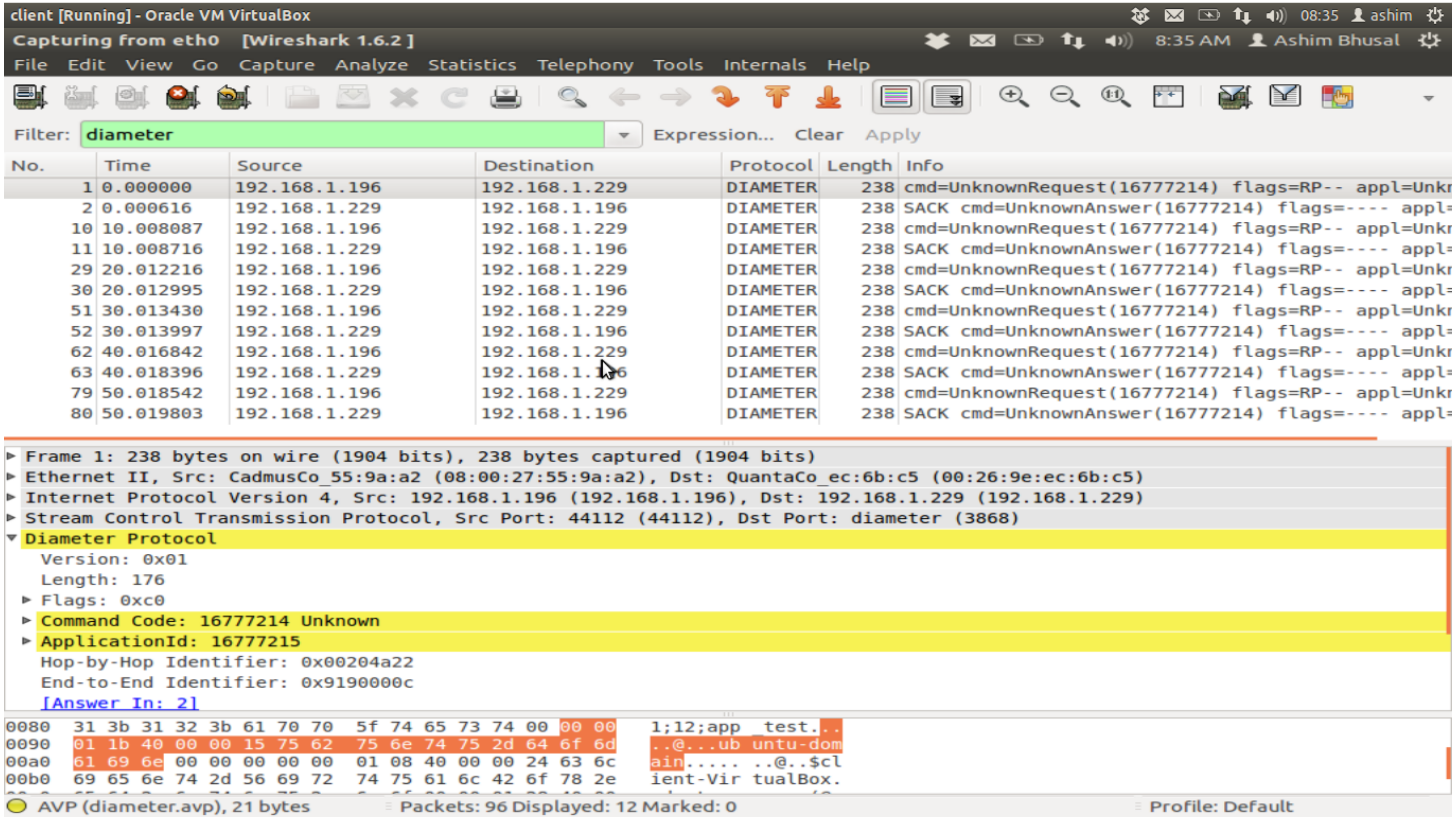
The Client-Sever Diameter communication is also captured by Wireshark with filter set to diameter. The screenshots 3.2 and 3.3 show the traffic patterns observed in Wireshark. In figure 3.3 Session-ID is seen as clear text. It was possible to observe Session-Id in clear text as the pattern observed here are of unprotected diameter messages i.e. IPsec was not started and TLS flag was set to NO_TLS in freeDiameter configuration file. The traffic was generated by using test application with command code (CC)= 16,777,214. This value of command code is reserved for experimental use by IANA, so the traffic shows the test application as unknown. The list of some values of our interest and observed in the Diameter traffic are as tabled in table 3.2.

Parameters	Observed Values
Hop-by-Hop Identifier	0x00204a22
End-to-End Identifier	0x9190000c
Session-Id	client-VirtualBox.ed.ntnu.no;1358407961;12;app_test

Table 3.2: Intended Diameter Parameters and their values.

The values of Hop-by-Hop Identifier and End-to-End Identifier, as observed in Wireshark (refer figures 3.2 and 3.3), are $00204a22$ and $9190000c$ respectively. These identifier are 32 bits integer value presented in decimal format. This Hop-by-Hop Identifier is used to match the request and reply while End-to-End Identifier is used for duplicate detection. The value of Session-Id as observed in screenshot and listed in table is: "client-VirtualBox.ed.ntnu.no;1358407961;12; app_test" which corresponds the value mentioned in the section 2.5. This value of Session-Id is decomposed as:

⇒ DiameterIdentity:	client-VirtualBox.ed.ntnu.no
⇒ high 32 bits:	1358407961
⇒ low 32 bits:	12
⇒ optional value:	app_test


client [Running] - Oracle VM VirtualBox
08:35 ashim
Capturing from eth0 [Wireshark 1.6.2]
8:35 AM Ashim Bhusal
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

Filter: diameter

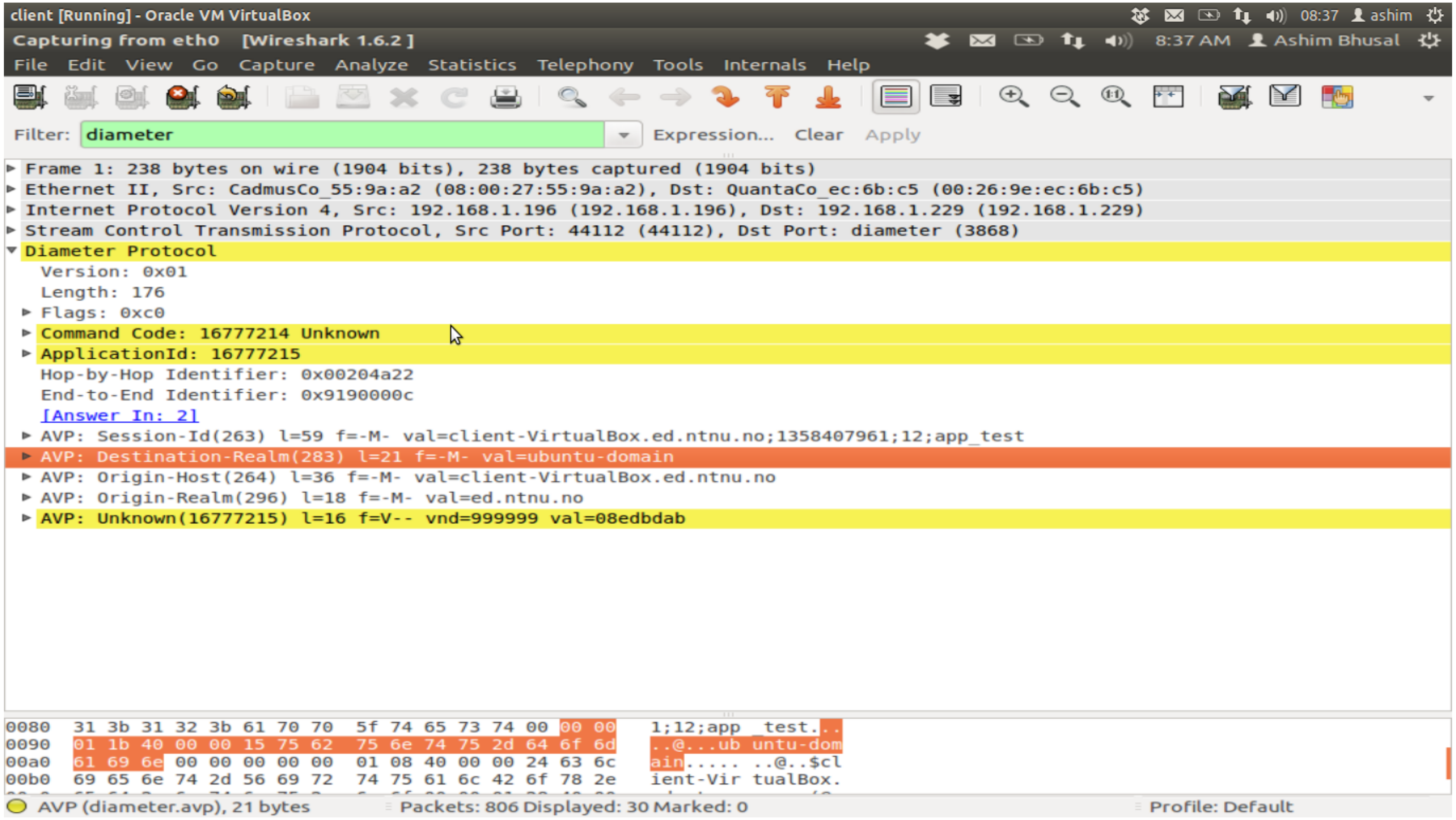
```
No.  Time       Source         Destination    Protocol Length Info
   1 0.000000   192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
   2 0.000616   192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
  10 10.008087  192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
  11 10.008716  192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
  29 20.012216  192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
  30 20.012995  192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
  51 30.013430  192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
  52 30.013997  192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
  62 40.016842  192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
  63 40.018396  192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
  79 50.018542  192.168.1.196  192.168.1.229  DIAMETER    238 cmd=UnknownRequest(16777214) flags=RP-- appl=Unkr
  80 50.019803  192.168.1.229  192.168.1.196  DIAMETER    238 SACK cmd=UnknownAnswer(16777214) flags=---- appl=
```

```
▶ Frame 1: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits)
▶ Ethernet II, Src: CadmusCo_55:9a:a2 (08:00:27:55:9a:a2), Dst: QuantaCo_ec:6b:c5 (00:26:9e:ec:6b:c5)
▶ Internet Protocol Version 4, Src: 192.168.1.196 (192.168.1.196), Dst: 192.168.1.229 (192.168.1.229)
▶ Stream Control Transmission Protocol, Src Port: 44112 (44112), Dst Port: diameter (3868)
▼ Diameter Protocol
    Version: 0x01
    Length: 176
  ▶ Flags: 0xc0
  ▶ Command Code: 16777214 Unknown
  ▶ ApplicationId: 16777215
    Hop-by-Hop Identifier: 0x00204a22
    End-to-End Identifier: 0x9190000c
    [Answer In: 2]
```

```
0080  31 3b 31 32 3b 61 70 70  5f 74 65 73 74 00 00 00   1;12;app _test...
0090  01 1b 40 00 00 15 75 62  75 6e 74 75 2d 64 6f 6d   ..@...ub untu-dom
00a0  61 69 6e 00 00 00 00 00  01 08 40 00 00 24 63 6c   ain..... ..@..$cl
00b0  69 65 6e 74 2d 56 69 72  74 75 61 6c 42 6f 78 2e   ient-Vir tualBox.
```

