Enhancing Security and Privacy in VoIP/IMS Environments

Nikos Vrakas

University of Piraeus

Systems Security Lab
Department of Digital Systems
University of Piraeus

Ph.D. Thesis

April 2013

Advisory Committee

Costas Lambrinoudakis, Assistant Professor (Supervisor)
University of Piraeus

Sokratis Katsikas, Professor
University of Piraeus

George Vassilacopoulos, Professor
University of Piraeus

Examination Committee

George Vassilacopoulos, Professor
University of Piraeus

Sokratis Katsikas, Professor
University of Piraeus

Spiridon Likothanassis, Professor
University of Patras

Gritzalis Stefanos, Professor
University of the Aegean

Costas Lambrinoudakis, Assistant Professor
University of Piraeus

Christos Xenakis, Assistant Professor
University of Piraeus

Georgios Kambourakis, Assistant Professor
University of the Aegean

Abstract

During the last decade, the development of technology has made Voice Over Internet Protocol (VoIP) and other multimedia services feasible even for mobile devices. Internet users have been gradually moving away from convectional computers since they can access such services from everywhere, through their mobile phones. The convergence of all these heterogeneous networks is achieved through the deployment of the IP Multimedia Subsystem (IMS). This platform is the main signaling component in next generation networks and can bring together all different types of networks under a common, all-IP, architecture.

As the main signaling component, the IMS is responsible for the session establishment and management of the provided multimedia services, such as internet telephony, video and call conferences and many more, by utilizing the session initiation protocol (SIP).

The variety of different networks and protocols, that take part in this architecture, introduce threats, vulnerabilities and new possibilities for attacks. The technical specifications of IMS cannot sufficiently prevent many security incidents while SIP has been proven vulnerable to many malicious acts.

This thesis is concerned with security and privacy issues in SIP based communication services such as the ones provided by VoIP/IMS infrastructures. All the vulnerabilities that can be exploited and violate confidentiality, integrity, availability and privacy requirements are presented and evaluated considering many different factors such as the time frame of the attack, the access level of the attacker and the security mechanisms employed. Since signaling messages' integrity can be protected with various methods, this research has been focused on preserving architecture's availability, messages' authenticity and users' privacy.

The development of a flooding attack detection mechanism has been the first objective of this research since such attacks pose serious threats to these environments. A metric, that is derived by modeling the session establishment procedure, has been proposed for detecting incomplete SIP connections. By applying simple mathematical computations, single and distributed source flooding attacks can be detected.

Afterwards, a more advanced framework that provides prevention of both SIP signaling and flooding attacks has been introduced. The content of packets that encapsulate SIP signaling is analyzed and the information gathered are bound with SIP authentication. The generated tuples are stored into tables with the support of the Bloom filters. Messages' manipulation can be detected by analysing the gathered data while a statistical model has been developed in order to detect deviations from devices' normal traffic behaviour.

Concerning privacy requirements, a mechanism that protects users's anonymity has been introduced. The unprotected users' identity is concealed by utilizing commutative functions. A new one-time identity is generated for every new session providing also unlinkability services.

All the developed mechanisms have been implemented and evaluated through numerous different scenarios. The experimental results prove the proposed mechanisms' efficiency while they demonstrate that only negligible overheads are introduced.

Περίληψη

Κατά την τελευταία δεκαετία, η πρόοδος της τεχνολογίας έκανε εφικτή τη χρήση υπηρεσιών διαδικτυακής τηλεφωνίας (VoIP – Voice over IP) και άλλων πολυμεσικών εφαρμογών, μέσω κινητών συσκευών. Οι χρήστες του διαδικτύου απομακρύνονται σταδιακά από τους συμβατικούς υπολογιστές αφού μπορούν να έχουν πρόσβαση σε τέτοιες υπηρεσίες από παντού, κάνοντας χρήση των κινητών τους τηλεφώνων. Η ενοποίηση όλων αυτών των ετερογενών δικτύων κάτω από μία κοινή αρχιτεκτονική, βασισμένη εξ' ολοκλήρου στο πρωτόκολλο IP, επιτυγχάνεται με την χρήση του Υποσυστήματος Πολυμέσων IP (IMS – IP Multimedia Subsystem).

Το IMS, ως κύριο σύστημα σηματοδοσίας των δικτύων επόμενης γενεάς, είναι υπεύθυνο για την εγκαθίδρυση συνόδων και τη διαχείριση των παρεχόμενων πολυμεσικών υπηρεσιών (όπως η διαδικτυακή τηλεφωνία, η τηλεδιάσκεψη μέσω εικόνας και ήχου και πολλές άλλες), αξιοποιώντας το πρωτόκολλο σηματοδοσίας SIP (Session Initiation Protocol).

Η ποικιλία των διαφορετικών δικτύων και πρωτοκόλλων που συμμετέχουν σε αυτή την αρχιτεκτονική, έχουν σαν αποτέλεσμα τη δημιουργία νέων ευπαθειών και νέου τύπου επιθέσεων. Οι τεχνικές προδιαγραφές του IMS δεν καλύπτουν επαρκώς τις νέες αυτές απαιτήσεις ασφάλειας, δεδομένου μάλιστα ότι το SIP έχει αποδειχθεί ευάλωτο σε αρκετές κακόβουλες ενέργειες.

Η παρούσα διατριβή μελετά τα ζητήματα ασφάλειας και ιδιωτικότητας στις τηλεπικοινωνιακές υπηρεσίες που είναι βασισμένες στο πρωτόκολλο SIP, όπως αυτές που προσφέρονται από τις υποδομές VoIP/IMS. Όλες οι αδυναμίες που μπορεί να αξιοποιήσει μια επίθεση για να παραβιάσει τις απαιτήσεις εμπιστευτικότητας, ακεραιότητας, διαθεσιμότητας και ιδιωτικότητας, παρουσιάζονται και αξιολογούνται λαμβάνοντας υπόψη διάφορους παράγοντες όπως το χρονικό διάστημα της επίθεσης, το επίπεδο πρόσβασης του επιτιθέμενου και τους υπάρχοντες μηχανισμούς ασφαλείας. Εφόσον η ακεραιότητα των μηνυμάτων σηματοδοσίας μπορεί να προστατευτεί με διάφορες μεθόδους, η διατριβή αυτή εστιάζει στην προστασία της διαθεσιμότητας της αρχιτεκτονικής, στην αυθεντικότητα των μηνυμάτων και στην προστασία της ιδιωτικότητας των χρηστών.

Η ανάπτυξη ενός μηχανισμού εντοπισμού επιθέσεων πλημμύρας αποτέλεσε έναν από τους αρχικούς στόχους της έρευνας, δεδομένου μάλιστα ότι επιθέσεις του τύπου αυτού μπορούν να επιφέρουν σημαντικές συνέπειες σε τέτοια περιβάλλοντα. Μία μετρική, η οποία προκύπτει από την μοντελοποίηση της διαδικασίας εγκαθίδρυσης συνόδου, προτάθηκε για τον εντοπισμό ημιτελών SIP συνδέσεων. Με την εφαρμογή απλών μαθηματικών υπολογισμών, γίνεται εφικτός ο εντοπισμός επιθέσεων πλημμύρας, τόσο ενός όσο και πολλαπλών γεννητόρων.

Στη συνέχεια παρουσιάζεται ένα εξελιγμένο πλαίσιο προστασίας από επιθέσεις πλημμύρας και σηματοδοσίας ταυτόχρονα. Το περιεχόμενο των πακέτων που ενθυλακώνουν σηματοδοσία SIP αναλύεται και οι πληροφορίες που συλλέγονται, συνδέονται με την αυθεντι-

κοποίηση του πρωτοκόλλου SIP. Οι παραγόμενες πλειάδες αποθηκεύονται σε πίνακες με τη βοήθεια Bloom φίλτρων. Η παραποίηση των μηνυμάτων μπορεί να εντοπιστεί αναλύοντας τα δεδομένα που έχουν συλλεχθεί, ενώ ένα στατιστικό μοντέλο έχει αναπτυχθεί προκειμένου να εντοπισθούν οι αποκλίσεις από την κανονική συμπεριφορά της δικτυακής κίνησης κάθε συσκευής.

Αναφορικά με τις απαιτήσεις ιδιωτικότητας, παρουσιάζεται ένας μηχανισμός που υποστηρίζει την ανωνυμία των χρηστών. Συγκεκριμένα, οι ταυτότητες των χρηστών προστατεύονται με την αξιοποίηση αντιμεταθετικών συναρτήσεων. Μια νέα, μίας χρήσης, ταυτότητα παράγεται για κάθε νέα σύνοδο, επιτυγχάνοντας έτσι μη συνδεσιμότητα.

Όλοι οι προτεινόμενοι μηχανισμοί υλοποιήθηκαν και αξιολογήθηκαν μέσω πολυάριθμων σεναρίων. Τα πειραματικά αποτελέσματα αποδεικνύουν τόσο την αποδοτικότητα των μηχανισμών όσο και την αμελητέα επιβάρυνση που εισάγουν στο σύστημα.

Acknowledgements

At this moment, I would like to thank the people who contributed greatly to the success of this research. I would firstly like to thank my supervisor Costas Lambrinoudakis for his guidance, scientific and moral support during my research. His valuable advices has not been only inspirational but also determinant in this research.

Additionally, I would like to thank Dr. Dimitris Geneiatakis (David) for his invaluable contribution and support to the completion of this work.

Appreciation also goes to the members of my advisory committee, Professors Sokratis Katsikas and George Vassilacopoulos for their directions and significant advices for keeping me up with the ph.d's objectives.

My special thanks go to my friends and colleagues Dr. Panos and Andreas Brezas for their care and moral support and also for their accurate comments and advises.

I am indebt to my colleagues Eleni Darra, Dimitra Georgiou and Nikos Pitropakis for providing a pleasant and fun environment, full of interesting discussions. I wish them to complete their research with success.

This ph.d. would not have been possible without the invaluable support of my family. Their love and encouragement has given me strength and inspiration throughout my research.

Contents

List of Figures

List of Tables

Nomenclature

Variables, Functions & Constants

δ	system capabilities factor
λ	number of hash functions
$A1, A2$	digest authentication functions
C_i	a counter in bloom filter
$Curr.C\_Sum$	current sum of incoming requests
$Curr.D\_Avg$	current distage average
$dist$	session distance metric
E	encryption algorithm
H, h	hash function
$Init.D$	initial distance
$Init.D\_Avg$	initial distance average
k	number of executed training procedures
keu	user equipment key
$kproxy$	proxy key
Nd	network delay
p	prime number
q	number of non-empty bloom registers
r	message string
$T.Dist$	distance between consequent messages
$T1, T2$	time intervals
T_{alarm}	distributed source threshold
T_{sd}	average of session distance
tr	tolerance rate

Trs, $T_{single[x]}$	threshold value for single source flooding
URT	user response time
DSalarm[x]	distributed source alarm
Init.C_Sum	initial sum of incoming requests
SSalarm	single source alarm
TS	timestamp

Sets

$\emptyset$	empty set
$\mathbb{N}$	set of natural numbers
A	set of SIP methods
D	set of identities
$E_{[i]}$	set of methods, identities and addresses
I	set of IP addresses
$K_{[i]}$, $F_{[i]}$	a subset of $E_{[i]}$
M	set of MAC addresses
O	finite union of $K_{[i]}$
S	set of SIP-IP addresses
W	finite union of $E_{[i]}$

Operators

$\approx$	approximately equal
$\bigcup_{i=1}^{n}$	union of n sets
$\cap$	intersection
$\cup$	union
$\forall$	for all
$\in$	exists in
$\leq$	less or equal
$\longrightarrow$	points to
$\neq$	not equal
$\notin$	not exists
$\rightarrow$	set value

$\rightarrowtail$	send to
$\subset$	subset
$\sum_{i=1}^{m}$	summation of m elements
$\sim$	an operation

Interfaces

Cx	Reference point between a CSCF and an HSS
Dh	Reference point between an AS and an SLF
Dx	Reference point between an I-CSCF and an SLF
Gm	Reference point between a UE and P-CSCF
Gq	Reference point between a P-CSCF and a PDF
Ici	Reference point between an IBCF and another external IBCF
ISC	Reference point between a CSCF and an Application Server
Mi	Reference point between a CSCF and a BGCF
Mj	Reference point between a BGCF and an MGCF
Mn	Reference point between a MGCF and IMS-MGW
Mr	Reference point between a CSCF and an MRFC
Mr′	Reference point between an AS and an MRFC for session control
Mw	Reference point between CSCFs
Sh	Reference point between an AS and a HSS

Acronyms

3GPP	Third Generation Partnership Project
ACF	Admission Confirmation
AKA	Authentication And Key Agreement
AMPS	Advanced Mobile Phone System
ARP	Address Resolution Protocol
ARQ	Admission Request
AS	Application Server
AUTN	Authentication Token
AV	Authentication Vector
BGCF	Breakout Gateway Control Function
CAMEL	Customized Applications For Mobile Network Enhanced Logic

CDR	Charging Data Records
CI	Confidence Interval
CK	Confidentiality Key
CLF	Connectivity Session Location And Repository Function
CPS	Call Per Second
CR	Constant Rate Flooding Attack
CS	Circuit Switched
CSCF	Call Session Control Functions
D-H	Diffie Hellman
DDoS	Distributed DoS
DHCP	Dynamic Host Configuration Protocol
DLP	Discrete Logarithm Problem
DNS	Domain Name System
DoS	Denial Of Service
DSL	Digital Subscriber Line
EDGE	Enhanced Data rates for GSM Evolution
ENUM	Telephone Number Mapping
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GCF	Gatekeeper Confirmation
GGSN	Gateway Gprs Support Node
GRUU	Globally Routable User Agent URI
GSM	Global System for Mobile Telecommunications
HN	Home Network
HSS	Home Subscriber Server
I-CSCF	Interrogating-CSCF
IBCF	Interconnection Border Control Function
ID	Identity
iFC	Initial Filter Criteria
IK	Integrity Key

IM CN	IMS Core Network
IM-SSF	IP Multimedia Switching Function
IMC	IMS Credentials
IMSI	International Mobile Subscriber Identity
IPv6	Internet Protocol Version 6
IR	Increasing Rate Flooding Attack
IRR/Irr	Include Register Request/Response
ISIM	IP Multimedia Service Identity Module
LCG	Legal Call Generator
LIA	Location-Info Answer
MCC	Mobile Country Code
MCG	Malicious Call Generator
MCU	Multipoint Control Units
MGCF	Media Gateway Control Function
MiM	Man In The Middle
MNC	Mobile Network Code
MRF	Media Resource Function
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Processor
MSIN	Mobile Subscriber Identification Number
NAPTR	Naming Authority Pointer
NGN	Next Generation Networks
O/S	Operating System
OSA	Open Service Access
OSA-SCS	Open Service AccessService Capability Server
OSI	Open Systems Interconnection
P-CSCF	Proxy-CSCF
P-GRUU	Public GRUU
PDF	Policy Decision Function
PI	Private Identity
PKI	Public Key Infrastructure

PLMN	Public Land Mobile Network
PM-C	Client Side Privacy Module
PM-S	Server Side Privacy Module
PSI	Public Service Identity
PSTN	Public Switched Telephone Network
PU	Public Identity
QoS	Quality of Service
RAND	Random
RCF	Registration Confirmation
RG	Registration Generator
ROC	Receiver Operating Characteristic
RTP	Real-time Transport Protocol
S-CSCF	Serving Call Session Control Function
SA	Security Association
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SGW	Signaling Gateway
SIM	Subscriber Identity Module
SIP	Session Initation Protocol
SLF	Subscription Locator Function
SPI	Security Parameter Index
SPTs	Service Point Triggers
SRTP	Secure RTP
T-GRUU	Temporary GRUU
TCP	Transmission Control Protocol
THIG	Topology Hiding Gateway
TP	True Positive
TPR	True Negative Rate
UA	User Agent
UDP	User Datagram Protocol
UE	User Equipment

UICC	Universal Integrated Circuit Card
UID	User Identity
URI	Uniform Resource Identifier
USIM	Universal SIM
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP

Definitions

Anonymity: The user is not identifiable by an attacker within a set of other users.

Data Confidentiality: The transmitted and stored data must be protected from unauthorized disclosure. Nobody but specific users (involved in the communication) and the core network can have access to the transmitted signaling data. Only specific core network elements can access the HSS database (stored data).

Data Integrity: The transmitted and stored data must be protected from unauthorized modification. Thus, unauthorized entities should not be allowed to modify the information exchanged among the communicating entities, neither the information stored in the HSS database.

Entity Authentication: Users and network devices must be able to prove the validity of their identities. Every authentication mechanism must provide the means in order for a UE/user to be able to authenticate itself/himself to the network and vice-versa (mutual authentication).

Pseudonymity: The utilization of a pseudonym as an identifier instead of the user's real identity.

Service Availability: The provided services must be available to the users at any time.

Unlinkability: An attacker cannot relate messages, actions and users with each other.

Users' Privacy: The users' private information shall not be disclosed. This information may involve actions, locations, identities and personal data associated with a specific person.

Chapter 1

Introduction

1.1 The IP Multimedia Subsystem Concept

The rapid growth of internet during the last decade comprises a fact that even the most doubtful could foresee. The usability, accessibility, low cost and the high volume of available information are the main reasons for its development. So, inevitably it became a part of users' daily life. The utilization of wireless networks, smart phones and generally mobile devices has given the opportunity for accessing internet services almost from everywhere. Voice, data, instant messaging, e-mail, internet television and other multimedia applications are examples of the supported services.