AVP (diameter.avp), 21 bytes   Packets: 96 Displayed: 12 Marked: 0   Profile: Default

Figure 3.2: Screenshot showing Diameter traffic.


client [Running] - Oracle VM VirtualBox
08:37 ashim
Capturing from eth0 [Wireshark 1.6.2]
8:37 AM Ashim Bhusal
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

Filter: diameter

```
▶ Frame 1: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits)
▶ Ethernet II, Src: CadmusCo_55:9a:a2 (08:00:27:55:9a:a2), Dst: QuantaCo_ec:6b:c5 (00:26:9e:ec:6b:c5)
▶ Internet Protocol Version 4, Src: 192.168.1.196 (192.168.1.196), Dst: 192.168.1.229 (192.168.1.229)
▶ Stream Control Transmission Protocol, Src Port: 44112 (44112), Dst Port: diameter (3868)
▼ Diameter Protocol
    Version: 0x01
    Length: 176
  ▶ Flags: 0xc0
  ▶ Command Code: 16777214 Unknown
  ▶ ApplicationId: 16777215
    Hop-by-Hop Identifier: 0x00204a22
    End-to-End Identifier: 0x9190000c
    [Answer In: 2]
  ▶ AVP: Session-Id(263) l=59 f=-M- val=client-VirtualBox.ed.ntnu.no;1358407961;12;app_test
  ▶ AVP: Destination-Realm(283) l=21 f=-M- val=ubuntu-domain
  ▶ AVP: Origin-Host(264) l=36 f=-M- val=client-VirtualBox.ed.ntnu.no
  ▶ AVP: Origin-Realm(296) l=18 f=-M- val=ed.ntnu.no
  ▶ AVP: Unknown(16777215) l=16 f=V-- vnd=999999 val=08edbdab
```

```
0080  31 3b 31 32 3b 61 70 70  5f 74 65 73 74 00 00 00   1;12;app _test...
0090  01 1b 40 00 00 15 75 62  75 6e 74 75 2d 64 6f 6d   ..@...ub untu-dom
00a0  61 69 6e 00 00 00 00 00  01 08 40 00 00 24 63 6c   ain..... ..@..$cl
00b0  69 65 6e 74 2d 56 69 72  74 75 61 6c 42 6f 78 2e   ient-Vir tualBox.
```

AVP (diameter.avp), 21 bytes Packets: 806 Displayed: 30 Marked: 0 Profile: Default

Figure 3.3: Screenshot showing Session-Id AVP.

3.4.1 Alternatives to freeDiameter

There are several other software or software/hardware implementing Diameter protocol. Some of these are $Seagull$ [1], dsTest [2], $MAPS^{TM}$ Diameter Protocol Emulator [3], Open Diameter [4] and so on. Some of these software like dsTest, $MAPS^{TM}$ Diameter Protocol Emulator are more intended towards business and large scale implementations, some like $Seagull$ is a traffic generator. These software lacked enough learning goals. The java codes avaiable in [47] for AIR and AIA were also found. But the attempt to obtain the full package overview or installation package was failed as the developer did not respond to the email seeking suggestions. The Open Diameter could have been a best alternative to this software but the complexity in its dependencies increased the preferences of easy freeDiameter which has equal capability as of Open Diameter. Thus, after loosing some time in search of effective and usable tool for simulation, freeDiameter was considered the most appropriate one.

3.4.2 Limitations of freeDiameter

The freeDiameter is easy in installation, configuration and operation. Even the website [46] provides full description of some of its applications with video tutorials. These convenience in using freeDiameter do not mask its limitation to encompass several application. The software is not matured one to support all the application of Diameter protocol. There is no mechanisms developed and explained in freeDiameter website regarding the support of freeDiameter in case of 3GPP applications. For instance there are no extensions for implementation of S6 interface. The software does not provide mechanism of sending user specific data from server to client. In the case of S6 interface where diameter is used to exchange authentication information from MME to HSS and vice versa, along with the success message HSS sends AVs to MME. This feature can not be achieved using freeDiameter. The freeDiameter is developed based on RFC 3588 which is obsoleted by RFC 6733. The developers have stopped working and providing support on DiamEAP project [5] based on freeDiameter which could have been useful in our case. There are no more extensions and upgrades provided by freeDiameter software as well as Open Diameter software. The only support provided by freeDiameter software is from one of the developer of freeDiameter via email. The other software/hardware intended for large scale solution and application of diameter protocol may compensate all above mentioned limitations. But these software are not open source and do not have good learning outcomes. In spite of these limitations, for the best understanding of

[1] http://gull.sourceforge.net/

[2] http://www.developingsolutions.com/products/s6-interface/

[3] http://www.gl.com/maps-diameter-protocol-emulator.html

[4] http://diameter.sourceforge.net/

[5] http://diameap.yagami.freediameter.net/

diameter protocol, for the development of other skills like using linux system, C languages, knowledge of dependencies etc, the freeDaimeter software is chosen. And an analogous system(model) to meet our requirement is developed.

3.5 Modelling(Proposed)

A software model is developed to implement MME and HSS communication based on recommended protocols. This model is a result of the theories provided in chapter 2 and is based on the software and tools as explained in above section 3.1. This model implements S6a interface as defined in section 2.5.3 and uses Diameter protocol as explained in sections 2.5 and 3.4. The Diameter protocol as well as IPsec are implemented between two nodes, one acting as Client and other as Server. The users or mobile devices are attached to the Client. The Client requests for authentication of the users trying to attach with it from the server. The server contains a database where the credentials of user are stored. The server finds if the user are authentic or authorised by comparing the user credentials sent by Client and the credentials for respective user in its database. Thus, we can call the model as Client-Server Communication. As we are dealing with 3GPP LTE systems, these user, client and server of our model should map to LTE components. In the case of this model, User of the model corresponds UE, Client of the model correspond MME and the Server of the model represents HSS of LTE system. The table 3.5 lists the corresponding mapping and the function of each entities. Thus, the terms of model and corresponding components of LTE may be used interchangeably throughout the text. Also Client and Server of the model are both Diameter enabled nodes.

Name used in LTE system	Name used in model	Function
UE/USIM	Users	Service Requester or Supplicant
MME of SN	Client	Service Provider/provides service after successful authentication/requests for authentication parameters with Server
HSS of HN	Server	Authenticator/contains database storing user related information/ generates and supplies parameters required for authentication of user

Table 3.3: Mapping of entities of LTE system to the modelled system.