On the other hand, voice services provided through the conventional telephony networks such as the cellular or the Public Switched Telephone Network (PSTN), are not capable of providing low cost multimedia services of decent quality, in order to cover users' needs. Nowadays, users require services similar to the ones that the internet can provide (voice, data, multimedia applications, etc.).

The provision of such services among different networks has triggered the need of an all-Internet Protocol (IP) [1] network architecture. Under this context, the Third Generation Partnership Project (3GPP) has designed the IP Multimedia Subsystem (IMS) [2] that standardizes and specifies the appropriate mechanisms to guarantee the Quality of Service (QoS), security, availability, and reliability of such services.

Specifically, IMS is the main component in 3rd (3G) and Next Generation Netwroks (NGNs). This architecture achieves the integration of heterogeneous networks (e.g. cellular, data networks) in order to provide telecommunication services in a platform fully based on IP protocol (See Fig. 1.1). In addition, it is responsible for the management and control of all multimedia connections (video conferencing, Internet telephony, etc.) by utilizing the Session Initiation Protocol (SIP) [3]. The latter was decided as the signaling protocol by the 3GGP due to its flexibility in incorporating new services. Therefore, new headers and methods were introduced to SIP, towards the implementation of complex and contemporary services such as video conferencing. The user devices involved in IMS infrastructures may vary. Devices such as

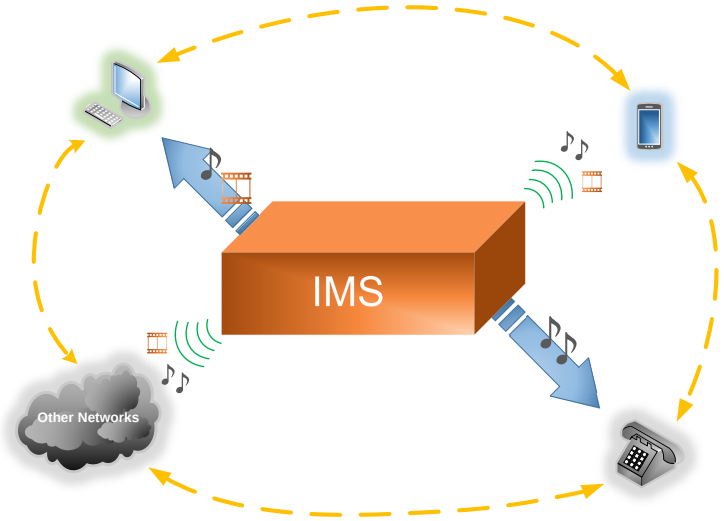


Figure 1.1: The IMS concept.

landline and low-end mobile phones can communicate with high-end smartphones, tablet devices and laptops.

1.1.1 The Evolution

The first type of cellular mobile telephony available to users was introduced in 1979. Bell Labs developed the Advanced Mobile Phone System (AMPS) [4], which comprises the first generation networks (1G). This analog telephone system could provide the users with only voice services while data was not supported. The digital telephony emerged later, known as second generation (2G), and provided better sound quality and higher capacity though the circuit-switched domain. Systems such as the Global System for Mobile Telecommunications (GSM) provided call hold, waiting and forwarding, caller identification and others.

While the telecommunication operators could not wait for the standardization of the 3rd generation networks, they added an overlay packet network to GSM to support IP data transmission. In 1997, the 2.5G networks were established with the General Packet Radio Service (GPRS) [5] and later, in 1999, emerged the General Packet Radio Service Enhanced Data rates for GSM Evolution (EDGE) [6]. User could have better video services, e-mail, instant messages and internet over the IP network with data rates up to 144kbit/s.

The native IP technology for mobile phone devices was introduced in 3G networks. This native IP support enables much faster data transmission rates, comparable to Digital Subscriber Line (DSL) ones. Video and VoIP calls, call conferences, instant messaging and presence information are examples of the services offered in 3G.

A major upgrade of 3G came along with the employment of an overlay architecture, the IMS. This architecture has enabled the provision of even higher quality of audio/video streaming, and VoIP calls, while the video and call conference services are now feasible (it was unstable, fragile and of low quality until now).

The main difference between VoIP, initial 3G networks and IMS infrastructures is the QoS and bandwidth management procedures. Particularly, the QoS in IMS can be adjusted in order to meet the specific device's capabilities. The session establishment procedures have been extended with resource reservation functions that enable the devices to determine the appropriate QoS level. Such an improvement in resource management gave the opportunity for new service provision such as the online multiparty gaming, real time video streaming, video on demand and many more. Further, state of the art security mechanism are described in the IMS specifications [7] while VoIP's security directions are considered rather loose.

The major IMS advantages are summarized in the following:

- Ensures the quality of the provided services
- Supports different network architectures
- Facilitates roaming procedures
- Wide range of new services
- Enhanced Security services and policies
- Charging functions and accounting

1.1.2 Contemporary Services with Old Threats

This contemporary form of communication arouses users' and service provider's interest at a significantly growing rate. The main rival to IMS deployments is not other upcoming architectures but only the security and privacy concerns that evolve. Since it is an IP based architecture, IMS applications inherit all the vulnerabilities associated with IP. Moreover, all the vulnerabilities that exist in other supported networks (e.g. WiMax, GSM, VoIP, GPRS, xDSL, etc.) threaten the IMS while the attackers can have access from one network to the other. The enormous volume of specifications introduced by the 3GPP is hardly manageable and even though it may provide a better security level, comparing to other network infrastructures, it is not adequate to protect users' privacy and prevent many security threats that originate from signaling protocol's vulnerabilities.

The security requirements in these environments as well as in most information management systems are based on the statement [8]:

"An important requirement of any information management system is to protect data and resources against unauthorized disclosure (secrecy) and unauthorized or improper modifications (integrity), while at the same time ensuring their availability to legitimate users (no denials-of-service)."

In such architectures however, two additional requirements are also equally important: the privacy of the users and the accountability of their transactions.

1.2 Problem Statement and Significance

Users desire reliable, secure, private and always available services from the telecommunication service providers. However, the utilization of open architectures and publicly accessible services, introduces numerous security and privacy concerns that are mainly originated from the vulnerabilities of the participating networks and utilized protocols.

1.2.1 Security Concerns

The main objective of IMS is to provide high quality services among different type of networks in a catholic all-IP networking infrastructure. The convergence of many different types of networks, render IMS an open architecture vulnerable to various threats. Malicious users do not only have adopted attacking methods from the lower layers of the internet protocol stack but they develop new ones, equally destructive and harmful. In addition to the vulnerabilities inherited form the underlaying protocols, attackers may exploit vulnerabilities from the architecture's main protocols. For instance, the clear text format of SIP's (the main signaling protocol in VoIP/IMS environments) messages, discloses crucial information about the communicating entities. Such data may reveal the users' location, identities and session's parameters. All these information can be utilized by malicious users in order to launch attacks that threatens the quality of the provided services and raises privacy concerns. Moreover, the interconnection of VoIP/IMS networks with PSTN introduces new threats. The closed PSTN architecture tends to become open, due to the capability of reaching PSTN networks through the deployment of gateway servers. Thus, the attacker can now intrude in traditional telephone networks and can render a communication impossible even in areas that are considered relatively safe.

Unlike other applications, voice services are very sensitive to variations in network's response times. Even a relatively small amount of delay is not acceptable. For example, when downloading a file via File Transfer Protocol (FTP), a delay of a few seconds is considered negligible, while in contrast a delay of 200 milliseconds in VoIP networks can constitute a communication fragile [9]. Such a network can be easily led to Denial of Service (DoS) conditions. On top of that, the low resource users' devices comprise an easy target for the attackers. Thus, the DoS issues that may rise, concerning home network's and/or user's devices, comprise a serious threat for voice transmissions and consequently for the architecture's availability and QoS.

1.2.2 Privacy Concerns

Protecting users' privacy is rather complicated due to the conflicting interests of the parties involved. Let's consider, for example, the provision of on-line personal services. Everyone wants to preserve his privacy, but no one says "no" to Amazon's excellent suggestions in books, nor to the exact results provided by the Google search engine, services that have been so successful due to continuous processing of personal information of previous users of the service. Information, insignificant until yesterday, may have tremendous value tomorrow.

Concerning IMS/VoIP environments, users are linked with their subscriptions with public and private identities. The disclosure of private information by a malicious user who "listens" the communication channel (since the SIP signaling is in clear text) seems inevitable. Such information is adequate in order to link the participants in a media session with other users, services, data and actions violating anonymity, unlinkability and unobservability requirements. Such a violation of users' privacy dramatically degrades the operator's reputation and exposes valuable and private information to entities that may use them at the expense of the users.

1.3 Motivation

Internet protocol has not been designed with security in mind at all and, consequently, malicious users exploit this fact in order to launch attacks. There are many security vulnerabilities that can be exploited by malicious internal or external users in order to degrade the QoS (causing denial of service), intercept the communication sessions, steal user's identities, credentials and impersonate network entities [10].

Moreover, the attacker can utilize techniques from the lower layers of the internet protocol stack in order to threaten SIP based services. For instance, IP spoofing [11] or Address Resolution Protocol (ARP) [12] poisoning can be the first step of an attacker aiming to manipulate a SIP request. Every architecture that utilizes SIP as its signaling protocol is susceptible to such behaviors. Many scientific works pinpoint these vulnerabilities [13–15].

One might assume, that the IMS infrastructures are adequately secured by the utilization of robust authentication schemes and state-of-the-art security mechanisms, such as Authentication and Key Agrement (AKA) with IP Secure (IPSec) [16] and Transport Layer Security (TLS) [17]. However, they cannot always prevent the malicious will of an internal, and otherwise legitimate, user who has a legal subscription to the service. In such a case the attacker may bypass the security mechanisms because he is able to authenticate the forged and malicious messages, or he may utilize his legitimate security tunnel in order to launch identity theft, SIP signaling and flooding attacks [18, 19](see Section 5.4). Also, attackers may exploit the fact that the very first signaling messages remain unprotected irrespective of the deployed security mechanism. So, by splitting the registration procedure into two phases (pre-registration that includes the User Equipment's (UE) capabilities in cryptographic protocols and registration that contains the authenticated credentials and the responses to server's challenge), it becomes clear that the pre-registration phase remains always unprotected (Fig. 1.2). As a result, this phase remains a target for message manipulation, security degrading and other attacks initiated by insiders or even by external users (see Chapter 5).

On top of this, and considering that user equipment with limited processing capabilities may be present in such environments, a tradeoff between security and service quality is inevitable. Specifically, low resource devices cannot utilize state of the art security mechanisms. This list is extended with User Equipments (UE) that do not utilize a Universal Subscriber Identity Module/IP Multimedia Services Identity Module (USIM/ISIM) which can only use

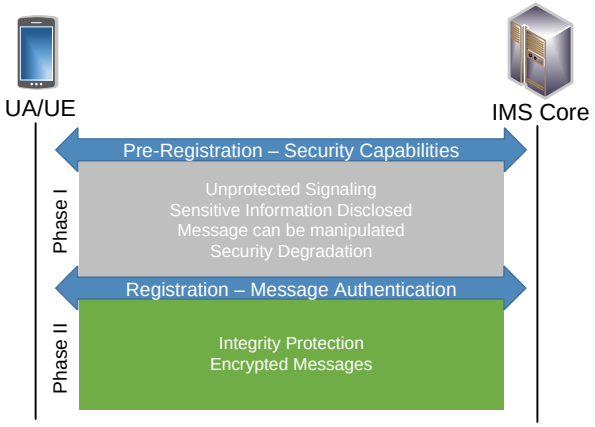


Figure 1.2: Registration phases in security tunneled modes.

the SIP Digest [20] protocol for authentication, as directed by the IMS specifications [18]. The utilization of such a protocol degrades the security level of the infrastructure since it does not provide confidentiality and integrity services, something that many researchers pinpoint [21,22]. Taking into account these facts, it can be derived that the lack of integrity protection in signaling messages or tunneling between users and servers, concerns a significant number of users who either have older equipments or originate requests with devices that do not employ a USIM/ISIM.

Many researchers have presented solutions for the detection of such attacks but most of them cover only a small subset of them [23–25], or they constrained in the detection without being able to prevent them [25–27], or even utilize [28] heavy weight protocols such as Public Key Infrastructures (PKIs) with a tradeoff between the security level and the introduced delay. Other solutions such as the Secure Multi-Purpose Internet Mail Extensions (S/MIME) cannot deter internal flooding and signaling attacks while they introduce significant overhead [29,30].

Similarly, the SIP specification [3] does not provide a mechanism that can adequately preserve users' privacy. An extension mechanism that protects users' privacy has been proposed in [31], but even in that mechanism the user identity remains unprotected during the first hop (between the user and the proxy server). In addition the proposed scheme is vulnerable to security degrading attacks. Coming to IMS, its security specifications [18] do not, also, propose any countermeasures for the protection of user identities. The employment of a mechanism such as S/MIME [32], could only protect small part of signaling's sensitive information but, on the other hand, the digital certificates involved may disclose user's identity. Alternatively, the AKA with IPSec and the SIP Digest with TLS [18] can protect users' privacy since they provide confidentiality and integrity services to the communication. However, during the registration procedure, the user must provide his identity in clear text.

The contribution of other researchers on privacy [32–35] is considered limited. The proposed schemes either require modification in the underlying infrastructure or the deployment of additional network entities [35]. The employment of a Public Key Infrastructure (PKI) is proposed in [32,34] which involves among others digital certificate management, signing,

revoking and updating procedures. On the other hand, symmetric encryption techniques introduce key related concerns (e.g. key exchange/agreement protocols, key storing).

Considering all these facts it is deduced that neither the SIP and IMS standards nor the researcher's proposals achieve a catholic protection framework for users and network operators from security and privacy breaches. Such incidents can inevitably degrade the QoS and consequently the reputation of the service provider while they will definitely have direct or indirect economical impact to the network operator.

1.4 Objectives

The primary objective of this thesis is the development of a security and privacy protection framework for SIP based environments (namely the VoIP/IMS infrastructures). This framework will be able to detect attacks, and prevent them before causing problems to the network and the users, and preserve user privacy.

An important parameter throughout this research was the support of low resource enabled devices and user devices that cannot utilize state of the art security protocols. Thus, heavy encryption protocols are not considered as a solution since such devices cannot support them and they also introduce significant overheads to an already burdened infrastructure.

Furthermore, crucial point of the proposed framework is the transparency to the network operations. Particulary, it must be easily employable, without the need of modifying the underlaying infrastructure, assuring therefore its inter-operability. Thus, the SIP transactions shall remain unchanged and mechanism's employment should not depend on a specific operator. Finally, the clients' operation are not affected.

Summarizing, the main objective of this research work is the development of a cross-layer architecture that:

- Ensures the availability of services
- Enhances the reliability of services
- Prevents unauthorized access and identity theft
- Protects users' privacy
- Supports low resource enabled user equipment
- Is lightweight, in regard with server's CPU and memory resources consumption
- Does not introduce significant overheads
- Is inter-operable
- Is transparent to network and users' operations

1.5 Methodology

In order to achieve the aforementioned objectives the work was divided into five phases:

1. Thorough study of the VoIP/IMS architecture and especially of the utilized communication protocols.
2. Study of existing vulnerabilities as described in the bibliography, and identification of new ones that may arise from the introduction of additional services in the system.
3. Study of the security mechanisms that can be utilized according to the SIP and IMS standards, as well as of those proposed by other researchers.
   (a) Comparative evaluation and implementation of the above security mechanisms when needed.
4. Setup of an experimental IMS architecture for simulating the existing and new attacks. New threats could also be detected during that stage.
5. Design and implement the proposed framework demonstrating that it achieves the desired objectives and that it satisfies the established requirements towards the detection and prevention of the threats identified in phases 2 and 4 above.
   (a) Experimental implementation of the proposed framework.
   (b) Comparative evaluation of the proposed framework and assessment of its contribution in that field of study.

The five main phase that followed during this Ph.D. research are depicted in Fig. 1.3.

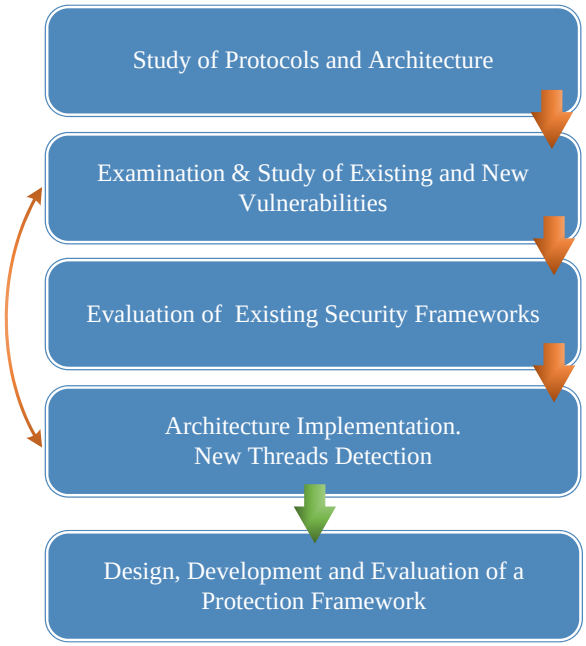


Figure 1.3: The main phases of research's methodology.

1.6 Contribution

The contribution of this Ph.D. research work is the development of a security and enhancing framework able to shield the IMS and VoIP infrastructures from malicious actions and attacks that originate from external or/and internal users. The contribution is depicted in Fig. 1.4.

Initially, a study of the IMS architecture and the utilized security protocols was performed [21]. The majority of the attacks that can be launched against VoIP and are also applicable in IMS environments, are thoroughly presented. Particularly, attack cases are studied against the security mechanisms that can be utilized in these environments, by considering two factors: the time frame of the attack and the users' access level. The study under this new context, actually evaluates the effectiveness of the security mechanisms and proves that they cannot always provide a satisfactory protection level against many security incidents. Further, a discussion of other researchers' proposals is included in order to assess their effectiveness and identify the specific set of threats and vulnerabilities that remain uncovered. Also, that study highlights the acts that may compromise users' privacy.

The development of an attacking tool for assessing the IMS's robustness against messages which not conform to SIP syntax (malformed) is presented in [36]. While these type of attacks are well-known from VoIP deployments, one might assume that they would have been eliminated in IMS due to its advanced and strict security policy. On the contrary, the experimental results prove the inability of the infrastructure to cope with slightly modified versions of SIP messages since the end-to-end delay revealed an increase of the precessing time of up to 4000%.

An attack where a malicious user can exploit a vulnerability of the SIP signaling in order to intercept a conference room is described in [37]. The development of a cross-layer mechanism that prevents these attacks is also presented. An advanced version of this mechanism [38] provides detection and prevention of attacks based on forging signaling messages.

IMS and generally VoIP availability, was the concern of [27]. Particularly, a bloom filter based framework was developed and implemented, towards the detection of attacks that threaten SIP-based systems' availability. As a development of [27] and [38], a detection and prevention framework of spoofed messages and endless sequences of concurrent messages (flooding attacks - see Section 5.4.2) that threaten these environment's availability, authenticity and accountability, is presented in [39].

Finally, the commutative functions are utilized for providing users with a meaningless one time identity without the need of any additional networking devices, schemes or any modification to the underlaying infrastructure. Every time a new transaction is originated from a user, a new identity is locally generated in order to preserve theirs anonymity.

It should be stressed that this Ph.D. is concerned with the signaling protocol SIP which is utilized by IMS cores and VoIP infrastructures. Summarizing, the contribution of this research is:

- Identification of threats inherited from VoIP to IMS infrastructures. Evaluation of the IMS

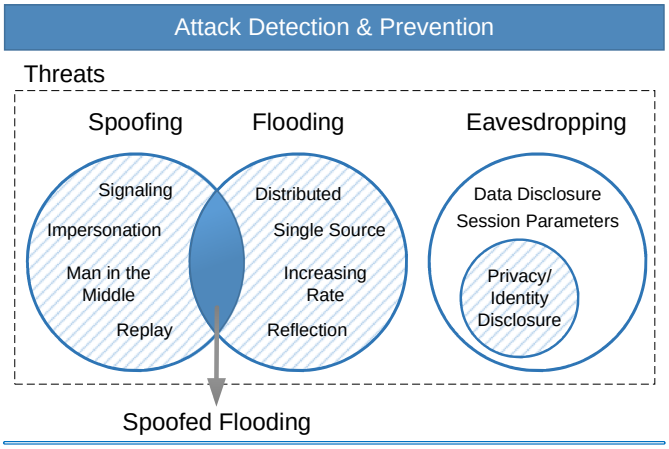


Figure 1.4: The contribution of the proposed security and privacy enhancing framework.

security protocols and other researchers' protection mechanisms' efficiency in preventing these attacks[1]

- Simulation of known attack scenarios and development of new ones[2][3]
- Experimental evaluation of the aforementioned attacks in an IMS test-bed environment[4][3]
- Development of a protection framework against a wide range of attacks:[5][6][7]
  - Impersonation of users and network entities.
  - Unauthorized call termination.
  - Denial of Service.
  - Billing fraud at the expense of other users.
- Privacy services to users[8]

[1] Vrakas N., Geneiatakis D., Lambrinoudakis C., "Evaluating the Security and Privacy Protection Level of IP Multimedia Subsystem Environments," *Communications Surveys & Tutorials*, IEEE, vol. PP, pp. 1-17, 2012.

[2] Vrakas N., Geneiatakis D., Lambrinoudakis C., "A Call Conference Room Interception Attack and its Detection," *in 7th International Conference on Trust, Privacy & Security in Digital Business*, Bilbao, Spain, 2010.

[3] Vrakas N., Geneiatakis D., Lambrinoudakis C., "Is IP Multimedia Subsystem Affected by "Malformed Message" Attacks? An Evaluation of OpenIMS", *in SECRYPT 2011, the International Conference on Security and Cryptography*, Seville, Spain, 2011.

[4] Geneiatakis D., Vrakas N., and Lambrinoudakis C., "Performance Evaluation of a Flooding Detection Mechanism for VoIP Networks," *in 16th International Conference onSystems, Signals and Image Processing*, Chalkida, Greece, 2009, pp. 1-5.

[5] Geneiatakis D., Vrakas N., and Lambrinoudakis C., "Utilizing Bloom Filters for Detecting Flooding Attacks against SIP Based Services," *Computers & Security*, vol. 28, pp. 578-591, 2009.

[6] Vrakas N., Geneiatakis D., Lambrinoudakis C., "A Cross Layer Spoofing Detection Mechanism for Multimedia Communication Services," *International Journal of Information Technologies and Systems Approach (IJITSA)*, vol. 4, pp. 32-47, 2011.

[7] Vrakas N. and Lambrinoudakis C., "An intrusion detection and prevention system for IMS and VoIP services," *International Journal of Information Security*, pp. 1-17, 2013.

[8] Vrakas N., Geneiatakis D., Lambrinoudakis C., "Obscuring Users' Identity in Next Generation Networks", *Computer Networks*, Elsevier (Under Review).

1.7 Structure

The second chapter of this thesis includes a description of the protocols utilized in the considered infrastructures. A more detailed description regards the application layer's protocols and specifically SIP. Its architecture, networking components, message's structure, headers and methods are also presented. Chapter 3 provides an overview of the IMS architecture and its components. User profile's structure and procedures such as media session and conference establishment are included in this chapter.

A detailed description of the authentication mechanisms is provided in Chapter 4. In the same chapter, the state of the art security mechanism are presented while their "compatibility" with the SIP authentication mechanisms is discussed. The security requirements and known vulnerabilities in IMS environments are discussed in the fifth chapter. The attacks are divided in different categories according to which security requirement they violate. At the same time, it is examined whether there are security mechanisms that can prevent them or not. Chapter 6 includes a thorough survey of other researchers related work while their advantages and drawbacks are also set forth and assessed.

Chapter 7 presents the protection frameworks proposed in this Ph.D. research. Specifically, a Bloom filter based mechanism is presented, which is able to detect attacks that threaten system's availability. Furthermore, mechanisms facilitating the detection and protection against a wide range of attacks that threaten system's accountability, authenticity and availability are presented. This chapter concludes with a privacy scheme for obscuring users' identities in order to provide anonymity services in SIP based environments. Chapter 8 concludes this thesis by summarizing the results of the research and giving directions for future work.

Chapter 2

Signaling and Media Protocols in VoIP/IMS Infrastructures

2.1 Introduction

Two or more entities willing to communicate must indicate their intention by utilizing a signaling protocol. Both members of the communication must inform each other when they want to start a session, when they are ready to accept or terminate it. There are protocols acting as media bearers or initiators for this session while others act as translators between addresses and domains. We can categorize the protocols involved in these infrastructures in three main categories:

- *Signaling protocols*: The session control functions handled by the singling protocols, are extremely significant in VoIP/IMS environments. A signaling protocol enables the communicating entities to establish a session, to update or modify it or even to terminate it and also facilitates the routing and accounting procedures. The SIP [3], the H.323 [40] and the Media Gateway Control Protocol (MGCP) [41] are the main protocols in this category. The SIP is the main signaling protocols for VoIP environments since the 3GPP has formally adopted it in the IMS infrastructures.
- *Media Control protocols*: These protocols act as media bearers and enable the network entities to indicate their capabilities in media codecs. The Real-time Transport Protocol (RTP) [42] is used for media streaming and the Session Description Protocol (SDP) [43] for negotiating the codecs for the upcoming session according to the capabilities of the communicating participants. Also, the Secure RTP (SRTP) [44] provides data integrity, confidentiality and origin authentication to the RTP.
- *Auxiliary protocols*: All other protocols utilized in VoIP/IMS environments are included in this category. These protocols provide address name resolution services and facilitate the proper configuration of the devices in the network. The Domain Name Service (DNS) [45] and the Dynamic Host Configuration Protocol (DHCP) [46] are examples of this

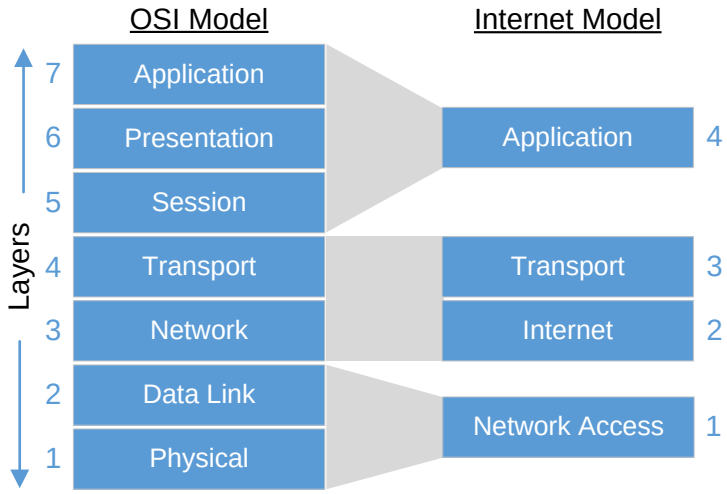


Figure 2.1: OSI to internet model association.

category. Also, the Diameter [47] protocol is widely utilized in IMS environments for providing Authentication Authorization and Accounting (AAA) services.

2.2 VoIP Protocols and the Internet Model

As the Open Systems Interconnection (OSI) model [48] defines, the protocols, that take part in a communicating session, encapsulate one another going from the lower layers to the higher ones and vice versa. Thus, the encapsulated protocols inherit the characteristics and also the flaws that a protocol possesses. The OSI is a model for categorizing the network protocols in layers and is divided in seven layers. For incorporating new technologies such as radio and satellite networks a new more flexible model was defined in 1974 [49], the TCP/IP model widely known as the internet protocol stack architecture [50]. The correlation between the TCP/IP and OSI models is illustrated in Fig. 2.1. According to TCP/IP reference model, there are four main layers: (i) the network access, (ii) the Internet, (iii) the transport and (iv) the application layer. The internet protocol stack and the auxiliary protocols are depicted in Fig. 2.2.

2.2.1 Physical Layer

The physical layer is the lower layer of the internet protocol model. The telephone trunk, the Digital Subscriber Line (DSL), the Ethernet (IEEE 802.3), the wireless networks (IEEE 802.11 and IEEE 802.16) are located in this layer. Moreover, it includes the V.90 modular protocol and Point-to-Point protocols such as the PPP and its predecessor the Serial Line Internet Protocol (SLIP).

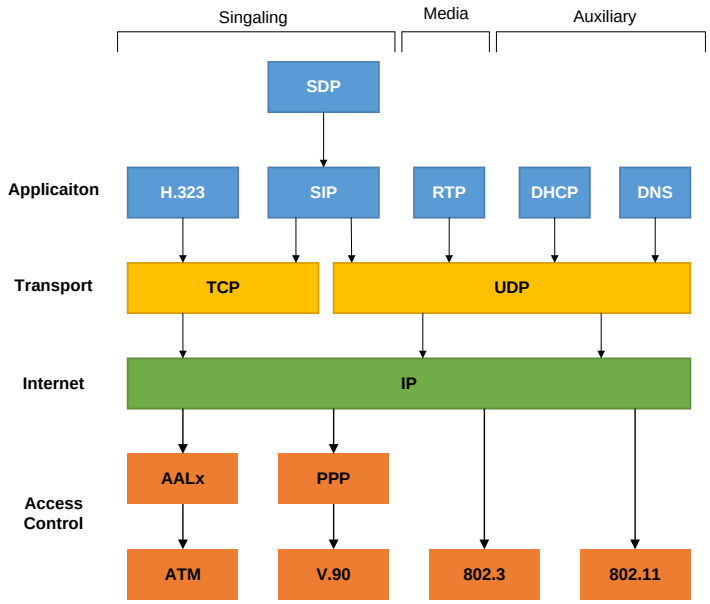


Figure 2.2: Internet protocol stack and auxiliary protocols.

2.2.2 Internet Layer

The internet layer is on top of the physical layer and hosts the Internet Protocol (IP). Every interface is associated with an IP address. The IP address can take the fourth's (IPv4) or sixth's version (IPv6) [51] format. The IPv4 is the commonly used protocol which supports 32 bit of addresses. The IP packet includes two different addresses in its header: the source's and the destination's address. Its successor, the IPv6 provides a much larger address space while it supports 128 bit addressing. Every incoming packet is routed according to the IP header and it is encapsulated in the physical layer frames. The same technique of encapsulation is applied when the data come from the higher layers towards the lower ones. Thus, the TCP segments of the transport layer are encapsulated into the IP packets. The IP is a connectionless protocol which means that the packet can be exchanged without any prior session establishment procedure.

2.2.3 Transport layer

In this layer, the port number is utilized for the proper routing of message fragments from the other layers to the corresponding IP addresses. The SIP signaling protocol usually listens to port 5060. For instance, the core signaling entities of IMS usually listens to ports 4060, 5060 and 6060. However this may vary since only the first 1024 are dedicated to specific services. The protocols belonging to this layer are the TCP, UDP, RTP, SCTP, SRTP and others. The TCP and UDP protocols are the main transport protocols for bearing the SIP signaling.

2.2.3.1 Transport Control Protocol

The TCP protocol [52] is used for session establishment and data exchange between the communicating entities over the IP. It is a connection oriented protocol and uses acknowledgement

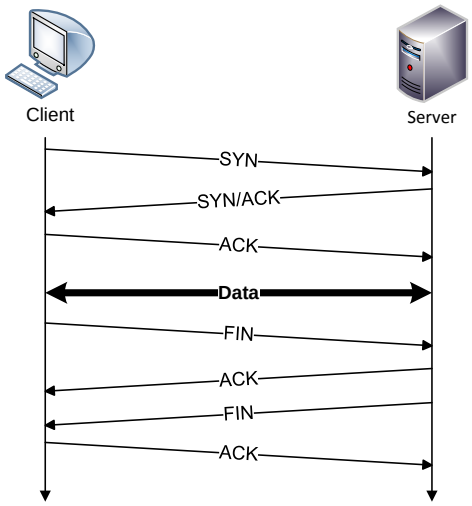


Figure 2.3: Session establishment and termination handshake in TCP.

requests and sequence numbers in order to detect lost messages. The missing fragments are retransmitted until they reach their destination. The damaged segments are detected through checksum functions included in their headers. Network entities can establish a TCP session through a 3-way handshake. This 3-way handshake has a major vulnerability that enables the attacker to launch flooding attacks in many different environments. This vulnerability is described in detail in Section 5.4.2. The TCP's session establishment procedure is depicted in Fig. 2.3.

2.2.3.2 User Datagram Protocol

The most significant advantage of UDP [53] is its capability for greater transmission rates in sending and receiving data. It is considered an unreliable protocol since it does not support mechanisms to ensure the reception of messages. It is a connectionless protocol so it does not require a handshake for session establishment similar to the TCP's.

2.2.3.3 Stream Control Transmission Protocol

The SCTP protocol [54] provides more reliable data transport than TCP. It supports message fragmentation and selective acknowledgements. Also, it provides sequenced delivery of messages and supports multi-homing at both source and destination. Finally, it is not vulnerable to the TCP-SYN type of flooding attacks since it requires an improved 4-way handshake.

2.2.4 Application Layer

At the top of the stack is located the application layer. The signaling protocols such as SIP and H.323, media streaming such as the RTP and media session initiation and codec negotiation such as SDP are included in this layer. Further, the DHCP protocol which is responsible for facilitating the configuration of network devices, the Simple Mail Transfer Protocol (SMTP)

[55] for handling e-mail transmissions and the DNS [45] for resolving domains to IP and vice versa are examples of protocols of this layer.

2.3 Session Initiation Protocol Architecture

The model of IP telephony requires a variety of different protocols in order to provide to the users an acceptable quality of service. Concerning session establishment, manipulation and management issues, three are the main protocols that can support them: the SIP, the H.323 and the MGCP [41]. While the latter is dedicated to transmitting media data among gateways of different networks, the H.323 and SIP have been the two candidates for session control and management in VoIP. Later, the SIP protocol has been established as the main signaling protocol for VoIP/IMS and generally Next Generation Networks (NGNs) after the 3GPP's decisions to include it in its standards [2]. The main advantages of SIP include its flexibility to incorporate new services and the text based form of its messages which comprises it a light weight protocol.

The SIP as a signaling protocol can provide the following basic services:

- Location information of a member
- Readiness of a member to establish a session
- The exchange of information required for establishing a session
- Informing about the capabilities of a specific device
- Session modification and update
- Session termination

The SIP protocol, has been extended to support additional requirements. This extension has given the opportunity to introduce new services such as instant messaging, conferencing and event and presence notification.