The roles of Client and Server are taken by two linux run machines. First the network for both machines and host is set such that Client and Server can communicate with each other. This successful communication is checked by using ping IP of other machines. After the successful pinging between these machines, all the IP based communication between server and client are secured by IPsec. The freeDiameter is installed in both the server and client. Both the machines are configured to communicate as Diameter nodes with each other (refer sections above for installation, configuration and testing). Now the machines are able to talk with Diameter protocol and the the Diameter messages are protected by IPsec. These two machines (server and client) are now peers to each other. The users are attached to client machine. This can be achieved either by use of hostapd/supplicant tools or inserting some database in client containing lists of users. The server stores the user specific parameters in its database. For each request from client on behalf of a user, the server generates AVs and supplies to client. It is assumed that the requesting user is a subscriber to that server. Now the client sends some parameters received from server to user and receives response from user. This response from user is compared to response from server in client. If the comparison is successful, service is granted to that user. The figure 3.4 illustrates a simple model explained above. The figure shows that the communication between Users and Client is IP based and the Client and Server communicate using IP in network layer and Diameter signalling protocol. The traffic between Client and Server are protected in network layer by using IPsec which is an option to TLS protection in case of Diameter. We were not successful to implement the model mentioned above. The reasons for this failure is mentioned in section 3.6.

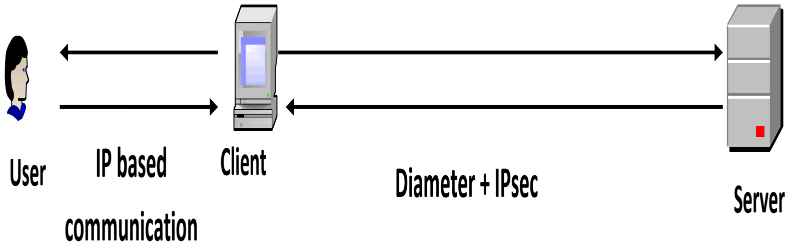


Figure 3.4: General model used for implementation.

3.6 Problem with Implementation

The implementation phase was not achieved as expected due to several factors. The initial decisions on what to implement were made after superficial studies of UMTS related SS7 and MAPsec and LTE related Diameter and IPsec protocols. Based on the available depth of information, availability of free tools and current practices recommended by specifications, the implementation LTE based

IPsec and Diameter protocols was finalised. The tools **ipsec-tools** and **freeDiameter** were selected to implement IPsec and Diameter protocols respectively. Using these tools IPsec and Diameter protocols were successfully implemented between two machines. The applications developed and provided along with freeDiameter tool were not enough to implement communication taking place in $S6a$ interface. This inability of freeDiameter was unanticipated and known very lately. The main problem faced was to fetch user specific AVs from database to node acting as HSS. Although freeDiameter is programmed in C, all the applications provided were in form of extensions (.fdx). And defining new 3GPP specific Diameter commands AIR and AIA required additions of some AVPs to the Dictionary. This complexity of 3GPP specific Diameter protocol, lack of skill to develop new diameter applications in limited time and same problem persisting in other alternatives to freeDiameter forced the implementation part to be undone. The solutions were not found even for **Open Diameter** software.

3.7 Suggestions for Implementation

The $S6a$ specific Diameter application, which is missing in many of Diameter specific tools, are either to be developed on own or should be implemented by using tools which are not free. The development of freeDiameter and Open Diameter specific applications require frequent support from the developers. Unavailability of other projects related to $S6a$ application imposed hurdles to find supports and suggestions from other people. The support received from developer of Mobicent [48] was too late to use it. But in this project, the source codes [47] for AIR and AIA are provided. The availability of java source adds a positive sign to carry on implementation of $S6a$ in future. From the personal experience of the use of freeDiameter and Open Diameter, and late look up of the AIR, AIA related java source from Mobicents, I will like to suggest that this code and several other 3GPP related sources may lead to easier implementation.

Chapter 4

Discussions

In this chapter, protection of AKA messages by underlying lower layer protocols are discussed. The response of session mix-up attack to the protection provided by protocols like Diameter/IPsec and SS7/MAPsec is analysed and the decisions for attainability of session mix-up attack are made. The issues which are not covered by this thesis work are also explained.

4.1 The Case of Inter Domain operations

The Inter Domain operations, i.e when the SN and HN does not belong to same provider, are governed by standards. There is need of mutual understanding and interoperability between the providers. So this need of interoperability is only fulfilled if both the network operators have common operation mechanisms. In order to attract huge mass of subscribers, network operators extend their coverage by roaming agreement with many other operators. The service of interoperation is only possible if all the operators within agreement apply same kind of standards. Thus, in case of inter domain, the recommended standards are followed. In case of the LTE system, the AKA messages are carried as $S6a$ application by Diameter protocol and the Diameter messages are protected by IPsec. Similarly, for the case of UMTS the AKA messages are carried by MAP of SS7 signalling and the protection of MAP messages is provided by MAPsec in application layer. The figures 4.1 and 4.2 illustrate the protocol stack for $S6a$ application and MAP application respectively.

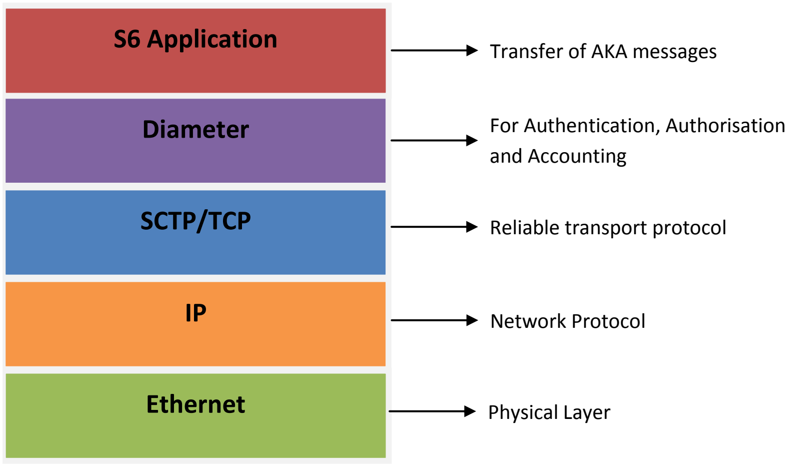
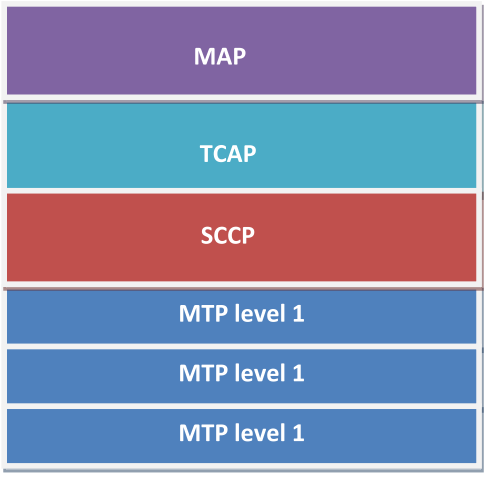


Figure 4.1: The S6 Protocol Stack (modified from [43, 49]).



MAP → Mobile Application Part
TCAP →Transaction Capabilities Application Part
SCCP →Signalling Connection Control Part
MTP →Message Transfer Part

Figure 4.2: The MAP Protocol Stack (modified from [50]).

4.1.1 Case of LTE: IPsec and Diameter

The specifications and several other documents showed that in LTE authentication information are carried out via $S6a$ interface. In addition it was confirmed that $S6a$ applications run over Diameter protocol and protection as per NDS-IP is used for messages over $S6a$ interface. With reference to these information obtained from several sources which are already explained in Chapter 2 the $S6a$ protocol stack can be drawn as shown in figure 4.1. The $S6a$ protocol stack shows different protocols running beneath AKA protocol.

The Diameter protocol defines Session-Id AVP to identify the sessions of different services. From the studies presented in earlier chapters, this Session-Id was found to be composed of $DiameterIdentity$, monotonically increasing $64bits$ and $Optional\ Values$ separated by semicolon (;). The 64 bits are further decomposed to higher 32 bits and lower 32 bits both represented by maximum of 10 decimal digits. These higher and lower bits are again separated by ′;′. The 64 bits values used to identify the session makes it difficult to guess this identifier. It may require upto 2^{32} attempts to find the session-Id if only higher bits are used. In addition the server may restrict the users to use the resources only for predefined time by issuing *Authorization- Lifetime AVP*. After the expiration of *Authorization- Lifetime AVP* all the user specific state information are released and the user needs to be re-authorised to get the access to the network. If this AVP is also implemented then the session-Ids are valid only for limited times. So, this time may add some more burden for the attacker to find the session-Id of a user within limited time. The other two fields of Diameter Message: The Hop-by-hop Identifier and End-to-End Identifier should match for Diameter request and corresponding Diameter Answer. The Hop-by-hop Identifier keeps track of a single connection while the later identifier is used to detect the duplicates of the Diameter messages. The attacker in order to get access to a connection created by one user, requires to know the Hop-by-hop Identifier value. And the messages generated or sent by the attacker should contain the same value of End-to-End Identifier to fraud the server from detecting the duplicate messages. So, it seems that the attacker needs some more labour, in addition to finding the Session-Ids, to swap the sessions of two users.