2.3.1 SIP Message Structure

The SIP is an application-layer signaling protocol for initiating, modifying, and terminating multimedia sessions among one or more participants. The general structure of the SIP protocol is inherited by Hyper Text Transfer Protocol (HTTP) [56] and thus SIP messages are text based similar to the HTTP ones. A typical SIP message consists of three distinct elements: the first line, the headers and the message body. A basic SIP message is illustrated in Fig. 2.4.

There are two different types of SIP messages: a) the requests and b) the responses. The type of the message is indicated in the first line and comes along with the receiver's address. The headers include information such as the message originator ("From" header), the recipient ("To" header), the sequence number of the message ("CSeq" header) and many more which are described in the Section 2.3.1.3). The "Call-ID" and "tag" are unique per session numbers which

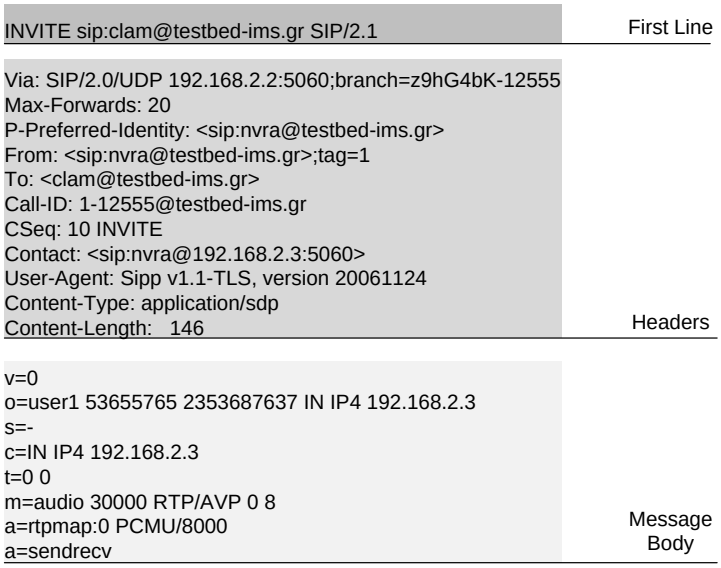


Figure 2.4: SIP message structure.

Table 2.1: Main SIP Request Methods

Request	Description
REGISTER	Registration or de-registration of a specific subscriber to the network for a specified period of time.
INVITE	Session establishment. This method includes message body with the device's capabilities in media codecs.
BYE	Terminates the current session. It does not include message boy.
ACK	Acknowledges the reception of final responses. It may include message body when the INVITE does not.
CANCEL	Cancels a pending request. It is used during a handshake in order to cancel it.
OPTIONS	Requests information about the capabilities of a device or server.

are generated by random number generators in conjunction with the device's host name or IP address. The addresses in SIP messages are not only in IP format but may take the form of the Uniform Resource Identifier (URI). The message body contains the SDP protocol that provides the media streaming information which is required for determining the appropriate/available audio and video codecs for the upcoming media session. Considering the example illustrated in Fig. 2.4, we can derive information about the IP addresses, the type of the session (audio), the utilized protocol (RTP) and the media codec (PCM). The message body is an optional element and is not included in all SIP messages.

2.3.1.1 SIP Requests

The main methods of request messages are the REGISTER, INVITE, BYE, ACK, CANCEL and OPTIONS. The methods define the purpose of a message and comprise the very first string

Table 2.2: Request Methods as Extension to SIP for Supporting New Services

Request	Description
SUBSCRIBE	Initiates the subscription to service for receiving notifications about specific events.
NOTIFY	Sends notifications to a subscribed device for a specific event as long as the duration of the subscription is valid.
REFER	Requests for a device to access a specific source. This can be used for call transfers or to invite a subscriber to access a conference URI (invite a user to join a conference).
UPDATE	It is similar to re-INVITE but modifies the parameters of a pending session not of an already established one.
MESSAGE	It is responsible for sending instant messages among the subscribers through a SIP request
PRACK	It acknowledges the reception of a provisional response (with code 1xx) excepting the "180 Ringing". The rest of the responses are acknowledged with ACK requests.
INFO	It sends information to a member of an already established session. In contrast with a re-INVITE, INFO does not modify the session's media characteristics. It may contain message body.

of the SIP message. The first line of the request messages also called *request line*. As already stated, the SIP method follows the URI of the requested recourse. The first line of the request ends with the SIP version number. The functionality of the main SIP methods that are included in the RFC 3261 [3] are defined in Table 2.1. Other request methods have been added later as extensions to SIP functionality such as the SUBSCRIBE, NOTIFY, REFER, UPDATE, MESSAGE, PRACK and INFO (see Table 2.2) and are defined in other RFCs and briefly described in Table 2.2. A more detailed description is included in the next section for the INVITE, REGISTER and REFER requests since they usually comprise a target for the attackers.

REGISTER Method

The REGISTER is utilized by users in order to inform the network about their current contacting address ("Contact" header). This type of request does not include message body. The registration messages include the header field "Expires" that can be used only in REGISTER methods. This header indicates the time period that a registration is valid. The numerical value denotes seconds and if this field is empty, a predefined value is assigned (usually 3600 seconds) by the server. When the user wants to turn off his device or to sign out of a service (de-registration), he sends a REGISTER request with the "expires" and "Contact" header set to "0" and "*" correspondingly. Therefore, there is not a dedicated request method for de-registration.

In the given registration example of Table 2.3, the user "nvra" wishes to register into the testbed-ims.gr service for 3600 seconds, his contacting address is the IP 192.168.2.3 and the port that his client listens is the 5061. Note that, the initiator and the receiver, "From" and "To" headers, include the same user (nvra) in registration requests.

Table 2.3: Example of a SIP Request - REGISTER Method

Registration Request
REGISTER sip:testbed-ims.gr SIP/2.0 Via: SIP/2.0/UDP example.com:5061;branch=z9hG4bKgqcqpfkh Max-Forwards: 70 To: "Nikos" <sip:nvra@testbed-ims.gr > From: "Nikos" <sip:nvra@testbed-ims.gr >;tag=zlpow Call-ID: pzhrakeslbvziwz@testbed-ims.gr CSeq: 1 REGISTER Contact: <sip:nvra@192.168.2.3:5061>;expires=3600 Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE User-Agent: Twinkle/1.2 Content-Length: 0

INVITE Method

A session can be established by utilizing the INVITE SIP request. Every response to this request is acknowledged with an ACK request. This method includes in message's body, the media capabilities that will be used during media codecs negotiation with the server for the upcoming media session. If the media parameters are not defined in the INVITE message then the ACK request must include them. In case where these parameters are not acceptable by the requested resource then the latter sends a BYE request in order to terminate the session. A re-INVITE may also be used in order to update the session's parameters. The re-INVITE does not comprise a new method so a normal INVITE is utilized in such a case. A re-INVITE request can be distinguished from a retransmission of a normal INVITE, by the headers "To", "From" and "Call-ID" which are still the same while the sequence number included in the "CSeq" header has been increased. An example INVITE request is depicted in Table 2.4. That request is originated by the user nvra who intends to call user clam. The message body is located right after the "Content-Length" header and includes the SDP parameters.

REFER Method

The REFER method is not included in the SIP specifications but it has been added latter as an extension to support new services. Its functionality is defined in RFC 3515 [57]. It is mainly used to implement call transfer services or to invite other users to join a media conference (see Section 3.5.2). The REFER method introduces new headers in the messages and requests from the indicated device to access a URI. This URI is included in the "Refer-To" header while the request initiator is included in the "Referred-By" header. The invited member is indicated in the request line of the SIP message.

2.3.1.2 SIP Responses

SIP responses are sent as a response to SIP request messages. They consist of a code and a string which indicate the exact type of the message. Unlike to the requests, the responses start

Table 2.4: Example of a SIP Request - INVITE Method

Session Establishment Request
INVITE sip:clam@testberd-ims.gr SIP/2.0 Via: SIP/2.0/UDP testbed-ims.gr:5061;branch=z9hG4bKeinfmycg Max-Forwards: 70 To: "Clam" <sip:clam@testbed-ims.gr > From: "Nvra" <sip:nvra@testbed-ims.gr>;tag=lqwmz Call-ID: cruvcrfknakzrre@testbed-ims.gr CSeq: 28 INVITE Contact: <sip:nvra@192.168.2.3:5061> Content-Type: application/sdp Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE User-Agent: Twinkle/1.2 Content-Length: 168 v=0 o=twinkle 645497100 172480165 IN IP4 testbed-ims.gr c=IN IP4 testberd-ims.gr m=audio 8000 RTP/AVP 98 97 8 0 3 101 a=rtpmap:8 PCMA/8000

with SIP's version and follows the response's status code. The first line of the response is called status line. The response codes can be divided in six classes (see Table 2.5). The first five are adopted from the HTTP protocol while the last ones are introduced especially for SIP.

If a device is not capable of identifying a response code, it must handle it according to the class in which the response code belongs to. For instance, if an error code "421 Extension Required" is not identified by a device, the latter translates it as a generic error of the 4xx class which the 421 belongs. Thus, the 421 corresponds to a "Client Error". The 100, 180, 183, 200, 202, 401/407, 423 and 486 response codes that they will be used in this thesis are described below.

100 Trying

This response code indicates that a request has been received by the corresponding entity and is currently being parsed. For instance, the server may execute queries to the databases or other time demanding process. It does not contain neither a "tag" field after the "To" header, nor a message body.

180 Ringing

This response informs the caller that the callee has received the INVITE request and the User Agent (UA) has all the resources available in order to establish a session. Message body may carry ringtones or security related information.

Table 2.5: Response Classes

Class	Type	Description
1xx	Provisional	Informs about the progress of a request.
2xx	Success	The request has been successfully processed.
3xx	Redirection	Provides other location where the request can be processed.
4xx	Client Error	Indicates that the processing of a request has been failed by the client. The request has to be resent according the response's directions.
5xx	Server Error	Indicates that the processing of a request cannot be fulfilled by the specific server.
6xx	Global Failure	Request is declined. Indicates that the processing of a request has been failed and the network cannot handle it.

183 Session Progress

It carries information about the progress of a call. However, it does not include information about the status of an INVITE request. It contains in its message body the media capabilities of a device. This information is used during media negotiation procedures in order, the communicating entities to decide about the type and the codecs of the upcoming media stream.

200 OK

It is an indication that a network entity has received and accepts a request while the retransmissions of the corresponding request are stopped. The 200 OK contains message body with device's media capabilities (when responds to an OPTIONS request). It may also include authentication information in "rspauth" header in order to authenticate the server to the UE when SIP Digest [20] is used. Message body is not allowed for the rest of request methods. Table 2.6 depicts a "200 OK" which includes a message body.

202 Accepted

It is a response to REFER and MESSAGE methods. According to RFC 3265 [58], the difference with a 200 OK response is that the 202 indicates that the request has been understood but the authorization is pending or it has been granted. Later, the RFC 6665 [59] has revoked the special meaning of "202 Accepted" and proposes that it shall be handled as a "200 OK". Moreover, it explicitly states that NOTIFY and SUBSCRIBE methods must not be responded with a "202 Accepted".

401 Unauthorized

It indicates that the corresponding request requires authentication in order to be processed. The 401 status code is similar to the "407 Proxy Authorization Required" with the slight difference

Table 2.6: SIP Response Example - 200 OK with Message Body

200 OK Response
SIP/2.0 200 OK Via: SIP/2.0/UDP testbed-ims.gr;branch=z9hG4bKec6e.6d72b475.0 To: <sip:nvra@testbed-ims.gr>;tag=phwza From: "clam" <sip:clam@testbed-ims.gr>;tag=252C2E2F Call-ID: 189243426524@testbed-ims.gr CSeq: 111 INVITE Contact: <sip:nvra@testbed-ims.gr:5070> Content-Type: application/sdp Allow: INVITE,ACK,BYE,CANCEL,OPTIONS, NOTIFY,SUBSCRIBE,INFO,MESSAGE Server: Twinkle/1.2 Content-Length: 199 v=0 o=twinkle 503369823 248224871 IN IP4 192.168.1.56 c=IN IP4 testbed-ims.gr m=audio 8000 RTP/AVP 97 101 a=rtpmap:0 PCMU/8000 a=rtpmap:101 telephone-event/8000 a=fmtp:101 0-15 a=ptime:20

that it is used by different network entities. Such a response contains information required for the initiation of the cryptographic algorithms ("nonce" header field), which algorithms will be used ("algorithm" header field) etc. (see Table 2.7). The qop (quality of protection) is optional and indicates the level of the protection, while the "stale" denoted whether the nonce value is valid or it has expired. The 401/407 responses do not contain message body.

423 Too Brief

When the server receives a registration request that contains a short expiration time, the server rejects the request with a "423 Interval Too Brief" which includes the minimum value allowed for expiration, in a registration request.

486 Busy Here

The session establishment request has been successfully received by the callee but the latter does not intend or its UA cannot process additional calls at this time. It is similar to the busy tone in PSTN networks.

2.3.1.3 SIP Headers

The SIP headers comprise the most important part of the message since they exist in every SIP request or response. There are some headers that are optional but a SIP message without

Table 2.7: SIP Response Example - 401 Unauthorized

401 Unauthorized Response
SIP/2.0 401 Unauthorized Via: SIP/2.0/TLS proxy.testbed-ims.gr:5060;branch=z9hG4bK-3254 From: "Bob" <sips:bob@ testbed-ims.gr >;tag=fjjhgwg21 To: "Bob" <sips:bob@ testbed-ims.gr >;tag=1410948204 Call-ID: 32144342@ testbed-ims.gr CSeq: 967 REGISTER WWW-Authenticate: Digest realm="testbed-ims.gr", qop="auth", nonce="494ee5e31ca55a3e2b88ca73f94da576bb", opaque="", stale=FALSE, algorithm=MD5 Content-Length: 0

headers at all is invalid, while in contrast a message body may not present, as described above. The most common headers are briefly described in this section.

We can categorize the SIP headers into four categories: (i) The generic headers which are important for protocol's basic functionality, (ii) the routing headers which are used for indicating and determining the routing path, (iii) the authentication headers used for exchanging authentication vectors and parameters and (iv) the headers introduced as extension for handling the signaling for new services provisioning.

Generic Headers

- *Allow*: It indicates the request methods supported by the device. The methods are separated with the comma "," symbol.
- *Call-ID*: This header is unique identifier of a dialog. Thus, until the end of the dialog, the exchanged messages include the same Call-ID. Also, it remains the same in the registration requests of the same UA. This value is generated by random number generators or encryption functions with a random key and a seed [58]. The produced number is concatenated with the domain of the request originator and takes the "random-number@host" form.
- *Content-Length*: It provides the size of the message body attached to a SIP message in bytes. When a message body is not present, this header takes the zero "0" value.
- *Content-Type*: It defines the type of data included in the message body. For instance, it may have the values "application/sdp", "text/plain", "text/html" or "application/xml+conf" when the message body includes SDP, plain text, HTML or XML code correspondingly.
- *Contact*: This header carries the address of a device where the latter can receive requests/responses. It takes the URI form and may contain a display name. When display name is present in the "Contact" header, the URI must be included in "< >". The de-registration requests contain the "*" symbol denoting that the UA is not available for receiving request.
- *CSeq*: It indicates the sequence number of a message and it is a mandatory header for

every message. The value is a decimal number and is increased by one for every new request. After that number, follows the method of the message and it is mainly used for distinguishing a new request from a re-transmission.

- *Expires and Min-Expires*: It denotes the time, in seconds, that a URI will be valid and available for receiving request. When a user initiates a de-registration the "expires" header field takes the zero value. This header is utilized by the attackers in order to launch de-registration attacks (see Section 5.2.1.4). The Min-Expires has the same use but it is utilized by the server when the user has included a very short amount of time for URI expiration. In such a case, the corresponding UA is informed about the minimum value allowed, with a "423 Interval Too Brief" response that includes that value in "Min-Expires" header.
- *From*: It includes the identity (SIP URI) of the message originator. When it contains a display name, the SIP URI must be enclosed into "< >". It may also contain the tel URI of the message initiator.
- *Max-Forwards*: It is a mandatory header in requests and denotes the maximum number of hops for a specific message. A proxy is able to edit this value by decreasing it by one after each hop.
- *Priority*: As its name implies, it indicates the priority of the message. It may take the values "emergency", "urgent", "non-urgent" and "normal". When this header is missing, the message is handled with normal priority.
- *Require*: It defines the extensions that an entity must support in order to be able to understand and process a request.
- *To*: It contains the destination of the signaling message. That is, it indicates the callee. It may also contain a display name.
- *User-Agent*: It contains information about the device utilized for accessing a resource. This functionality may disclose critical information about the utilized UA's vulnerabilities and thus an attacker could exploit them in order to launch attacks.

Routing Headers

- *Record-route*: It forces the signaling traffic through a server. It contains the addresses of the specific servers that must be traversed. Every server inserts its SIP URI in this header. In registration request this header is ignored.
- *Route*: It forces the signaling traffic through specific proxies listed in that header. It can be added to a request according to "Record-route" header.
- *Via*: This header contains the path that the request has traversed. The order of the URIs is important since the responses to the specific request will take that route. Via header also includes the utilized transport protocol and the port.

Authentication Headers

- *Authorization*: It is included in the request when a UA must be authenticated to the network after receiving the "401 unauthorized". It contains the field "username", the "response", the "digest-uri" taken form the request line, the optional "cnonce" which is used as a pre-defined input to a hash function, the "nonce-count" which indicates the number of requests for enhanced security level and the "qop".
- *Authentication-Info*: This header provides mutual authentication to HTTP digest. The server includes the "rspauth" which contains the proof that it knows the password of the user's UA. This header is included in the 200 OK server's response after a successful authentication of a user.
- *Proxy-Authenticate*: It is a header field used when authentication is required. Specifically, when a server requires the UA to authenticate a request, challenges the latter with the data included in the "Proxy-Authenticate" header. The UA responds to that challenge based on this received information. This header includes the fields:
  1. *Digest realm*: Provides the user with the origin of the server in order to know which username to use.
  2. *qop*: Takes two values, the "auth" or "auth-int" which define how the digest is generated [20].
  3. *nonce*: A unique hexadecimal string based on a time stamp and a private key used for calculating the response digest from a UA.
  4. *opaque*: It is a value that must be returned unchanged to the server.
  5. *stale*: It can take the values TRUE or FALSE denoting if the nonce value is new or an old one and thus the UA must resend the request with a new one.
  6. *algorithm*: Provides the security algorithm that is used for generating the authentication response.
- *Proxy-Authorization*: It is used in requests initiated by the UA, as a response to the challenge for authentication (in response to "407 Proxy Authorization Required"). This field includes the user's name in "Digest username" field, the response to the challenge in "response" field (indicates that the user knows the password) and the used nonce, opaque and the realm.
- *Security-Client*: It contains the client's supported security suites. It may contain values such as "tls", "ipsec-3gpp" and "digest" denoting the SIP Digest with TLS, IMS AKA with IPSec and SIP Digest correspondingly (see Chapter 4).
- *Security-Server*: It contains the server's supported encryption suites.
- *WWW-Authenticate*: When the server receives an unauthorized request or the authentication data is unacceptable, it responds with "401 Unauthorized" including an authentica-

tion challenge in the "WWW-Authenticate" header. It contains the same header values as the "Proxy-Authenticate".

SIP Extension Headers

- *Privacy*: Indicates the level of privacy in SIP messages. It accepts the values "header", "session", "user", "critical" and "none" [31]. A detailed description is included in Section 6.4.
- *P-Asserted-Identity*: It contains the public identity of a user that the server has asserted. This happens when the user has more than one registered identities and he has not provided ("P-Preferred-Identity" header missing) which one intends to use for this session. Another case where the server asserts an identity is when the user has chosen an invalid one for the corresponding service.
- *P-Preferred-Identity*: It specifies the public user identity in cases where a user owns more than one. Since, different identities are associated with different services, the user must define which identity must be used.
- *P-Network-Access-Info*: This header provides information concerning the type of access. This information is important for the application servers in order to customize the service parameters for providing better quality of service. It is an extension to SIP for IMS [60] and contains two fields separated by a semicolon: the type of the utilized technology (UTRAN, GERAN, etc.) and optionally the identity of the radio cell.
- *Referred-by*: It contains the URI of the REFER request originator and it is optional.
- *Refer-To*: It contains the SIP URI, tel URI, mailto URI or a URL of the resource that the receiver of the REFER request has to access. It is only contained in REFER requests and its presence is mandatory.
- *P-Media-Authorization*: It is used for media authorization. It includes a number of media authorization tokens which are responsible for QoS and media stream authorization, according to the type of the message body.

2.3.2 Network Entities in SIP

There are two main network entities in SIP: (i) the UAs which are the devices that support SIP message exchange and media streaming and (ii) the servers for receiving, routing, parsing, managing SIP requests and handling the subscribers' records.

2.3.2.1 User Agents

The UA is a device that supports the SIP protocol. A fundamental requirement for a UA is to be able to establish a media session between two or more UAs. It shall also support SDP and TCP messages transmission. A UA provides its capabilities, in SIP requests for facilitating the

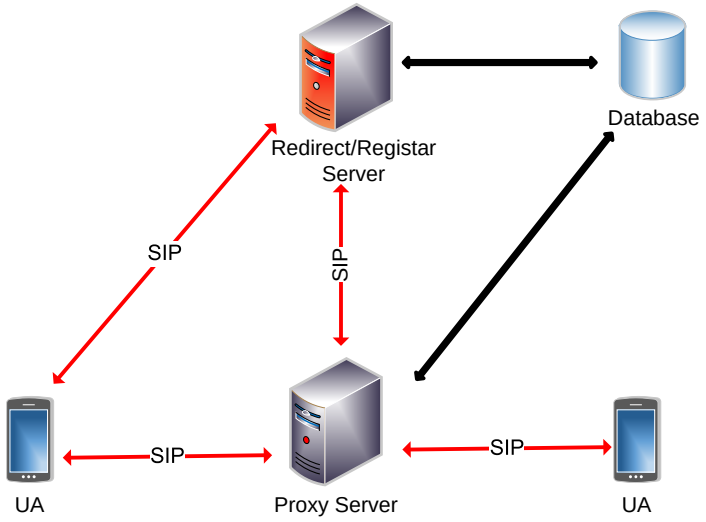


Figure 2.5: SIP servers and UAs interaction.

maximum utilization of the resources and achieving a better level of communication quality with other UAs.

The UA consists of two logical entities at the same time: the UA Server (UAS) and the UA Client (UAC) which are used for originating requests and responses correspondingly. A UAC can process response messages and initiate a request by taking as input an external command (such as a user's intention to place a call). A UAS can process requests and initiate the corresponding responses. A UA act as UAS or UAC by generating/processing responses and requests during a session.

2.3.2.2 SIP Servers

The SIP servers are the network elements that are dedicated to receive and process the SIP requests. There are three different types of SIP servers:

1. The SIP proxy server
2. The SIP registrar server
3. The SIP redirect server

These servers comprise a logical entity and depending on the implementation they can coexist. The server-database communication is not achieved through the SIP protocol. The interaction between the UA and the server is illustrated in Fig. 2.5.

SIP Proxy Server

The proxy servers receive the SIP signaling from the UAs and they forward it to other proxies or UAs. They also respond according to the type of request. A proxy is not allowed to modify the SIP headers in the majority of the messages in contrast with the UA which has this ability.

They may contact a location service or a database in order to determine the next hop or a UA. They are not capable of parsing or modifying the message body and they cannot remove it from a request or response. However, they can modify the headers which related to the routing of the messages. The main headers that bear such an information are the "Via" and "Record-Route". It is also allowed to modify the Request-URI included in the first line of a request message.

Two modes of operation are supported by the proxies:

- *Stateful mode*: In this mode of operation, the proxy stores the data from requests and responses in order to use them for handling future transactions. It can re-transmit requests for a specified period of time until the reception of a response. Finally, it may require authentication of a UA.
- *Stateless mode*: In this mode of operation, the proxy does not store the signaling data. Therefore, it discards all transaction's data when they are parsed and forwarded to the next hop. Also, it cannot re-transmit requests and does not employ timers. Responses are not generated at this mode of operation.

SIP Registration Server

Registration servers are responsible for receiving, parsing and processing SIP REGISTER requests (see Section 2.3.1.1). Moreover, they can store user's contact address in a database or a location service (see Fig. 2.5). They accept only REGISTER requests, in all other cases they respond with a server failure and specifically with a "501 Not Implemented".

SIP Redirection server

Redirection servers do not forward requests, like the proxy servers, but they are only limited to send responses. They execute queries to the database or to the location service in order to determine the location of a network entity. They respond with responses of the class 3xx (except only to CANCEL requests where they respond with responses of the class 2xx) informing the UA with a list of alternative servers that they are capable of processing UA's request. The employment of a redirect server is required in cases where a proxy or a UA has no connectivity due to a technical problem or when load balancing techniques are applied for avoiding network congestion.

2.3.3 Session Control

For controlling a session, the SIP standard [3] defines a specific exchange of requests and responses between the UA and the responsible server. Such a transaction (handshake) between the UA and the servers, provides the user with the capability of establishing and terminating sessions and other procedures that the SIP request methods allow (see Section 2.3.1.1). The three deferent servers described above are in charge of handling different events. The most significant

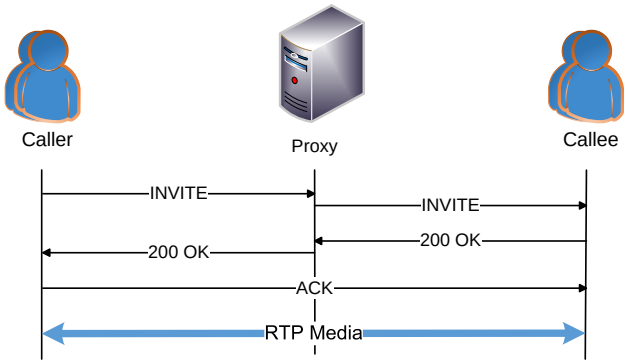


Figure 2.6: Session establishment in SIP.

procedures are the user registration, handled by the SIP registrar, the session establishment, handled by the SIP proxy, and the redirection, handled by the SIP redirection server.

2.3.3.1 Registration

An exchange of two messages between the UA and the registration server is required, in the most basic version of user's registration transaction. The UA sends a SIP REGISTER request to the registration server. The request line contains the server's address (SIP URI) while both the headers "From" and "To" contain the initiating users identity. The registration server parses the request and if it is valid it responds with a "200 OK" message back to the UA, denoting a successful registration. The registration server also stores the UA's address for routing purposes. In case where the registration server requires authentication, it responds with a "401 Unauthorized" that contains the challenge data. A new REGISTER message shall be generated, by the UA, which shall contain the response to that challenge, proving that it knows the password. The authentication procedures are described in detail in Chapter 4.

2.3.3.2 Session establishment

Three messages are required for a successful session establishment. The proxy server acts as the intermediate between the caller and the callee and forwards the messages from the first to the other and backwards. The example in Fig. 2.6 illustrates the role of the proxy server during a session establishment procedure. The caller generates an INVITE message and sends it to the proxy. The latter forwards the message to the callee which in turn responds with a "200 OK" accepting the invitation. The response reaches the caller through the proxy server. The caller sends an acknowledgement (ACK request) directly to the callee according to the RFC's 3261 [3] directions for end-to-end routing of "ACK" requests that follow a "200 OK" response. After the reception of the ACK request by the callee, the session has been successfully established and the UAs may start exchange media data. In case the proxy requires authentication, the latter responds to the caller's INVITE with a challenge in a "407 Proxy Authentication Request". Therefore three more messages are required for the session establishment (a detailed description of authentication procedures can be found in Chapter 4):

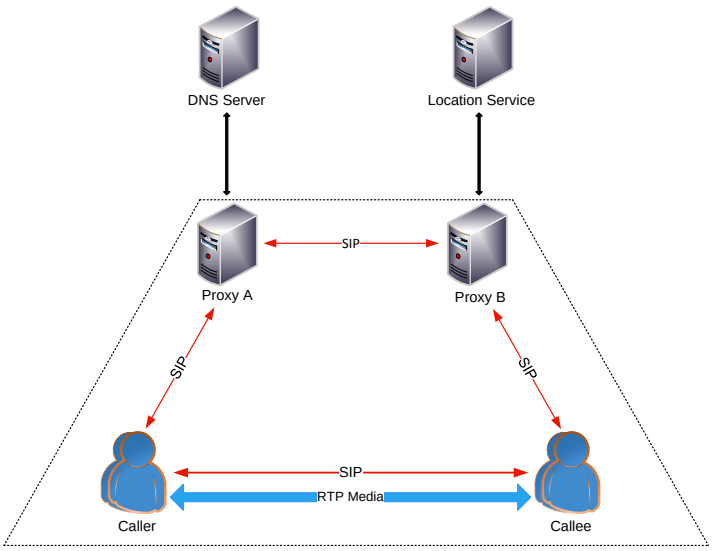


Figure 2.7: SIP trapezoid.

1. The server's response (407 Proxy Authentication Request) that contains the challenge data to the caller.
2. The reception acknowledgement by the caller.
3. The generation of a new INVITE request that contains a valid response to the server's challenge from the caller.

The most common scenario of VoIP topology is the SIP trapezoid [3]. This term has been assigned to VoIP topology due to the trapezoid shape formed during the signaling among UAs and servers (see Fig. 2.7). The SIP trapezoid derives from the following scenario:

- The caller sends an INVITE request to Proxy A in order to establish a session with the caller which has the "clam@testbed-ims.gr" URI address.
- The Proxy A receives the request but it is not responsible for the testbed-ims.gr domain and therefore queries the DNS server.
- The DNS server responds with the address of proxy B.
- The proxy A forwards the INVITE request to proxy B.
- The proxy B (which is responsible for the testbed-ims.gr domain) receives the INVITE request and executes a query to the location service in order to locate the callee.
- The proxy B forwards the INVITE to the callee while the latter responds with a "200 OK".
- The reverse procedure is executed until the reception of "200 OK" from the caller. The intermediary queries to the location services and the DNS servers are not required since the route has been stored in the "Record-Route" header of the SIP message.
- Finally, the caller receives the "200 OK", and sends an ACK request directly (end-to-end)

to the caller, finalizing in that way the trapezoid shape.

2.3.3.3 Redirection

When load balancing scenarios are employed or a server is not functioning, a redirection is executed by the redirection server. Such a scenario is illustrated in Fig. 2.8. The caller sends an INVITE request to the redirection server which responds with a single "302 Moved temporally". That response SIP message includes the address of a fully functional server. That response is acknowledged with an ACK request from the caller while the latter has to resend the INVITE to the address of the indicated server.

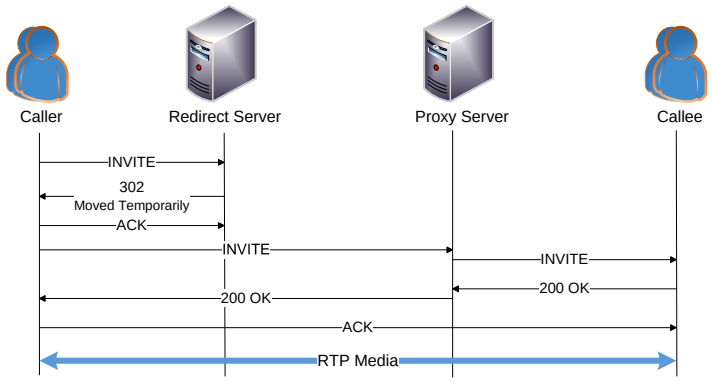


Figure 2.8: SIP redirection example.

2.4 Media Description and Transfer protocols

2.4.1 Session Description Protocol

The request methods that are utilized for establishing, updating or modifying a session, include the SDP's options in the SIP's message body. The SDP [43] is a text-based protocol and it is used for negotiating the properties of a media session. UAs provide their capabilities in codecs (mpeg, H.261, etc.) and the type (audio/video etc.) of the session they intend to establish, utilizing the SDP protocol. Other information such as the duration of the specific session, the address and ports that the initiator is supposed to use, are also described in the SDP.

In contrast with SIP, the SDP lines have fixed order to provide easier parsing and error detection. The lines start with the type (a single character) before the equal sign "=" and follows its value, without any whitespaces among them. An example of an SDP description is depicted in Table 2.8. The version of the SDP is included in the "v" field. Then, follows the "o" field which defines the session originator. This field is comprised of six values: the <username>, a <session-id> (503322823) and a <session-version> (248221171) which are actually timestamps, the <network type > which has the value "IN" denoting the internet, the <address-type> (IPv4, IPv6 etc.) and finally the originator's address. The "s" field includes the name of the session in text format (IMS Call) while its duration is included in "t" (start and stop times separated by a

Table 2.8: Example of SDP in SIP Message Body

Session Description Protocol	
SIP	Content-Type: application/sdp Content-Length: 174
Message Body	v=0 o=twinkle 503322823 248221171 IN IP4 192.168.2.2 s= IMS Call t= 0 0 m=audio 18533 RTP/AVP 3 0 101 a=rtpmap:0 PCMU/8000 a=rtpmap:101 telephone-event/8000 a=fmtp:101 0-11

white space). The media description is defined in the "m" field: Its accepted values start with the media type (take values such as application, audio, video, message and text) and the port (18533). The utilized media protocol (RTP) with its format ends that line. More than one media attributes can be defined in the "a" field. It is an optional field and provides information about the payload format and the clock rates ("rtpmap" value) and codec specific parameters ("fmtp" value). For more information refer to RFC 4566 [43].

2.4.2 Real-time Transport Protocol

The RTP [42] is an application layer protocol developed for providing real time data through the IP enabled networks among many users. Real time data, such as voice, are greatly affected by transmission delays something that imposes a significant impact to the quality of the provided services. The RTP streams contain sequence numbers and therefore the missing packets can be detected. Taking into account that in real time packets cannot be retransmitted (time sensitive), RTP does not provide flow control or any re-transmission support. Also, it does not support a mechanism for ensuring the accurate time that data reach their destination or any QoS reservation procedures. However, such capabilities can be provided by the underlying services that utilize the RTP. Finally, timestamping is provided by RTP. This enables the synchronization among different media stream that may require concurrent playback. For instance, a video may be transmitted with three different audio streams (e.g. English, Greek and French dubs) and thus the timestamps can associate audio with video packets for providing accurate synchronization among them. The RTP is encapsulated in the payload of the UDP packets. Its header has 32bit length and the header values are briefly described in Table 2.9

2.4.3 RTP Control Protocol

The RTP Control Protocol (RTCP) [42] is used in conjunction with the RTP and does not bear any media data. It is periodically transmitted among the communicating entities to provide quality of service information, packet delivery/loss detection and session control information.