A part, 3GPP considers that the $S6a$ interface carrying sensitive authentication messages should be secured. So, integrity and confidentiality protection of this interface is mandated and this confidentiality and integrity protection are to be provided by IPsec ESP mechanisms. Thus, the sufficient length of Session-Ids, use of Hop-by-hop Identifier, End-to-End Identifier and the protection of the Diameter messages by IPsec mechanisms can provide effective resistant to the Session Mix-up attack. Although specifications recommend Diameter protocol, no specifications were found to mandate the use of Diameter protocol. Thus, the AKA may still be vulnerable to Session Mix-up attack in case the recommended protocols are not implemented or the protocols are not implemented in recommended ways.

4.1.2 Case of UMTS: MAPsec and SS7

In the UMTS system, MAP, the mobile specific part of SS7 signalling is used to carry the authentication information request and answer messages between SN and HN of a user. The TCAP protocol defined under SS7 by ITU-T manages concurrencies of operations. For this Transaction-IDs and invoke-IDs are used. The protocol stack for MAP messages is illustrated in figure 4.2. The length of invoke-ID is defined to be one octet while the length of transaction-IDs may range from one to four octets. For the case of 3GPP it is found that the size of transaction-IDs is (1..2) octets. Further some 3GPP specifications show that only four bits of first octet of layer 3 message are used to identify upto 16 transactions. This number of concurrent transactions may be further extended by one octet value of PTI. These number of bits of invoke-ID and transaction-ID used in MAP of 3GPP can be considered as short enough to protect the IDs from being guessed or being generated. Only upto $2^8 = 512$ guesses or generations are required to find the value of identifiers in case one octet is used. But 3GPP considers authentication retrieval messages to be sensitive and requiring further protections. The 3GPP defines MAPsec for the protection of the MAP messages. If the MAPsec as specified in 3GPP is implemented between the network elements, than the following services are confirmed:

- Data Integrity
- Data Origin Authentication
- Replay protection
- Confidentiality (encryption)

In such case, the above mentioned service with no doubts, will protect the MAP messages and their sessions even if the length of invoke-IDs or transaction-IDs are not sufficient for protection.

4.2 The Case of Intra Domain operations

The Intra Domain operations are the operations performed whenever a user is served by the SN controlled by same network operator to which the user has subscription. In such services, all the participating nodes are owned by same operator. In this case, all the communicating parties i.e. The User, the SN and the HN are under same domain, so it is the operator itself to decide which protocols or security mechanisms to be used. Although the trends to use the standards is increasing, some vendors may chose to stick with their own proprietary or other mechanisms which are already in use. Some of several reasons for this may be, the requirement of new resources and experts of new mechanisms and so on. In order to find the mechanisms used by operators inside their own domain, several people from different operators were contacted. As, none of the contacted people revealed the information regarding the mechanisms used in intra domain case, the attempts to analyse Session Mix-up attacks in intra domain communication were not successful. The behaviour and possibility of Session Mix-up attack in intra domain scenario is still unknown.

Unless the actual implementation mechanisms adopted by the operators are known, it can not be confirmed that all the operators are not vulnerable to Session Mix-up attack.

Chapter 5

Conclusions and Further Extensions

5.1 Conclusions

In this thesis work, we have found some protocols and security mechanisms to counter Session Mix-up attack against UMTS/LTE AKA protocols. Thus, found protocols are recommended by 3GPP and ETSI specifications and are defined in ITU-T, IETF and 3GPP standards. The large portion of the study is based on LTE/EPS system and some analysis is also made for UMTS system. The protocols recommended by 3GPP to transfer EPS based AKA messages is found to be Diameter which is further protected by IPsec. Both the Diameter and IPsec are defined by IETF RFCs. The explanations of these protocols are assisted by implementing them and capturing the Diameter and IP based traffics between two nodes. The format of messages exchanged are looked over and session related fields are checked with general format explained in RFC 6733. Similarly in case of UMTS, it was found that MAP of SS7 is protected by MAPsec mechanisms.

The theories presented in Chapter 2 and discussed in Chapter 4 suggest that the Session Mix-up attack is effectively countered by sufficiently long and secure Session-Identifiers which are further protected by IPsec in case of LTE and UMTS where Diameter and IPsec are implemented. While the length of invoke-Id and Transaction-Id used in MAP of UMTS were not sufficiently long and may be vulnerable to Session Mix-up Attack. But the implementation of MAPsec to secure these MAP messages prevent the system being vulnerable to Session Mix-up attack. The possibility of the attack in case of intra domain where the security and signalling mechanisms may be vendor specific is still unknown. In the case of intra domain system, the attack may be possible if the session identifiers are not sufficiently formatted and protected. Hence this thesis work shows that both the UMTS and LTE system are protected against Session Mix-up attacks if all the recommended protocols are implemented effectively. No new security mechanisms or change in some steps of AKA procedure are required to counter this attack if standards are followed. The only requirement is the confirmation of implementation of the recommended protocols.

After performing this thesis work, broad protocol level knowledge of 3GPP UMTS and LTE system were obtained. In addition, several recommendations and standards in field of communication and security were well understood by implementing some of them. The recent recommendations for signalling mechanisms and their protection were known after performing this work.

There were certain problems faced which hindered the thesis work. The expected information regarding real time scenario and intra domain communication were not obtained since no response was heard from the people contacted. The deficiency of such information led to make the analysis based only on the recommendations from specifications. So the analysis presented in Chapter 4 (Discussions) are based for Inter Domain communication and for Intra Domain only if the recommended standards are used by the operators. The tool which was chosen to implement Diameter based $S6a$ application was not able to fetch AVs from database, send back to client and to make authentication decision on client not server. This inability of the software was unanticipated. The other options to the selected tools were also not sufficient to provide the authentication mechanisms along $S6a$ interface. Although the implementation phase took more than two third of thesis time, the problem with the tool shaded the whole work of implementation which was tried to perform.

In spite of the lack of desired information and unanticipated limitation in tools, this thesis work concludes that the Session Mix-up attack in AKA of Diameter based LTE communication is not attainable in practice while there are lot of unsolved holes left in MAP based UMTS where AKA may be vulnerable to the Session Mix-up attack.

5.2 Further Works

This thesis work has not drawn the conclusions based on the results of implementation. In addition the analysis made in this thesis are not complimented by the real scenarios implemented by the operators. Owing to aforementioned weaknesses, this work can be further developed with following countermeasures.

- **Analysis by Implementation**
  Here, the limitations in free tools and limited time and knowledge to develop new tools let down the implementation phase. In this case it can be suggested that, either to use some tools which may not be free providing

full flex implementation or the work should be assisted by professional software developers who can build and integrate new modules to the existing ones. Also the implementation of MAPsec to show the attainability of Session Mix-up in case of UMTS may be performed. Some suggestions regarding the tools are also provided in section 3.7.

- **Support from Operators**
  This thesis work has a bad experience in collection of the information by directly contacting the people working for several service providers. There were no responses from seven people working for five companies in three different countries(refer A.1 of Appendix A). This experience suggests that either the work should be carried out under some companies or should be performed by the insider of the companies. The acquisition of the procedures in real implementation is the only basis to conclude whether the operators are prone to the Session Mix-up attack or not. Further suggestions for mitigation, in case of possibility of this attack, can only be provided after deriving some conclusions.

- **Analysis in interworking environment**
  The proper implementation of this work can be further used to analyse the Session Mix-up attack in case of interworking environment. The backward compatibility of a system with the legacy system may be vulnerable to attacks related to old system. For instance, a replay attack was possible on a UMTS system working in a GSM environment. Further works can be carried to show if the attack is possible in case of LTE system working in UMTS environment and vice versa or LTE/UMTS user served by GSM environment.

References

[1] S. F. Mjølsnes and J.-K. Tsay, "Computational security analysis of the umts and lte aka." Submitted on 17 Mar 2012 to ArXiv.org `http://arxiv.org/abs/1203.3866`. [cited at p. 3, 12, 13]

[2] S. F. Mjølsnes and J.-K. Tsay, "Computational analysis of the umts and lte authentication and key agreement protocols." In the Eighth Workshop on Formal and Computation Cryptography FCC 2012, June 27-28. [cited at p. 3]

[3] S. F. Mjølsnes and J.-K. Tsay, "A vulnerability in the umts and lte authentication and key agreement protocols." In Sixth International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, NTNU, 2012, October 16-19. [cited at p. 3]

[4] 3GPP TS 09.02 V7.15.0 (2004-03), "3^{rd} Generation Partnership Project; Technical Specification Group Core Network; Mobile Application Part (MAP) specification (Release 1998)," March 2004. [cited at p. 7]

[5] 3GPP TS 33.102 V11.5.0 (2012-12), "3^{rd} Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 11)," December 2012. [cited at p. 9, 12]

[6] 3GPP TS 33.401 V12.6.0 (2012-12), "3^{rd} Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 12)," December 2012. [cited at p. 9]

[7] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi, *LTE Security*. A John Wiley and Sons, Inc, September 2010. Published Online: 4 OCT 2010 at Library of NTNU. [cited at p. 9, 14]

[8] A. Bhusal, "Validation of attacks on the UMTS/LTE AKA protocol," June 2012. A specialisation project on Information Security, Department of Telematics, NTNU. [cited at p. 9, 11, 12, 13]

[9] ETSI TS 133 200 V6.1.0 (2005-03), "Universal Mobile Telecommunications System (UMTS); 3G Security; Network Domain Security (NDS); Mobile

Application Part (MAP) application layer security (3GPP TS 33.200 version 6.1.0 Release 6)," April 2005. [cited at p. 13, 14, 15]

[10] G. M. Køien, "An evolved UMTS Network Domain Security architecture," September 2002. Technical Report N28/2002, Telenor. [cited at p. 13, 14]

[11] G. M. Køien, "UMTS Network Domain Security." `http://www.telektronikk.com/volumes/pdf/1.2002/Page_164-171.pdf`. Online; accessed 30-September-2012. [cited at p. 14]

[12] V. Niemi and K. Nyberg, *UMTS Security*. A John Wiley and Sons, Inc, November 2003. [cited at p. 14, 15, 17, 19]

[13] Wikipedia, "Transaction Capabilities Application Part." `http://en.wikipedia.org/wiki/Transaction_Capabilities_Application_Part`, 2012. Online; accessed 30 December 2012. [cited at p. 14]

[14] ITU-T, "ITU-T Recommendation Q.772 – Specifications of Signalling System No. 7 – Transaction capabilities application part – Transaction capabilities information element definitions," June 1997. [cited at p. 14]

[15] ITU-T, "ITU-T Recommendation Q.771 – Specifications of Signalling System No. 7 – Transaction capabilities application part – Functional description of transaction capabilities ," June 1997. [cited at p. 14]

[16] ITU-T, "ITU-T Recommendation Q.773 – Specifications of Signalling System No. 7 – Transaction capabilities application part – Transaction capabilities formats and encoding," June 1997. [cited at p. 14]

[17] ITU-T, "ITU-T Recommendation Q.774 – Specifications of Signalling System No. 7 – Transaction capabilities application part – Transaction capabilities procedures," June 1997. [cited at p. 14]

[18] ITU-T, "ITU-T Recommendation Q.775 – Specifications of Signalling System No. 7 – Transaction capabilities application part – Guidelines for using transaction capabilities," June 1997. [cited at p. 14, 16, 17]

[19] 3GPP TS 29.002 V11.5.0 (2012-12), "3^{rd} Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mobile Application Part (MAP) specification (Release 11)," December 2012. [cited at p. 14, 16, 17]

[20] 3GPP TS 33.200 V5.0.0 (2002-03), "3^{rd} Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network Domain Security; MAP application layer security (Release 5)," March 2002. [cited at p. 15]

[21] 3GPP TS 24.007 V11.0.0 (2012-06), "3^{rd} Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mobile radio interface signalling layer 3; General aspects (Release 11)," June 2012. [cited at p. 17]

[22] S. Kent and K. Seo, "Security Architecture for the Internet Protocol." RFC 4301, 2005. `http://www.ietf.org/rfc/rfc4301.txt`. [cited at p. 17, 23]

[23] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap." RFC 6071 (Informational), Feb. 2011. `http://www.ietf.org/rfc/rfc6071.txt`. [cited at p. 17]

[24] Microsoft, "Internet Protocol Security (IPSec)." `http://technet.microsoft.com/en-us/library/cc783420(v=ws.10).aspx`. Online; accessed 29-November-2012. [cited at p. 17]

[25] ETSI TS 133 210 V10.3.0 (2011-06), "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; 3G security; Network Domain Security (NDS); IP network layer security (3GPP TS 33.210 version 10.3.0 Release 10)," June 2011. [cited at p. 17, 18, 19, 23]

[26] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol." RFC 2401, 1998. `http://tools.ietf.org/pdf/rfc2401.pdf`. [cited at p. 17, 23]

[27] Microsoft, "IPSec Protocol Types." `http://technet.microsoft.com/en-us/library/cc757712(v=ws.10).aspx`. Online; accessed 29-November-2012. [cited at p. 17, 18, 19, 20]

[28] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)." RFC 2409 (Proposed Standard), Nov. 1998. Obsoleted by RFC 4306, updated by RFC 4109. [cited at p. 18]

[29] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)." RFC 5996 (Proposed Standard), Sept. 2010. Updated by RFC 5998. [cited at p. 18]

[30] P. Hoffman, "Algorithms for Internet Key Exchange version 1 (IKEv1)." RFC 4109 (Proposed Standard), May 2005. `http://www.ietf.org/rfc/rfc4109.txt`. [cited at p. 18]

[31] Top Global, "How IPSec Works." `http://www.topglobalusa.com/mb8000.htm`. Online; accessed 29 October 2012. [cited at p. 18]

[32] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Security Architecture for the Internet Protocol." RFC 2408, 1998. `http://www.ietf.org/rfc/rfc2408.txt`. [cited at p. 18]

[33] S. Kent, "IP Authentication Header." RFC 4302 (Proposed Standard), Dec. 2005. http://www.ietf.org/rfc/rfc4302.txt. [cited at p. 18]

[34] S. Kent, "IP Encapsulating Security Payload (ESP)." RFC 4303 (Proposed Standard), Dec. 2005. http://www.ietf.org/rfc/rfc4303.txt. [cited at p. 18]

[35] Wikipedia, "IPsec." http://en.wikipedia.org/wiki/IPsec, 2012. Online; accessed 28 October 2012. [cited at p. 18]

[36] 3GPP TS 33.401 V12.5.1 (2012-10), "3^{rd} Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 12)," October 2012. [cited at p. 19, 23]

[37] Chainring Circus, "QoS Pre-Classify and End-to-End QoS." http://www.chainringcircus.org/qos-pre-classify-and-end-to-end-qos/. Online; accessed 29 October 2012. [cited at p. 20]

[38] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter Base Protocol." RFC 6733, 2012. http://tools.ietf.org/html/rfc6733. [cited at p. 21, 22, 23, 24, 26, 29, 36]

[39] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol." RFC 3588, 2003. http://www.ietf.org/rfc/rfc3588.txt. [cited at p. 21, 26, 35]

[40] IBM, "Introduction to Diameter Get the next generation AAA protocol." http://www.ibm.com/developerworks/library/wi-diameter/index.html. Online; accessed 30 November 2012. [cited at p. 21, 23]

[41] ETSI TS 129 272 V10.6.0 (2012-04), "Universal Mobile Telecommunications System (UMTS); LTE; Evolved Packet System (EPS); Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter protocol (3GPP TS 29.272 version 10.6.0 Release 10)," April 2012. [cited at p. 23, 25, 26, 27, 28, 30]

[42] 3GPP TS 23.401 V11.3.0 (2012-09), "3^{rd} Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 11)," September 2012. [cited at p. 25, 27]

[43] Developing Solutions, "S6a/S6d Interface." http://www.developingsolutions.com/products/s6-interface/. Online; accessed 28 November 2012. [cited at p. 27, 46]

[44] IPsec HOWTO, "The official IPsec Howto for Linux." `http://www.ipsec-howto.org/`, 2007. Online; accessed 21 September 2012. [cited at p. 33]

[45] Keimpe de Jong, "IPSecHowTo." `https://help.ubuntu.com/community/IPSecHowTo`, 2011. Online; accessed 21 September 2012. [cited at p. 33]

[46] W. Project and NICT, "freeDiameter Diameter open implementation." `http://www.freediameter.net/trac/browser/freeDiameter/LICENSE`, 2008-2011. Online; accessed 28 September 2012. [cited at p. 35, 40]

[47] Google Project Hosting/brainslog, "mobicents Open Source Cloud Communications." `http://code.google.com/p/mobicents/source/browse/trunk/servers/jain-slee/resources/diameter-s6a/events/src/main/java/org/mobicents/slee/resource/diameter/s6a/events/`. Online; accessed 27 December 2012. [cited at p. 40, 43]

[48] Mobicents, "Mobicents The Open Source Cloud Communications Platform." `http://www.mobicents.org/products.html`. Online; accessed 27 December 2012. [cited at p. 43]

[49] LTE AND BEYOND, "Interfaces and their protocol stacks." `http://www.lteandbeyond.com/2012/01/interfaces-and-their-protocol-stacks.html`. Online; accessed 28 November 2012. [cited at p. 46]

[50] Top Global, "SS7 Protocol Overview." `http://www.ss7-training.net/sigtran-training/ch04lev1sec4.html`. Online; accessed 1 January 2013. [cited at p. 46]

Appendices

Appendix A

Communication with people from different fields

A.1 List of people requested to provide information

Table A.1: Contacted people and their field

Name	Office	Country
Professor Dr. Do van Thanh	Telenor/NTNU	Norway
Bipin Timalsina	Ncell Private Ltd	Nepal
Binod Karki	Huawei Technologies Co. Ltd.	India
Håkon Styri	Norwegian Post and Telecommunications Authority.	Norway
Professor Jan Arild Audestad	Telenor/NTNU	Norway
Chinmay Anand	Nepal Telecom	Nepal
Dinesh Panthy	Nepal Telecom	Nepal