Specifically, it supports network diagnosing as data monitoring functions in order to observe

Table 2.9: RTP Header Fields

Header Field	Description
V	RTP version. V=2 for the RFC's 3550 version
P	Padding octets that are not part of the payload
X	Indicates the support of header extensions
CC	The number of sources (up to15)
M	Marks the start or the boundaries of a stream
PT	The payload type format (audio/video codecs)
Seq. Numb.	Indicates the sequence number of the packet
Timestamp	The timestamp of the packet
SSRC	Identifies the ID of source transport address
CSRC	Identifies the IDs of contributing sources in the payload

packet loss or congestion events and to asses its impact and origin (local or global). A mechanism that multicasts such information is also supported to inform all the participant about such events.

A canonical name (CNAME) is also included in the RTCP packet. This is actually an identifier in "user@host" format for avoiding de-synchronization of the source when the random SSRC identifier may change due to a restart or conflict. The CNAME is unique identifier and is bound with a specific participant (the receivers need to associate RTP streams to a participant).

Since all the communicating participants exchange control information, they can individually determine the exact number of them. This is a significant information for scaling up or down the rate of packets being sent according to that number. Finally, it optionally provides human readable identification of the participant to all the devices.

2.5 Auxiliary Protocols

2.5.1 Media Gateway Control Protocol

The media gateways provide media connectivity between IP and PSTN networks (including the analog interfaces RJ11, cable modems, xDSL devices through the residential, access and network access gateways [41]). Specifically, it supports the RTP media streaming from an IP network towards to a PSTN and backwards. These network devices act actually as translators among the different type of data and are located at the entry points of the IMS/VoIP core networks.

The MGCP [41] consist of two elements: the media gateway controller or call agent and the gateways. In contrast with the SIP, the MGCP is a master/slave protocol and therefore the slave devices can transmit information or execute commands only when they are directed by the master device.

2.5.2 H.323

The H.323 [40] protocol has been the main adversary of SIP for the main VoIP signaling protocol until the 3GPP's decision to include the latter in its specifications for the IMS. Its capabilities are not limited to IP but also supports ATM and PSTN networks.

The H.323 is comprised by four network elements in order to provide telephony services:

- Terminals
- Gatekeepers for signaling services
- Gateways for providing connectivity with other networks
- Multipoint Control Units (MCUs) for conferencing services

A registration procedure is required by the user in order to be able to access the service. The gatekeepers are responsible for receiving and processing registration requests. When the user does not know the address of the gatekeeper, he sends the request to the multicast address. The gatekeeper precesses the request and responds with a Gatekeeper Confirmation (GCF) to inform the user for its address. Afterwards the user sends the registration request to the received gatekeeper's address where the latter accepts the registration by sending a Registration Confirmation (RCF).

A registered user may send an Admission Request (ARQ) to the gatekeeper in order to establish a session with another user (callee). The gatekeeper evaluates the access rights of the caller and the available resources for session handling and then responds with an Admission Confirmation (ACF) message to indicate the acceptance of the request. At this time, the caller sends a SETUP message to the callee inviting him to a call. When the latter receive the request, he responds with an ALERTING message accepting, in this manner, the call. The caller sends a connection message (CONNECT) and the media codecs negotiation now takes place.

The release of the call and consequently of the reserved resources, is achieved by sending the appropriate request to the gatekeeper while the latter forwards it to the callee.

2.5.3 Other Protocols

For providing telephony services, VoIP and IMS architectures utilize protocols that they have been initially developed for other infrastructures. These protocols are mandatory for the VoIP/IMS networks to function properly. The most important ones are the following:

- DHCP: It is utilized for initiating the user's device in order to have access to an IP network. The DHCP provides automatically all the required parameters to the device to function properly.
- Telephone Number Mapping (ENUM): It is responsible for translating telephone numbers to internet addresses and backwards. These addresses are stored into a DNS in Naming Authority Pointer Form (NAPTR) [61].

Chapter 3

IMS Architecture

3.1 Introduction

The IMS architecture is much more sophisticated and complex from the conventional VoIP deployments. There are numerous networking components (see Fig. 3.1) required even for the provision of basic functionality such as call establishment or instant messaging services. An advanced version of user's device replaces the User Agent (UA) in these environments. The User Equipment (UE) is a SIP enabled device, like the UA, but it is capable of executing QoS reservations and contemporary encryption algorithms. IMS deployments are comprised of QoS dedicated components, core signaling functions and databases while the interworking components are of major importance since they fulfill the all-IP concept.

Following the description of IMS's functionality presented in the introduction, this chapter provides a detailed description of the IMS architecture. Specifically, the core networking components, the gateways and the media/application components are presented in order to provide an overall picture of an IMS deployment and its topology. Furthermore, the role of the subscribers and their means of identification and profiling in such environments is discussed in conjunction with terms that are also bound with the IMS architecture such as: (i) service profiles, (ii) initial filter criteria and (iii) private and public user identities. All these components communicate by utilizing different interfaces, the reference points that specify the protocol that will be used among the communicating entities.

3.2 IMS Networking Components

According to [2] and [62], the IMS architecture consists of five main components, namely: (a) the call session control function, (b) the multimedia resource function, (c) the interworking components, (d) the application server, and (e) the databases (see Fig. 3.1). These components employ the Session Initiation Protocol (SIP) for their communication. Furthermore, it should be stressed that they can be logical entities and not only physical, as they may be implemented under the same physical network element.

3.2.1 Call Session Control Function

The Call Session Control Function (CSCF) is considered the core of the IMS architecture and is responsible for session management. It consists of three distinct elements:

1. The Proxy CSCF (P-CSCF) is the entrance point to the IMS services and routes incoming requests to the appropriate network entities. The registration and the other requests are forwarded, from this entity, to the server (S-CSCF) which has been assigned to the corresponding UE during the registration procedure. The connection is realized through the Gm reference point (see Fig. 3.1). When the user accesses the service through a UMTS network, the Gateway GPRS Support Node (GGSN) is being involved in the communication. It also maintains security associations with the UE, accomplishes SIP compression and decompression, handles billing-specific information (generates the Charging Data Records - CDRs) and emergency session requests. The specific network entity is responsible for ensuring that the given information about the users' location is up-to-date. It may incorporate the Policy Decision Function (PDF) for QoS reservation or it may communicate with it over the Diameter protocol [47] (Gq reference point - Fig. 3.1). Furthermore, it checks if the SIP messages include correct information about the time zone as provided by the access network. Finally, it may insert, in any message (request or response), the type of access network used by the UE.

2. The Interrogating CSCF (I-CSCF) communicates with the Home Subscriber Server (HSS), over the Diameter protocol (Cx reference point - Fig. 3.1), in order to assign a Serving CSCF for the registration of a UE. It is responsible for forwarding requests to home S-CSCF (also in cases where the messages originate from another networks).

   Based on the HSS's response, the I-CSCF may decide to forward the message, when the session terminates, outside the IMS or to reject it by sending the appropriate error status code to the originating entity. In general it acts as the contact point for connections established inside the home network or destined to a roaming user within the operator's service area. In earlier versions [2], it provided Topology Hiding Inter-Network Gateway (THIG) functionality through the encryption of headers which reveal topology specific information: "Record-route", "Via", "Service-Route", and "Path". It is also responsible for translating the E.164 addresses; converts the SIP URI into phone number format (Tel: URI format [63]).

3. The Serving CSCF (S-CSCF) is a SIP server and acts as registrar (as defined in SIP RFC 3261 [3]) during the registration procedures. It is responsible for the session handling and for the retrieval of the authentication vectors and user profiles from the HSS. In multi-HSS environments, it communicates with the Subscription Locator Function (SLF), over the Dx reference point (see Fig. 3.1), in order to locate the HSS that holds the data of a specific user. It also forwards the SIP messages to the Breakout Gateway Control Function (BGCF) to communicate with the PSTN network. When the SIP URIs of the communicating end points are located in the same network operator, it forwards the SIP

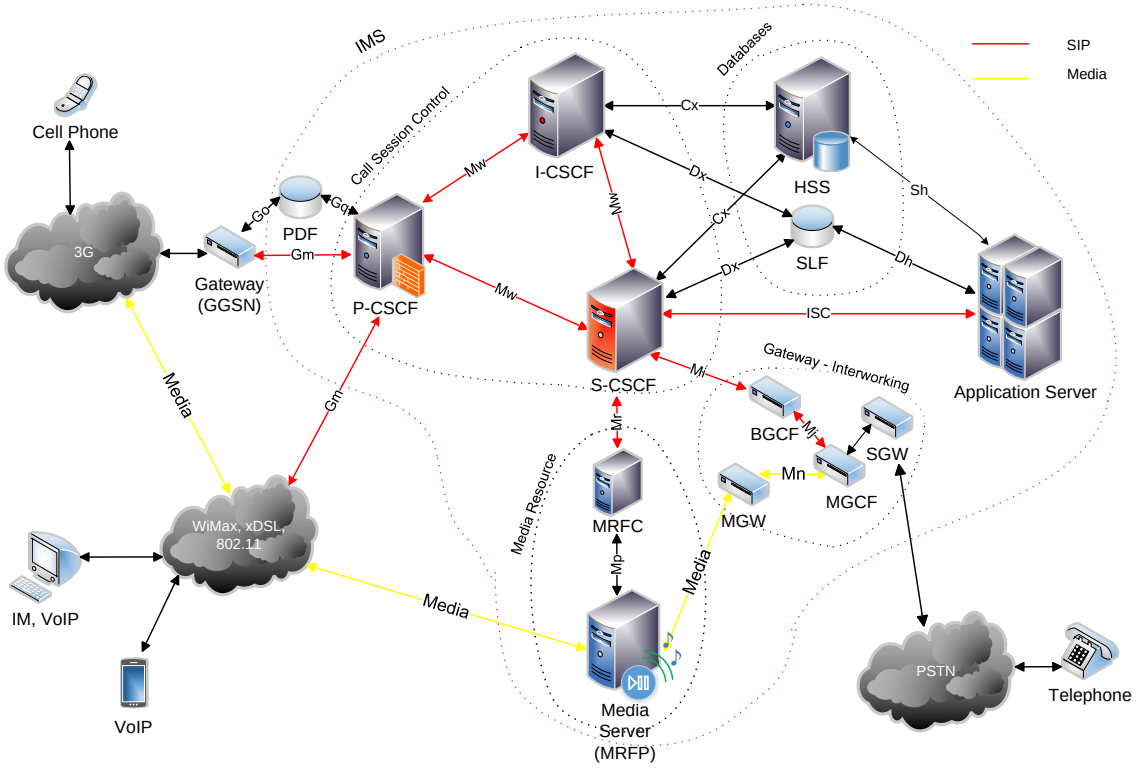


Figure 3.1: Basic IMS deployment.

signaling to the I-CSCF. It also forwards messages to other servers of the same ISP, located outside the IMS Core Network (IM CN) Subsystem. It also translates the E.164 addresses (international public telecommunication numbering plan) to a globally routable SIP URI. When it is incapable of this translation, it forwards the request to the BGCF in order to handle the routing toward the Circuit Switched (CS) network.

3.2.2 Multimedia Resources

The Multimedia Resource Function (MRF) handles the media services such as conferencing sessions, audio/video transcoding, and more. The MRF consists of two subcomponents: (a) the Media Resource Function Processor (MRFP) and (b) the Media Resource Functions Controller (MRFC). The first is responsible for the media stream mixing and provides media sourcing. It executes media analysis and transcoding and handles the shared resources in terms of access rights of the involved entities. The MRFC controls media resources and is responsible for interpreting the incoming traffic/signaling from an S-CSCF (over Mr reference point - see Fig. 3.2) or an Application Server (over Mr' reference point - see Fig. 3.2) in order to control the MRFP [62]. The communication between the two MRFs is achieved through the Mp reference point which is fully compliant with the H.248 protocol [64]. The Cr interface enables the interaction between the Application Server (AS) and the MRFC in order to exchange control notifications, requests and documents [65].

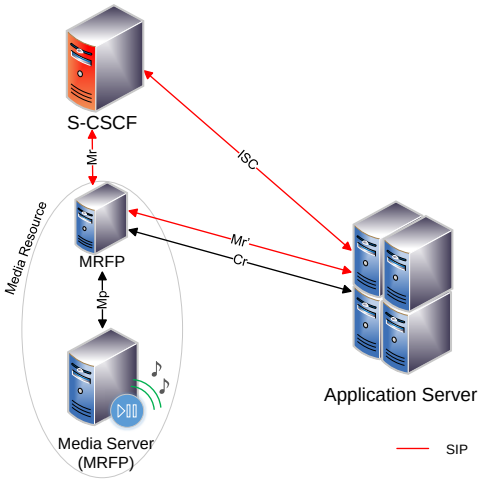


Figure 3.2: Media resources.

3.2.3 Interworking Components

The IMS architecture, in order to achieve interworking, must communicate with existing telecommunication networks (e.g., Public Switch Telephone Network (PSTN), mobile networks, etc.). For this reason, the IMS architecture includes specific components that perform media and signaling conversion between circuit switched (or mobile) and IP networks. Specifically for routing a request, signaling data are forwarded through the BGCF which accepts the SIP signaling from the S-CSCF and determines the next hop in the PSTN. On top of that, it also locates an appropriate Media Gateway Control Function (MGCF), which is responsible for mapping different signaling protocols between the IMS and the PSTN network. When the message destination is outside the home network, the BGCF forwards them to the BGCF of the terminating network (CS domain) or to the I-CSCF of the terminating IMS network. Similarly, the Media Gateway (MGW) converts and transcodes the media exchanged between the IMS and the circuit switched network entities. Finally, the Signaling Gateway (SGW) provides bidirectional signaling conversion between the IP and SS7 transport protocols (SCTP/IP and MTP - see Fig. 3.3) [66].

When the requests' destination is located to some other domain, the signaling is sent to the Interconnection Border Control Function (IBCF) which forwards them to the entry point of the terminating domain (Fig. 3.4). When the IBCF receives a request, it selects the appropriate signaling protocol and invokes the Interworking Function (IWF) for dealing with them (e.g. SIP and H.323). Further, it supports session establishment between different IMS networks with different media codecs and provides THIG. The IBCF acts as the entry and exit point to other IMS Cores and SIP networks and provides border control functions through the Mx and Ici reference points (see Fig. 3.4) correspondingly [65].

3.2.4 Application Server

The Application Server (AS) provides the IP multimedia services (such as presence, call conferences, instant messaging, push to talk and many more) to the home network or to other

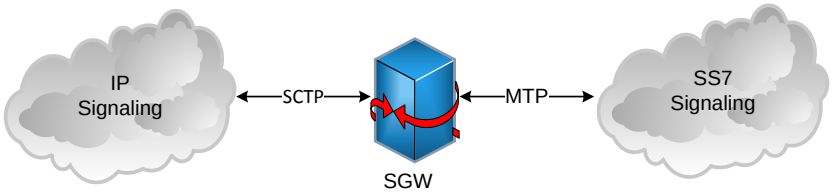


Figure 3.3: Signaling gateway to legacy networks.

locations. It can also provide OSA (Open Service Access) services in the IMS and a gateway to legacy service networks, such as Customized Applications for Mobile Network Enhanced Logic (CAMEL) [67], through the deployment of the corresponding servers. Further, the IP Multimedia Service Switching Function (IM-SSF) acts as an interface between the IMS domain and the CAMEL services used in 2G networks. An AS utilizes the Diameter protocol for communicating with the HSS or the SIP protocol for communicating with the S-CSCF. The AS may also interface with the MRFC for media control.

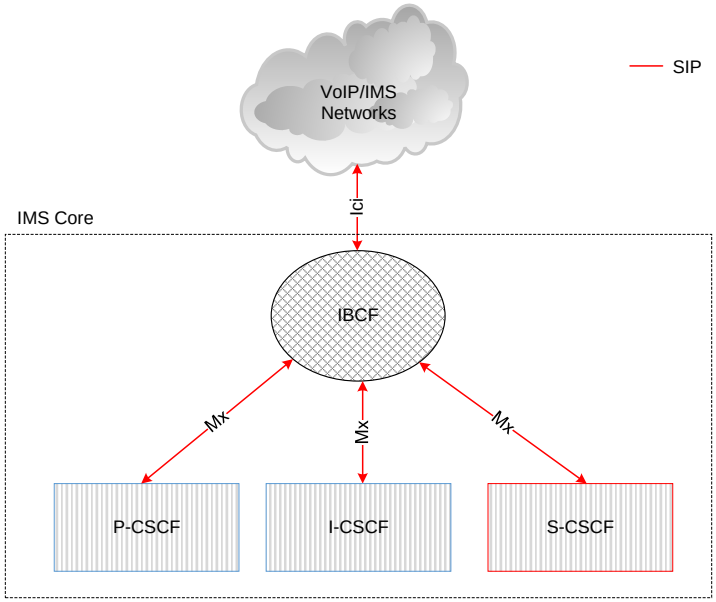


Figure 3.4: IBCF and other networks.

3.2.5 Databases

The Home Subscriber Server (HSS) is the main repository of IMS subscriber's data. It holds users' profiles, location, security information (generates authentication, integrity and ciphering), and any other information required for service provisioning. The HSS is contacted by other network elements (e.g., S-CSCF or I-CSCF), whenever an incoming request requires authentication or authorization, according to the provider's policy, by utilizing the Diameter protocol [47] over the Cx Interface (see Fig. 3.1). An AS may also contact the HSS (using Diameter protocol - Dh reference point) when subscriber's data are needed for the provision of a service. In multi-HSS environments, the SLF is deployed to provide the name and the specific

location of the HSS, which holds the data for a given user. The SLF is contacted by the I-CSCF or the P-CSCF over the Diameter protocol (over the Dx reference point).