Appendix B

Configurations

Configuration files B.1: Sample of freeDiameter.conf file

```
# This is a sample configuration file for freeDiameter daemon.

# Only the "TLS_Cred" directive is really mandatory in this file.

##############################################################
##  Peer identity and realm

# The Diameter Identity of this daemon.
# This must be a valid FQDN that resolves to the local host.
# Default: hostname's FQDN
Identity = "client-VirtualBox.ed.ntnu.no";

# The Diameter Realm of this daemon.
# Default: the domain part of Identity (after the first dot).
#Realm = "koganei.freediameter.net";

##############################################################
##  Transport protocol configuration

# The port this peer is listening on for incoming connections (TCP
    and SCTP).
# Default: 3868
#Port = 3868;

# The port this peer is listening on for incoming TLS connections
   (TCP and SCTP).
# See TLS_old_method for more information.
# Default: 3869
#SecPort = 3869;

# Use RFC3588 method for TLS protection, where TLS is negociated
   after CER/CEA
```

```
# on the same port. This only affects outgoing connections. It can be overwritten
# on per peer basis.
# Default: use RFC3588bis method with separate port for TLS.
#TLS_old_method;

# Disable use of TCP protocol (only listen and connect in SCTP)
# Default : TCP enabled
#No_TCP;

# Disable use of SCTP protocol (only listen and connect in TCP)
# Default : SCTP enabled
#No_SCTP;
# This option has no effect if freeDiameter is compiled with DISABLE_SCTP option,
# in which case the value is forced to "SCTP disabled".

# Prefer TCP over SCTP for establishing new connections.
# It may be overwritten per peer in peer configuration blocs.
# Default : SCTP is prefered.
#Prefer_TCP;

# Default number of streams per SCTP associations.
# It can be overwritten per peer basis.
# Default : 30 streams
#SCTP_streams = 30;

##############################################################
##  Endpoints configuration

# Disable use of IP addresses (only IPv6)
# Default : IP enabled
#No_IP;

# Disable use of IPv6 addresses (only IP)
# Default : IPv6 enabled
#No_IPv6;

# Specify local addresses where the server must listen
# Default : listen on all addresses available.
#ListenOn = "202.249.37.5";
#ListenOn = "2001:200:903:2::202:1";
#ListenOn = "fe80::21c:5ff:fe98:7d62%eth0";

# Note: although by default freeDiameter listens also on the loopback interface, it
# will not be able to connect to the loopback address.

##############################################################
##  TLS Configuration

```

```
# TLS is managed by the GNUTLS library in the freeDiameter daemon.
# You may find more information about parameters and special behaviors
# in the relevant documentation.
# http://www.gnu.org/software/gnutls/manual/

# Credentials of the local peer
# The X509 certificate and private key file to use for the local peer.
# The files must contain PKCS-1 encoded RSA key, in PEM format.
# (These parameters are passed to gnutls_certificate_set_x509_key_file function)
# Default : NO DEFAULT
#TLS_Cred = "<x509 certif file.PEM>" , "<x509 private key file.PEM>";
TLS_Cred = "/home/client/thesis/CA/clientcerts/clicert.pem", "/home/client/thesis/CA/clientcerts/cliprivkey.pem";

# Certificate authority / trust anchors
# The file containing the list of trusted Certificate Authorities (PEM list)
# (This parameter is passed to gnutls_certificate_set_x509_trust_file function)
# The directive can appear several times to specify several files.
# Default : GNUTLS default behavior
TLS_CA = "/home/client/thesis/CA/clientcerts/clicert.pem";

# Certificate Revocation List file
# The information about revoked certificates.
# The file contains a list of trusted CRLs in PEM format. They should have been verified before.
# (This parameter is passed to gnutls_certificate_set_x509_crl_file function)
# Note: currently, openssl CRL seems not supported...
# Default : GNUTLS default behavior
#TLS_CRL = "<file.PEM>";

# GNU TLS Priority string
# This string allows to configure the behavior of GNUTLS key exchanges
# algorithms. See gnutls_priority_init function documentation for information.
# You should also refer to the Diameter required TLS support here:
#   http://tools.ietf.org/html/draft-ietf-dime-rfc3588bis-18#section-13.1
# Default : "NORMAL"
# Example: TLS_Prio = "NONE:+VERS-TLS1.1:+AES-128-CBC:+RSA:+SHA1:+COMP-NULL";
#TLS_Prio = "NORMAL";

# Diffie-Hellman parameters size
# Set the number of bits for generated DH parameters
# Valid value should be 768, 1024, 2048, 3072 or 4096.
# (This parameter is passed to gnutls_dh_params_generate2 function
   ,
# it usually should match RSA key size)
# Default : 1024
TLS_DH_Bits = 1024;

# Alternatively, you can specify a file to load the PKCS#3 encoded
# DH parameters directly from. This accelerates the daemon start
# but is slightly less secure. If this file is provided, the
# TLS_DH_Bits parameters has no effect.
# Default : no default.
TLS_DH_File = "/home/client/thesis/CA/clientcerts/dh.pem";


##############################################################
##  Timers configuration

# The Tc timer of this peer.
# It is the delay before a new attempt is made to reconnect a
   disconnected peer.
# The value is expressed in seconds. The recommended value is 30
   seconds.
# Default: 30
#TcTimer = 30;

# The Tw timer of this peer.
# It is the delay before a watchdog message is sent, as described
   in RFC 3539.
# The value is expressed in seconds. The default value is 30
   seconds. Value must
# be greater or equal to 6 seconds. See details in the RFC.
# Default: 30
#TwTimer = 30;

##############################################################
##  Applications configuration

# Disable the relaying of Diameter messages?
# For messages not handled locally, the default behavior is to
   forward the
# message to another peer if any is available, according to the
   routing
# algorithms. In addition the "0xffffff" application is advertised
    in CER/CEA
# exchanges.
# Default: Relaying is enabled.
#NoRelay;

```

```
# Number of server threads that can handle incoming messages at
    the same time.
#  TODO: implement dynamic # of threads depending on the length of
     the queue.
# Default: 4
#AppServThreads = 4;

# Other applications are configured by loading appropriate
    extensions.

##############################################################
##  Extensions configuration

#  The freeDiameter daemon merely provides support for
# Diameter Base Protocol. The specific application behaviors,
# as well as advanced functions of the daemon, are provided
# by loadable extensions (plug-ins).
#  These extensions may in addition receive the name of a
# configuration file, the format of which is extension-specific.
#
# Format:
#LoadExtension = "/path/to/extension" [ : "/optional/configuration
    /file" ] ;
#
# Examples:
#LoadExtension = "extensions/sample.fdx";
LoadExtension = "/home/client/thesis/freediameter/fDbuild/
    extensions/test_app.fdx":"/home/client/thesis/freediameter/
    fDbuild/conf/test_app.conf";


##############################################################
##  Peers configuration

#  The local server listens for incoming connections. By default,
# all unknown connecting peers are rejected. Extensions can
    override this behavior.
#
#  In addition to incoming connections, the local peer can
# be configured to establish and maintain connections to some
# Diameter nodes and allow connections from these nodes.
#  This is achieved with the ConnectPeer directive described
    bellow.
#
# Note that the configured Diameter Id MUST match
# the information received inside CEA, or the connection will be
    aborted.
#
# Note also, loopback addresses are not allowed currently in
    freeDiameter
# (because of a bad behavior if they are allowed).
# As a workaround, one might provide a public address of the local
    machine to
# test locally.
#
# Format:
#ConnectPeer = "diameterid" [ { parameter1; parameter2; ...} ] ;
# Parameters that can be specified in the peer's parameter list:
#  No_TCP; No_SCTP; No_IP; No_IPv6; Prefer_TCP; TLS_old_method;
#  No_TLS;       # assume transparent security instead of TLS
#  Port = 3868;  # The port to connect to
#  TcTimer = 30;
#  TwTimer = 30;
#  ConnectTo = "202.249.37.5";
#  ConnectTo = "2001:200:903:2::202:1";
#  TLS_Prio = "NORMAL";
#  Realm = "realm.net"; # Reject the peer if it does not advertise
    this realm.
# Examples:
#ConnectPeer = "aaa.wide.ad.jp";
#ConnectPeer = "old.diameter.serv" { TcTimer = 60; TLS_old_method;
    No_SCTP; } ;
ConnectPeer = "ubuntu.ubuntu-domain" { ConnectTo =
   "192.168.1.229"; NO_TLS; } ;

################################################################
```

B.1 Configurations in Server Side

Configuration files B.2: IPsec-tools.conf for Server

```
# Configuration of IPsec ESP for server(192.168.1.229)

# Flush the SAD and SPD
flush;
spdflush;


# AH SAs using 128 bit long keys
#add 192.168.1.229 192.168.1.196 ah 0x200 -A hmac-md5 0
   xdc8b6a45388289101c6fc1815d21b31d;
#add ip of other machine.
#add 192.168.1.196 192.168.1.229 ah 0x300 -A hmac-md5 0
   x368ab975d1191a4d242482c9f2599149;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add 192.168.1.229 192.168.1.196 esp 0x201 -E 3des-cbc 0
   xc43a117c6a124fadbfeea7894d6788cdfc81200691cb89f4;
add 192.168.1.196 192.168.1.229 esp 0x301 -E 3des-cbc 0
   x7fe856c581b8210a33ff10f5382e8ed7d6c698023cadd0cf;

# Security policies
```

```
spdadd 192.168.1.229 192.168.1.196 any -P out ipsec
           esp/transport//require;
           #ah/transport//require;

spdadd 192.168.1.196 192.168.1.229 any -P in ipsec
           esp/transport//require;
           #ah/transport//require;
```

Configuration files B.3: freeDiameter.conf for Server

```
# This is a configuration file for freeDiameter daemon in server.

Identity = "ubuntu.ubuntu-domain";

TLS_Cred = "/home/ashim/CA/server/cert.pem", "/home/ashim/CA/
    server/privkey.pem";

TLS_CA = "/home/ashim/CA/ca.pem";

TLS_DH_Bits = 1024;

TLS_DH_File = "/home/ashim/CA/server/dh.pem";

LoadExtension = "/home/ashim/freeDiameter/fDbuild/extensions/
    test_app.fdx":"/home/ashim/freeDiameter/fDbuild/conf/test_app.
    conf";


ConnectPeer = "client-VirtualBox.ed.ntnu.no" {ConnectTo =
    "192.168.1.196"; NO_TLS; } ;
```

Configuration files B.4: test_app.conf on Server

```
#######################
# This file contains the description of configuration and general
    information about the
# "App_test" extension.

# This extension provides a simple way to send a predefined
    message over the Diameter Network.
# It may be used to test the Routing or other base mechanisms from
     the Diameter network.

# In order to enable this extension, the main freeDiameter
    configuration file
# must contain the following declaration:
# LoadExtension = "extensions/app_test.fdx" : "/path/to/app_test.
    conf" ;
# Note that the conffile may be omitted, in which case default
    parameters will be assumed.
#######################

```

```
#######################
# Configuration of the test message

# This application is defined as a Vendor-Specific application.
# Since freeDiameter does not have a IANA-assigned Vendor ID, we let a configurable value here:
# vendor-id = 999999;

# The application id. Same remark as previously.
# appli-id = 999999;
# appli-id = 4;

# The command code for Test-Request and Test-Answer. The range 0xfffffe-ffffff (dec: 16777215) is reserved for experimental use.
 #cmd-id = 16777214;
#cmd-id = 318;


# The AVP id for the test.
# avp-id = 345678;
#avp-id = 258;

#######################
# Configuration of the extension behavior

# The mode for the extension.
# - server: Answer incoming requests. The signal is ignored.
# - client: Send a request when the signal is received, and measure the time to receiving answer.
# - both: acts as client and server
# mode = both;
mode = server;

# The behavior can be changed by specifying additional "benchmark;" keyword.
# When this keyword appears, it changes the behavior as follow:
#  - server is silent on message reception, only the activity summary is displayed every 30 seconds
#  - client attempts to send as many messages as possible during 10 seconds and counts them.
# The benchmark keyword can be followed optionaly by two integers:
#   duration is the time for the measurement, in seconds (default 10).
#   concurrency is the number of messages that can be on the wire before waiting for an answer (default 100).
# benchmark [duration concurrency];


#######################
```

```
# Client-specific configuration

# The Destination-Realm for the message
# (default is sending to same realm as local peer).
# dest-realm = "foreign.net";
#dest-realm = "ed.ntnu.no";

# The Destination-Host for the message.
# (default is not providing this AVP).
# dest-host = "server.foreign.net";

# The User-Name for the message (may be useful for some routing
    tests).
# (default is not providing this AVP).
# user-name = "user@server.foreign.net";

# The signal that triggers sending the test message
# Note: Symbolic names are now recognized, you must use integers
# signal = 10;
```

B.2 Configurations in Client Side

Configuration files B.5: IPsec-tools.conf for Client

```
        # Configuration for (192.168.1.196)

    # Flush the SAD and SPD
    flush;
    spdflush;

    # AH SAs using 128 bit long keys
   # add 192.168.1.229 192.168.1.196 ah 0x200 -A hmac-md5 0
       xdc8b6a45388289101c6fc1815d21b31d;
    #add ip of other machine.
    #add 192.168.1.196 192.168.1.229 ah 0x300 -A hmac-md5 0
        x368ab975d1191a4d242482c9f2599149;

    # ESP SAs using 192 bit long keys (168 + 24 parity)
    add 192.168.1.229 192.168.1.196 esp 0x201 -E 3des-cbc 0
        xc43a117c6a124fadbfeea7894d6788cdfc81200691cb89f4;
    add 192.168.1.196 192.168.1.229 esp 0x301 -E 3des-cbc 0
        x7fe856c581b8210a33ff10f5382e8ed7d6c698023cadd0cf;

    # Security policies
    spdadd 192.168.1.229 192.168.1.196 any -P in ipsec
    esp/transport//require;
    #ah/transport//require;

    spdadd 192.168.1.196 192.168.1.229 any -P out ipsec
    esp/transport//require;
#    ah/transport//require;
```

Configuration files B.6: freeDiameter.conf for Client

```
Identity = "client-VirtualBox.ed.ntnu.no";

TLS_Cred = "/home/client/thesis/CA/clientcerts/clicert.pem", "/
    home/client/thesis/CA/clientcerts/cliprivkey.pem";

TLS_CA = "/home/client/thesis/CA/clientcerts/clicert.pem";

TLS_DH_Bits = 1024;

TLS_DH_File = "/home/client/thesis/CA/clientcerts/dh.pem";

LoadExtension = "/home/client/thesis/freediameter/fDbuild/
    extensions/test_app.fdx":"/home/client/thesis/freediameter/
    fDbuild/conf/test_app.conf";


ConnectPeer = "ubuntu.ubuntu-domain" { ConnectTo =
    "192.168.1.229"; NO_TLS; } ;
##############################################################
```

Configuration files B.7: test_app.conf on Client

```
mode = client;

signal = 10;
```