3.3 Subscribers and Identification

The users in every networking environment are uniquely identified by an identity. For instance, the phone number acts as a user identity in the PSTN networks. Likewise, a user can be identified with several different ways in IMS networks. In addition to the users, in IMS there is need to identify services and UEs. The identities utilized are: (i) IMS Private User Identity (IMPI), (ii) IMS Public User Identity (IMPU), (iii) Anonymous User Identity, (iv) Globally Routable User Agent URI (GRUU) and (v) the Public Service Identities (PSIs).

3.3.1 IMS Private User Identity

The private identity is not selected by the user but it is generated and assigned to a user's subscription by the network operator. The validity of the IMPI lasts as long as the user remains a client to the specific network operator. It is used for accomplishing authorization, authentication, administration and accounting purposes and not for message routing. The private identity is included in requests requiring authentication. For instance, REGISTER SIP messages include the authorization header which carries the user's private identity. The IMPI is stored in the ISIM or in the IMS Credentials (IMC) when the 3GPP access is not supported by the UE. From the operator's side, the IMPI is stored in the HSS where the P-CSCF will seek for it in order to gather the corresponding authentication vectors for user's authentication. The IMPI formatting conforms to the Network Access Identifier (NAI) [68] format (NAI = username / (username "@" realm)). A valid format is the "nvra@testbed-ims.gr" while "nvra@testbed_ims.gr" and "nikos;vra@testbed-ims.gr" are not. When the UE lacks ISIM, then the private identity is derived from the International Mobile Subscriber Identity (IMSI). The IMSI consist of three concatenated numbers: the Mobile Country Code (MCC) which is a three digit number that uniquely identifies a country, the Mobile Network Code (MNC) which defines the network operator (two or three digit number) and the Mobile Subscriber Identification Number (MSIN) that denotes the subscriber. The length of the IMSI cannot exceed 15 digits [69].

3.3.2 IMS Public User Identity

Every user in IMS may have one or more IMPUs, which are stored in the HSS. The relationship between public and private identities is depicted in Fig. 3.5. This diversity of the identities gives the opportunity to the subscribers to have different identification per service (different ID for the IPTV, instant messaging, games, etc.). The user needs the public identity of another user in order to contact him. The ISIM or the IMC (when the UE does not support 3GPP access) is responsible for storing the identity of the user. The users must be registered first with one of their own IMPUs in order to be able to initiate a multimedia session. The public

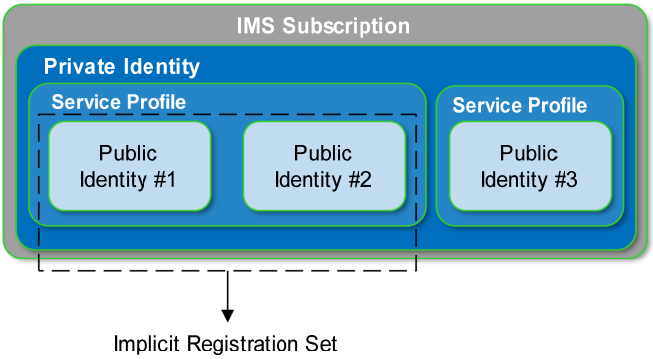


Figure 3.5: Public and private identities in IMS.

identities can also be implicitly registered[1]. The set of all public identities associated with a user and taking part in a single registration procedure, is called implicit registration set. In such a case, it is mandatory at least one of the public identities to be taken from a SIP URI. According to IMS specifications [69], the IMPUs are not authenticated by the network and they are included in the "From" SIP header. They are in SIP URI form or in tel URI format simply as a telephone number. The SIP URI format includes the protocol, the username and the domain (e.g. "sip:nvra@testbed-ims.gr"). The tel URI has the format "tel:+30-211-0116521", where the hyphens separate country, area code and subscriber number (the hyphens are optional and are used only for improving the readability of the numbers) [63].

3.3.3 Anonymous User Identity

When a user requires a basic level of anonymity (i.e. the callee will not be able to reveal the caller's identity or telephone number), the anonymous user identity can be used. It is in SIP URI format containing the username at a domain ("sip:username@domain"). Both user and domain parts are fixed. The string "anonymous" is used for the username while "anonymous.invalid" denotes the domain (e.g. "sip:anonymous@anonymous.invalid").

3.3.4 Globally Routable User Agent URI

As stated in the previous section, a user may have more than one public identities registered at the same time. Consequently, many UEs are probably used by the specific user at the same time accomplishing different tasks. For instance, a user may choose to have a UE only for accepting instant messages while some other more powerful UE is used for handling the more resource demanding operations such as the video conferencing or gaming. The GRUU [70] is used for identifying which UE utilizes the specific public identity. There are two types of GRUUs, the temporary (T-GRUU) and the public (P-GRUU). The temporary one lasts until the explicit de-registration of the specific identity while the P-GRUU has a more permanent duration since it remains valid as long as a combination of a UE and IMPU is valid. The P-GRUU reveals the IMPU of the user while the T-GRUU does not include any identities. The GRUUs are included

[1]Implicit registration is the registration where the user may be registered with two or more public identities within a single registration procedure.

in the "Contact" header and are defined by the parameters "temp-gruu" and "pub-gruu" which contain the temporary and the public GRUU correspondingly.

Table 3.1: Identity Examples

Identity	Format
IMPI	nvra@testbed-ims.gr
IMPU	sip:nvra@testbed-ims.gr
Anonymous	sip:anonymous@anonymous.invalid
GRUU	sip:nvra@testbed-ims.gr;gr=urn:uuid:c80cbf70-11e1-87e0-e1739504534c
PSI	sip:chatgroup@testbed-ims.gr

3.3.5 Public Service Identities

Public service identities [69] are utilized for the identification of services, so that the users would be able to access them. For example, an instant messaging service may use a PSI. This PSI has to be contacted by a number of users in order to establish a group messaging session. Other cases where PSIs are employed are conferencing and group services. The PSIs are stored in the HSS and take the form of a SIP or Tel URI such as the public user identities (e.g. "sip:chatgroup@testbed-ims.gr").

3.4 Profiles

All the user specific data are included in a structure called user profile that is stored in the HSS. Therefore, the user profile contains all the public user's identities that are associated with a specific private identity. All these data are available to the S-CSCF through the Cx interface (Fig. 3.1). The user profile also contains the service profile (see Fig. 3.6) which includes all the services and applications that users are allowed to use according to their subscription. The service profile contains the public identification, the core network service authorization, the initial filter criteria (iFC) and the shared initial filter criteria sets [71].

The public identification contains all the public service and user identities associated with the specific service profile and other fields such as the display name of the public identities owner and the number of simultaneous registrations that are allowed for a specific identity

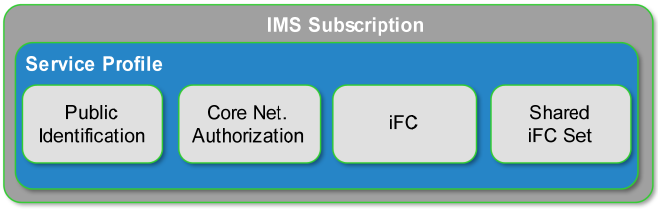


Figure 3.6: The IMS service profile.

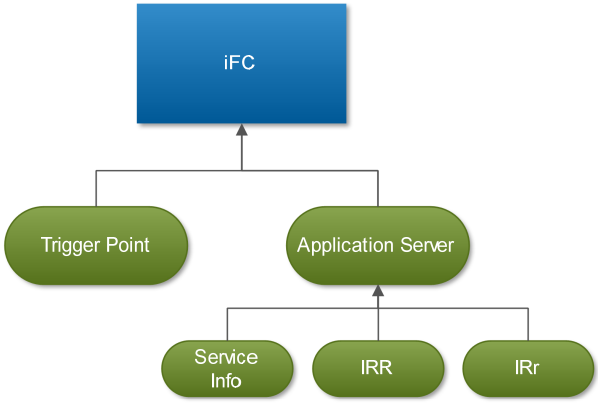


Figure 3.7: Initial filter criteria structure.

[72]. The core network service authorization contains the media profile of the subscriber. The media profile indicates the codecs and generally the SDP parameters allowed for a subscription. Specifically, it defines whether a user may have access to applications, audio, video or to all of them. The set of these parameters is denoted by a value, the subscriber media profile identifier. The S-CSCF is able to determine if the subscriber is authorized to use the requested parameters (e.g. video codec) by checking this identifier.

The initial filter criteria contain the required information for contacting an AS. These criteria are stored in the HSS and retrieved by the S-CSCF during the user's registration procedure. The S-CSCF, by evaluating the iFC, can determine how to route the SIP request and which AS is responsible for handling the specific session. Therefore, iFC provides the association of a set of public user identities with services and applications and they are valid until user's de-registration. Also, when the user profile is modified, the associated iFC will be also updated. The iFC structure, presented in Fig. 3.7, contains many different fields:

- The "iFC priority" value field: Determines the order in which the iFC will be evaluated in cases of many iFC.
- The address of the AS in SIP URI format.
- The "service information" field: It can be included only in registration requests and contains information that is not parsed by the CSCF or the HSS (may contain the IMSI).
- The trigger points: They consist of logical expressions that form a set of individual Service Point Triggers (SPTs). The selection or not of a particular AS is derived by the evaluation trigger points.
- The "include register request/response" (IRR/IRr) field: When the iFC are satisfied, it informs the S-CSCF that the incoming REGISTER request or the final SIP response to that request will be forwarded to the AS (as an attachment in the message body of a third party request).
- The default handling set: When the AS is not accessible for any reason; the default

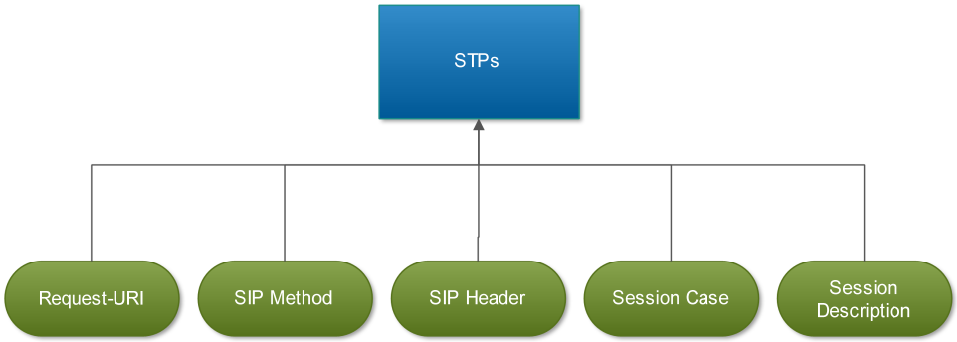


Figure 3.8: Service trigger points structure.

handling rules are executed. These rules may indicate that the evaluation shall be continued with the rest of the criteria with lower priority or that the session shall be dropped.

The SPTs (see Fig. 3.8) contain the SIP method of the request, the type of registration (re-registration, de-registration or initial registration), the SIP header which indicates whether a header is missing or not, the session case which defines the direction of the call (whether the served user is the caller or the callee and if the session is initiated or terminated to/by a registered user) and the session description information which includes the STPs for handling the SDP content of the SIP messages [72]. An example of a trigger point is the logical expression presented in Function 3.1. The atomic expressions are the STPs.

$$\begin{aligned}(\text{Method} = "INVITE" \ \text{OR} \ \text{Method} = "MESSAGE") \ \text{AND} \\ (\text{NOT} \ \ \text{Header} = "to" \ \ \text{Content} = "nvra")\end{aligned} \tag{3.1}$$

According to Function 3.1, a pre-specified action will be executed for every message which has the method INVITE or MESSAGE in the first line of the SIP signaling and does not includes the header "To" and the public identity "nvra".

Finally, the shared iFC set is an optional feature that facilitates the management and the evaluation of the iFC. Specifically, the iFC that overlap among many different service profiles are categorized in sets and stored in a local database. These sets are mapped with a unique identifier and thus the S-CSCF does not have to retrieve the specific iFCs from the HSS every time, but only the identifiers of a set.

3.5 Session Control

The session handling procedures are more complex in IMS environments, as compared to VoIP infrastructures. The main reason is the number of network entities involved in the handshakes for assuring better quality of service and providing an improved security level.

3.5.1 Media Session Establishment

The media session handshake is illustrated in Fig. 3.9. The INVITE SIP request is utilized by the UE which needs to establish a media session. This message (1) is sent to the P-CSCF including the UE's media capabilities in the SDP of the message's body. The P-CSCF, after receiving the INVITE request, responds (2) with a "100 Trying" to acknowledge its reception. The P-CSCF appends its address in the "Record-route" and "Via" headers and adds the charging information in the message ("P-Charging-Vector" header). Afterwards, it forwards the message to the S-CSCF. When the former receives the INVITE, it responds with a "100 Trying" (4) and proceeds with the iFC evaluation (5). During the iFC evaluation, the server determines the suitable media codes and discards the rest of them. At this time, the phone number is converted to a SIP URI and if the conversion cannot be achieved, the message is forwarded to the BGCF for resolving the routing towards a PSTN network [19]. Next, the S-CSCF executes DNS queries (through the ENUM-DNS [73]) in order to determine the location (the service provider of the callee) of the terminating user. Subsequently, it adds the network's identifier ("orig-ioi" header filed), in the form of the Inter Operator Identifier (IOI), to the INVITE message (6) and forwards it to the discovered I-CSCF. The I-CSCF, after responding with a "100 Trying" (7), communicates (over the Cx interface) with the HSS (8) to discover the S-CSCF that is associated with the specific user.

The HSS responds with a DIAMETER Location-Info Answer (LIA) [74]. The I-CSCF by receiving the LIA, extracts the address of the terminating S-CSCF from the "SIP-Server-URI" DIAMETER field and adds it to the Record-route header in the SIP INVITE message. Afterwards, it sends that message to the S-CSCF (9). The latter, after responding with a "100 Trying" message (10), parses the service profile of the terminating user and evaluates the iFC (11) in order to decide how to handle the request (determines the appropriate AS for serving the session). Subsequently, the S-CSCF responds with a "100 Trying" (11) to the I-CSCF and sends the INVITE request (12) to the specific P-CSCF which the callee has contacted during his last registration procedure. The P-CSCF authorizes the media resources (14) for the current session and sends the request to the callee (15), while the latter responds with a "100 Trying" to the S-CSCF.

The callee, upon receiving the INVITE request, responds to the P-CSCF with a "100 Trying" (16), to inform that the message has reached the UE. Further, the UE 2 includes its SDP answer and the signaling path in a "183 Session Progress" (17) SIP message and sends it back to the P-CSCF [75]. The latter, forwards the message (18) to the S-CSCF, which inserts its network IOI ("term-ioi" header field) to facilitate the data exchange between the operators for charging purposes. This response reaches the P-CSCF of the originating network by traversing back the route of the INVITE message (19a-19.c). The P-CSCF authorizes the media resources (20) and adds the media authorization token to the "P-Media-Authorization" header [76] and sends the messages to the call initiator (21).

Upon receiving the 183 response with the media offer from the UE 2, the originating UE has all the required information for determining the set of the audio/video codecs which will be used after the session establishment. Then, it acknowledges the reception of the 183 response with a

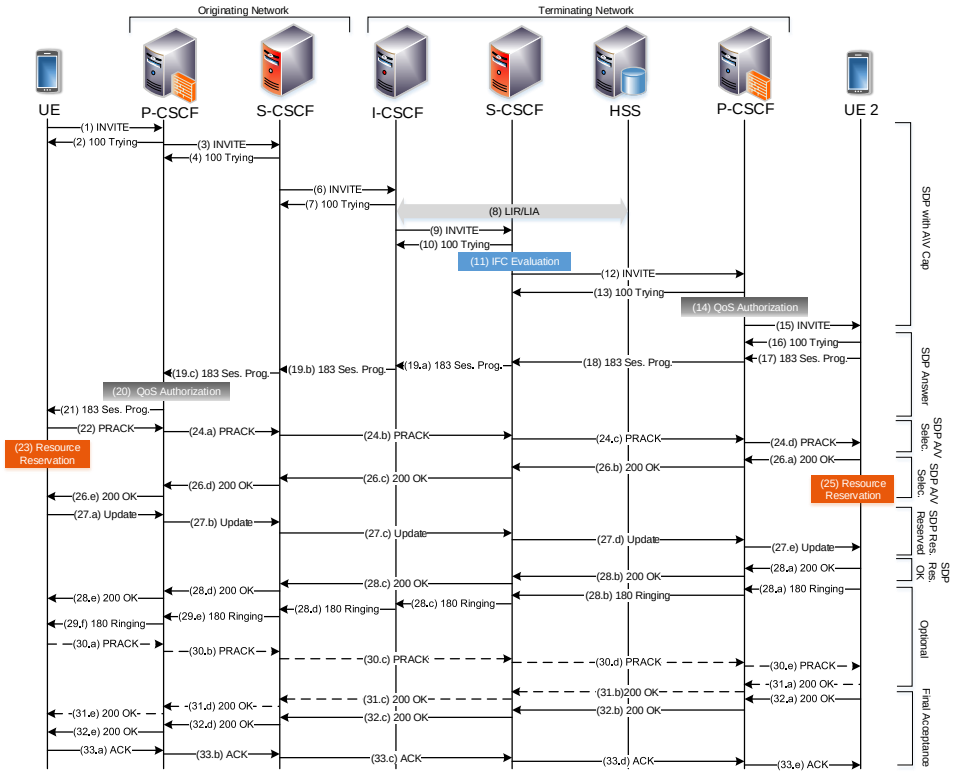


Figure 3.9: Session establishment procedure in IMS.