Appendix C

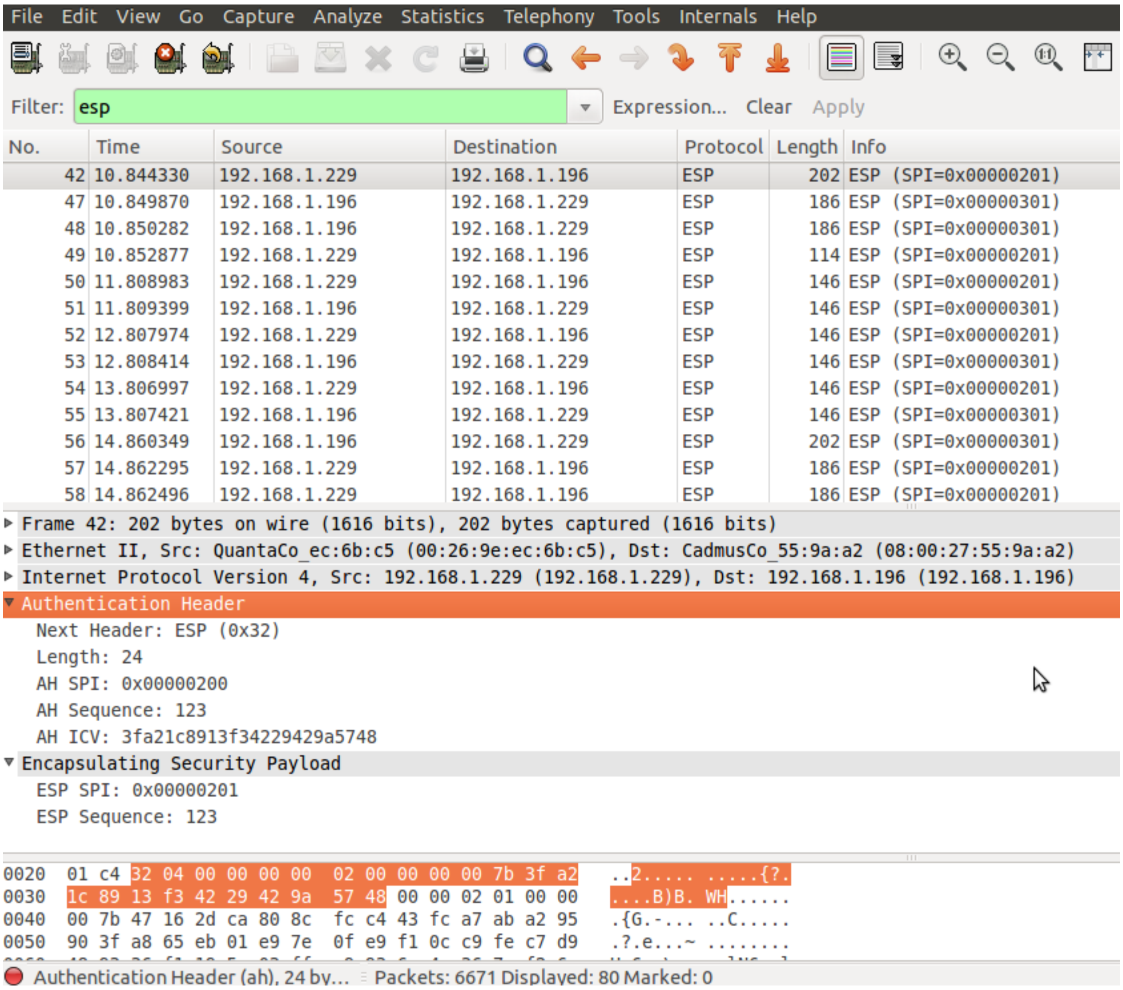
Outputs of Program

Outputs C.1: Behaviour observed in terminal of Server side

```
-------------------------------------------------------------------------------

ashim@ubuntu:~$ freeDiameterd -c /home/ashim/freeDiameter/fDbuild/
    conf/freediameter.conf.test.app
libfdproto initialized.
libgnutls '2.10.5' initialized.
Loading : /home/ashim/freeDiameter/fDbuild/extensions/test_app.fdx
Extension Test_App initialized with configuration: '/home/ashim/
    freeDiameter/fDbuild/conf/test_app.conf'
------- app_test configuration dump: ---------
 Vendor Id .......... : 999999
 Application Id ..... : 16777215
 Command Id ......... : 16777214
 AVP Id ............. : 16777215
 Mode ............... : Serv
 Destination Realm .. : ubuntu-domain
 Destination Host ... : - none -
 Signal ............. : 10
------- /app_test configuration dump ---------
All extensions loaded.
-- Configuration :
  Debug trace level ...... : +1
  Configuration file ..... : /home/ashim/freeDiameter/fDbuild/conf
      /freediameter.conf.test.app
  Diameter Identity ...... : ubuntu.ubuntu-domain (l:20)
  Diameter Realm ......... : ubuntu-domain (l:13)
  Tc Timer ............... : 30
  Tw Timer ............... : 30
  Local port ............. : 3868
  Local secure port ...... : 3869
  Number of SCTP streams . : 30
  Number of server threads : 4
  Local endpoints ........ : Default (use all available)
  Local applications ..... : App: 16777215  Au--  Vnd: 999999
  Flags : - IP .......... : Enabled
          - IPv6 ......... : Enabled
          - Relay app .... : Enabled
          - TCP .......... : Enabled
          - SCTP ......... : Enabled
          - Pref. proto .. : SCTP
          - TLS method ... : Separate port
  TLS :   - Certificate .. : /home/ashim/CA/server/cert.pem
          - Private key .. : /home/ashim/CA/server/privkey.pem
          - CA (trust) ... : /home/ashim/CA/ca.pem (2 certs)
          - CRL .......... : (none)
          - Priority ..... : (default: 'NORMAL')
          - DH file ...... : /home/ashim/CA/server/dh.pem
  Origin-State-Id ........ : 1358407585
freeDiameterd daemon initialized.
Unable to connect to the peer client-VirtualBox.ed.ntnu.no,
   aborting attempts for now.
```

Outputs C.2: Behaviour observed in terminal of Client side

```
client@client-VirtualBox:~$ freeDiameterd -c /home/client/thesis/
   freediameter/fDbuild/conf/freediameter.conf
libfdproto initialized.
libgnutls '2.10.5' initialized.
Loading : /home/client/thesis/freediameter/fDbuild/extensions/
   test_app.fdx
Extension Test_App initialized with configuration: '/home/client/
   thesis/freediameter/fDbuild/conf/test_app.conf'
------- app_test configuration dump: ---------
 Vendor Id .......... : 999999
 Application Id ..... : 16777215
 Command Id ......... : 16777214
 AVP Id ............. : 16777215
 Mode ............... : Cli
 Destination Realm .. : ubuntu-domain
 Destination Host ... : - none -
 Signal ............. : 10
------- /app_test configuration dump ---------
All extensions loaded.
-- Configuration :
  Debug trace level ...... : +1
  Configuration file ..... : /home/client/thesis/freediameter/
     fDbuild/conf/freediameter.conf
  Diameter Identity ...... : client-VirtualBox.ed.ntnu.no (l:28)
  Diameter Realm ......... : ed.ntnu.no (l:10)
  Tc Timer ............... : 30
  Tw Timer ............... : 30
  Local port ............. : 3868
  Local secure port ...... : 3869
  Number of SCTP streams . : 30
  Number of server threads : 4
  Local endpoints ........ : Default (use all available)
  Local applications ..... : App: 16777215  Au--  Vnd: 999999
  Flags : - IP ........... : Enabled
          - IPv6 ......... : Enabled
          - Relay app .... : Enabled
          - TCP .......... : Enabled
          - SCTP ......... : Enabled
          - Pref. proto .. : SCTP
          - TLS method ... : Separate port
  TLS :   - Certificate .. : /home/client/thesis/CA/clientcerts/
     clicert.pem
          - Private key .. : /home/client/thesis/CA/clientcerts/
             cliprivkey.pem
          - CA (trust) ... : /home/client/thesis/CA/clientcerts/
             clicert.pem (1 certs)
          - CRL .......... : (none)
          - Priority ..... : (default: 'NORMAL')
          - DH file ...... : /home/client/thesis/CA/clientcerts/dh
             .pem
  Origin-State-Id ........ : 1358407679
freeDiameterd daemon initialized.
Connection established to server '[ubuntu.local]:3868' (SCTP:12,
   30/30 streams).
Sent to 'SCTP to [192.168.1.229]:3868 (12)'
Logged: 01/17/13,08:27:59.421656

 |MSG: 0x8cec830
 |   model : v/m:R---/RPET, 257 "Capabilities-Exchange-Request"
 |   public: V:1 L:216 fl:R--- CC:257 A:0 hi:204a16 ei:7ff00000
 |   intern: rwb:(nil) rt:0 cb:(nil)((nil)) qry:(nil) asso:0 sess
    :(nil) src:(nil)(0)
Received 228b from 'ubuntu.ubuntu-domain' (STATE_WAITCEA)
Logged: 01/17/13,08:27:59.425827

 |MSG: 0x8cecee8
 |   (no model)
 |   public: V:1 L:228 fl:---- CC:257 A:0 hi:204a16 ei:7ff00000
 |   intern: rwb:0x8ced348 rt:0 cb:(nil)((nil)) qry:(nil) asso:0
    sess:(nil) src:(nil)(0)
No TLS protection negotiated with peer 'ubuntu.ubuntu-domain'.
'STATE_WAITCEA' -> 'STATE_OPEN' 'ubuntu.ubuntu-domain'
Received 84b from 'ubuntu.ubuntu-domain' (STATE_OPEN)
Logged: 01/17/13,08:27:59.430908
```



Filter: esp

```
No.  Time      Source         Destination    Protocol Length Info
  42 10.844330 192.168.1.229  192.168.1.196  ESP         202 ESP (SPI=0x00000201)
  47 10.849870 192.168.1.196  192.168.1.229  ESP         186 ESP (SPI=0x00000301)
  48 10.850282 192.168.1.196  192.168.1.229  ESP         186 ESP (SPI=0x00000301)
  49 10.852877 192.168.1.229  192.168.1.196  ESP         114 ESP (SPI=0x00000201)
  50 11.808983 192.168.1.229  192.168.1.196  ESP         146 ESP (SPI=0x00000201)
  51 11.809399 192.168.1.196  192.168.1.229  ESP         146 ESP (SPI=0x00000301)
  52 12.807974 192.168.1.229  192.168.1.196  ESP         146 ESP (SPI=0x00000201)
  53 12.808414 192.168.1.196  192.168.1.229  ESP         146 ESP (SPI=0x00000301)
  54 13.806997 192.168.1.229  192.168.1.196  ESP         146 ESP (SPI=0x00000201)
  55 13.807421 192.168.1.196  192.168.1.229  ESP         146 ESP (SPI=0x00000301)
  56 14.860349 192.168.1.196  192.168.1.229  ESP         202 ESP (SPI=0x00000301)
  57 14.862295 192.168.1.229  192.168.1.196  ESP         186 ESP (SPI=0x00000201)
  58 14.862496 192.168.1.229  192.168.1.196  ESP         186 ESP (SPI=0x00000201)
```

```
▶ Frame 42: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits)
▶ Ethernet II, Src: QuantaCo_ec:6b:c5 (00:26:9e:ec:6b:c5), Dst: CadmusCo_55:9a:a2 (08:00:27:55:9a:a2)
▶ Internet Protocol Version 4, Src: 192.168.1.229 (192.168.1.229), Dst: 192.168.1.196 (192.168.1.196)
▼ Authentication Header
    Next Header: ESP (0x32)
    Length: 24
    AH SPI: 0x00000200
    AH Sequence: 123
    AH ICV: 3fa21c8913f34229429a5748
▼ Encapsulating Security Payload
    ESP SPI: 0x00000201
    ESP Sequence: 123
```

```
0020  01 c4 32 04 00 00 00 00  02 00 00 00 00 7b 3f a2   ..2..... .....{?.
0030  1c 89 13 f3 42 29 42 9a  57 48 00 00 02 01 00 00   ....B)B. WH......
0040  00 7b 47 16 2d ca 80 8c  fc c4 43 fc a7 ab a2 95   .{G.-... ..C.....
0050  90 3f a8 65 eb 01 e9 7e  0f e9 f1 0c c9 fe c7 d9   .?.e...~ ........
```

Authentication Header (ah), 24 by... Packets: 6671 Displayed: 80 Marked: 0

Figure C.1: Wireshark Snapshot showing IPsec headers.