PRACK request (22) which may contains a new SDP offer in cases where it has selected more than one codecs for the negotiated media session. At this time and while the media codecs have been decided, the UE starts reserving the required resources (23) for the media session. The P-CSCF forwards the PRACK to the remaining network entities until it reaches the UE 2 (24.a-24.d). The UE 2 extracts the information indicating the media codecs, starts the resource reservation (25) and acknowledges the reception of the PRACK with a "200 OK response" back to the originating UE (26a-26e).

When the UE receives the "200 OK", crafts an UPDATE SIP request to inform that the resource reservation has been successfully completed. This UPDATE message traverses the signaling path which is derived from the "Record-Route" header where every CSCF has added its address, and terminates to the UE 2 (27.a-27.e). The UE 2 acknowledges the reception of the UPDATE with a "200 OK" response to the originating UE (28.a-28.e). As soon as everything has been successfully accomplished, the UE 2 has to notify the UE that all required resources have been allocated and it is ready to establish the current session. Thus, it sends a "180 Ringing" message to the caller through the CSCF path (29.a-29.f). The caller now is informed that the UE 2 is ringing, and the first acknowledges that event with a PRACK request back to the UE through the CSCF path (30.a-30e). The UE 2 responds back, through the same signaling path, with a "200 OK" to indicate the reception of the PRACK request (31.a-31.e). The UE 2, as a final acceptance to the session establishment procedure (that has been initiated by the UE), responds to the first INVITE (1) with a "200 OK" (32.a-32.e) and the latter acknowledges its reception with an ACK request (33.a-33.e). From now on, the media traffic can take place between the UE

and UE 2.

3.5.2 Conference Establishment

A conference room is actually a call in which more than two members can participate at the same time. It is therefore a multi-party communication. The members of a conference (participants) can communicate using audio, video or text messaging (chat rooms). A conference room can be established by utilizing the INVITE SIP request and it is denoted by a PSI. The serving AS acts as the central point and it is called SIP focus [77]. Fig. 3.10 depicts a conference establishment handshake with the media server, in accordance to 3GPP's specifications [78].

Similarly to the session establishment procedure, the UE crafts and sends to the proxy an INVITE request (1) which contains the SDP offer, namely the media codecs that this device supports for the upcoming conference. This set of codecs actually determines the type of the session (e.g. video conference, audio conference etc.). The conference URI is included in the Request-URI header. The P-CSCF responds with a "100 Trying" (2) and forwards (3) the message to the S-CSCF. The latter, responds with a "100 Trying" (4) and evaluates the iFC (5) as described above. The S-CSCF forwards the INVITE (6) to MRFC which also responds with a "100 Trying" (7). The MRFC creates the conference URI (8) that will be utilized by the subscriber in order to join the conference, and opens a connection with the MRFP (through the H.248 protocol (9)). This connection informs the MRFP about the server's available codec sets and instructs it to reserve the conference resources. The UE is informed about the MRFC's available codecs along with its IP address (contact header, the conference's URI is not yet available to the UE) through the "183 Session Progress" (10.a-10.b, 12) response that follows the reverse signaling path towards the conference originator (i.e. the UE). Meanwhile, the P-CSCF executes the media authorization (11) by contacting the PDF. The UE, upon reception of the SDP offer, included in the 183 response, starts the resource reservation in the same way as described in session establishment handshake (see Section 3.5.1), and sends an acknowledgement (PRACK) to the MRFC using the same signaling path (14.a-14.c). The MRFC sends back a "200 OK" (15, 17.a-17.b) response to acknowledge the reception of PRACK and opens a H.248 connection (16) with the MRFP in order to indicate that the resource reservation process should be initiated for handling the conference.

As soon as the UE has reserved the required resources, it sends an UPDATE request (18.a-18.c) to the MRFC through the CSCF (the signaling is the same and it is obtained by the "Route" header). The MRFC responds with a "200 OK" (26, 28.a-28.b) from where it can be derived that the resources have been successfully reserved for both end points (from the SDP protocol). Also, through the H.248 protocol (27), it opens the multimedia resources connection in MRFP for the UE. Subsequently, the MRFC sends a "200 OK" (29.a-29.c) as a response to the first INVITE (1) request message including the conference URI in the "Contact" header. Finally, the UE sends an ACK (30.a-30.c) to the MRFC (through the CSCF route) and starts the media stream with the MRFP (31).

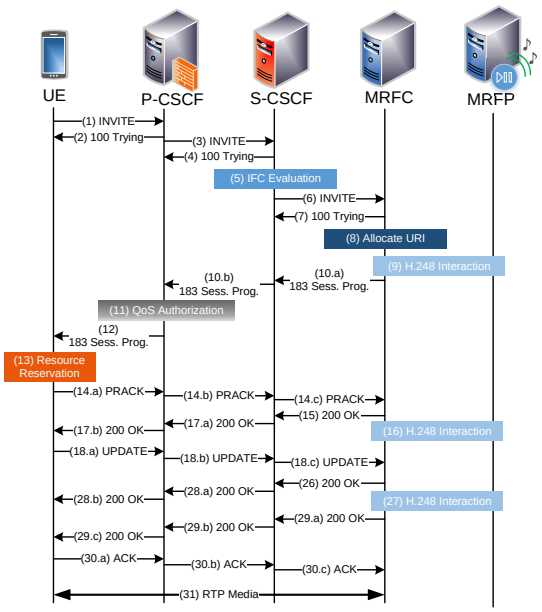


Figure 3.10: Media conference establishment in IMS.

3.5.2.1 User joins a Conference

A user may join a conference by following different procedures as described in the standard [78]. During the initial conference establishment handshake (see Fig.3.10), the conference initiator can include a list of desired users (in the form of SIP URIs) in the body of the first INVITE message. After a successful conference room establishment, and if a user knows the conference URI, he can send an INVITE request that includes that URI in the request line of the message (request URI), towards the conference's server.

A conference participant can also invite other users to join a conference. This can be achieved through two different methods. The participant may send:

- A REFER request to the conferencing server: The request includes the conference URI in the first line of the message while the user's SIP URI is included in the "Refer-To" header. This header may also include the parameter "method=INVITE". The "Referred-By" can be added indicating the participant, as described above.
- A REFER request to the desired user directly: The request includes the user's URI in the first line of the message and the conference's URI in the "Refer-To" header. This header may also include the parameter "method" set to "INVITE" (after a semicolon that denotes the end of user's URI). The "Refer-By" may be added optionally, in order to indicate the conference participant who invites the specific user to join the conference.

The case where a participant (UE) invites another user (UE 2) to join the conference is illustrated in Fig. 3.11. After the conference establishment, UE has obtained the conference URI and has started the media stream with the MRFP (1). The UE, in order to invite UE 2 to the conference, crafts a REFER request that includes the SIP URI of the UE 2 in the request URI field, the conference URI in the "Refer-To" header and the UE's URI in the "Referred-By" header.

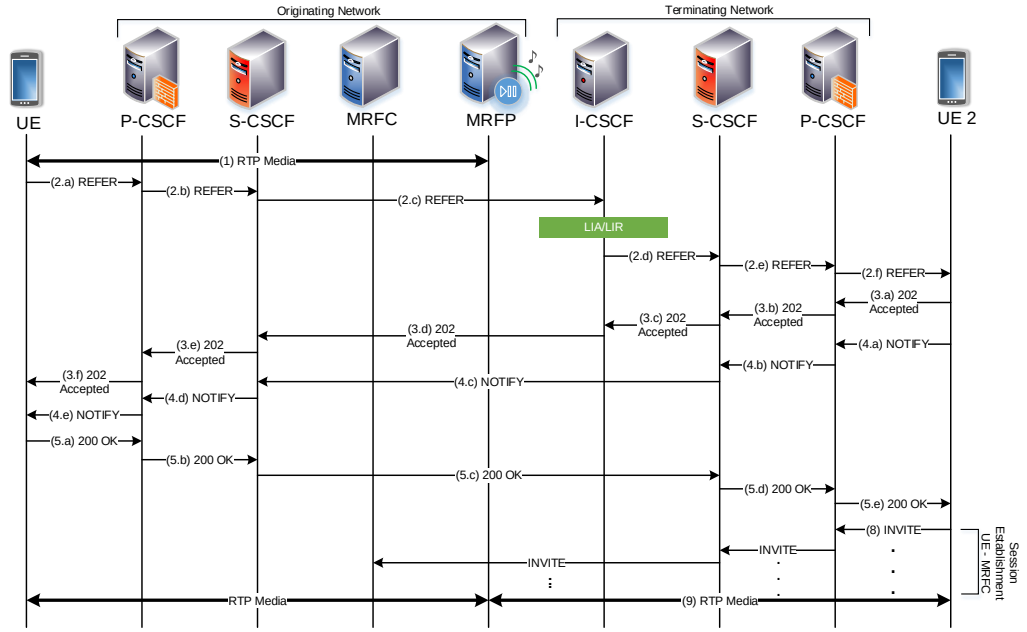


Figure 3.11: A participant (UE) invites another user (UE 2) to the Conference.

Afterwards, the UE sends the REFER request towards the UE 2 (2.a-2.f). The I-CSCF, of the terminating network, has executed a LIA/LIR (over the Cx interface) in order to determine the S-CSCF that is associated with the UE, as described in Section 3.5.1. When the UE 2 receives the REFER, it responds with a "202 Accepted" (3.a-3.f) to the UE (through the CSCF route) accepting the invitation. To indicate the status of the processed request, the UE 2 sends a NOTIFY SIP message to the UE (4.a-4.e). The UE, on the other side, acknowledgers the reception of the NOTIFY with a "200 OK" response to the UE 2 (5.a-5.e). Finally, the UE 2 starts a session establishment handshake (8) with the MRFC by utilizing the INVITE SIP method (as described in Section 3.5.1). At this time, UE 2 is ready to exchange media data with the MRFC and communicate with the other participants (9).

3.5.3 Session Termination

The session termination procedure is identical to the one in generic VoIP infrastructures. When a caller desires to terminate the current session, he sends the BYE SIP request to the callee. The handshake is depicted in Fig. 3.12.

The hang up of the caller's UE, automatically initiates the crafting of a BYE message. This message includes the Request-URI (first line of the message), the IP taken from the "Contact" header, while the headers "To" and "From" contain the caller's and the callee's identities respectively. The UE send the session termination request (1) to the P-CSCF which removes the authorization for resources (2) for the UE. The BYE traverses through the S-CSCFs and reaches the terminating network's P-CSCF (3.a-3.c) which executes authorization removal procedures (4) for the UE 2. Afterwards the BYE request reaches UE 2 (5). The latter responds with a "200 OK" that follows the reverse route of the BYE request toward the UE (6.a-6.e). At this time the media session is released.

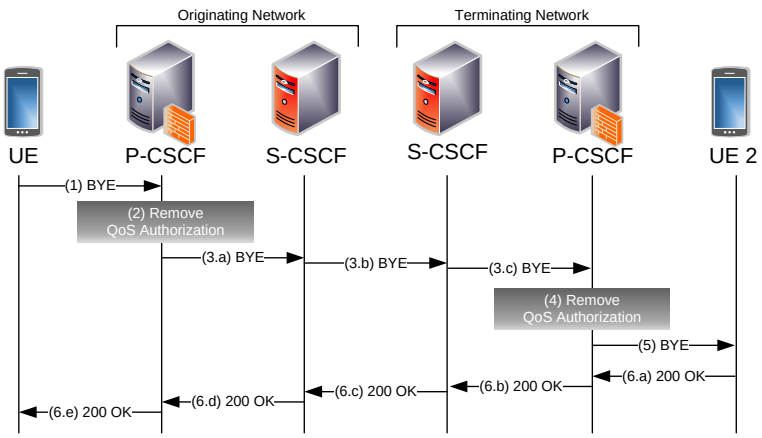


Figure 3.12: Session termination procedure in IMS.

Chapter 4

Authentication and Security Protocols in IMS

4.1 General Information

The need for security mechanisms in IMS environments is for exactly the same reasons as in every other information system: (a) to protect system's assets from unauthorized access, disclosure and modification, and (b) services availability. The assets can be physical entities such as the users and network elements, or non-physical such as the transmitted data. All these requirements must be firstly defined in order to come up with a set of security services that a mechanism shall provide. The most important security requirements in such environments include:

Data Confidentiality: The transmitted and stored data must be protected from unauthorized disclosure. Nobody but specific users (involved in the communication) and the core network can have access to the transmitted signaling data. Only specific core network elements can access the HSS database (stored data).

Data Integrity: The transmitted and stored data must be protected from unauthorized modification. Thus, unauthorized entities should not be allowed to modify the information exchanged among the communicating entities, neither the information stored in the HSS database. The latter could be a P-CSCF that adds its address in SIP signaling messages or a S-CSCF which can update a user profile stored in the HSS.

Entity Authentication: Users and network devices must be able to prove the validity of their identities. Every authentication mechanism must provide the means in order for a UE/user to be able to authenticate itself/himself to the network and vice-versa (mutual authentication).

Service Availability: The provided services must be available to the users at any time.

Users' Privacy: The users' private information shall not be disclosed. This information may

involve actions, locations, identities and personal data associated with a specific person.

4.2 Security Protocols in IMS

In IMS architectures, authentication takes place during the registration procedure. The SIP Digest, IMS Authentication and Key Agreement (IMS AKA) with the IPSec and TLS with SIP Digest are the main authentication schemes [18] that provide mutual authentication between the UE and CSCF components. Whenever authentication is required, the UE should notify the network side about the cipher suites that it supports. The supported security mechanisms are included in the "Security-Client" header [79] of the very first registration message, which according to IMS specifications, is unprotected. If this specific header is missing, then the GPRS-IMS-Bundled Authentication (GIBA) [80] or the NASS-IMS-Bundled Authentication (NIBA) [81] will be employed depending on the type of connectivity. It is worth mentioning that the IMS AKA and TLS authentication schemes provide integrity and confidentiality services to the communicating entities through the establishment of security tunnels.

4.2.1 SIP Digest

The SIP Digest [20] is a password-based challenge-response authentication protocol without any integrity or confidentiality provisions in communication (Fig. 4.1). It is utilized in the IMS whenever the UE lacks ISIM or USIM. The response of the network side to every non-authenticated registration request is a "401 Unauthorized" message, which indicates that authentication is required.

More specifically, whenever a UE sends a registration request towards the IMS network, the P-CSCF as the first contacting point receives and forwards it to the I-CSCF. The latter, locates and send that request to the S-CSCF which is responsible for handling the specific UE. The S-CSCF, in turn, determines the clients' capabilities in cryptographic functions (SIP Digest is indicated as MD5 in "algorithm" header) and retrieves from the HSS an Authentication Vector (AV) for the specific user identity.

The S-CSCF stores locally the AV and generates a unique data string (nonce) that, for instance, could be the result of hashing a timestamp concatenated with the server's private key. Afterwards, responds with a "401 Unauthorized" message (via I-CSCF), including the authentication parameters utilized by the UE to compute a valid authentication string for the next registration message. Note that the authentication parameters contain all the information (such as nonce, hash algorithm, realm, opaque, etc.) required for message authentication, according to RFC 2617 [18]. The opaque is a data string of the same encoding and is also generated by the S-CSCF. It must be returned unchanged to the S-CSCF in the "Authorization" header field of the UE's authentication response.

Afterwards, the UE extracts the authentication parameters from the response message (401 Unauthorized) in order to compute the credentials that will be used in the next UE's request,

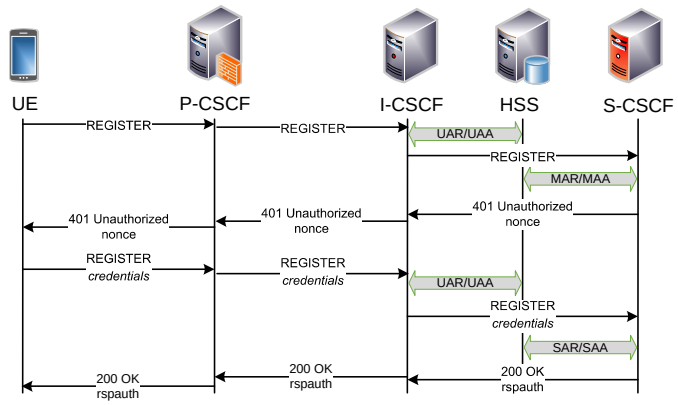


Figure 4.1: SIP Digest authentication in IMS.

utilizing the function 4.1:

$$H(H(A1),\ nonce\ ":"\ H(A2)) \tag{4.1}$$

where H is a hash function, $A1 = username$ ":" $realm$ ":" $password$ and $A2 = Method$ ":"$digest\text{-}uri\text{-}value$. In case the user has indicated a better protection level by setting the algorithm header to "MD5-sess", the A1 includes the nonce counters and calculated by utilizing the function 4.2:

$$A1 = H(username\ ":"\ realm\ ":"\ password)\ ":"\ nonce\ ":"\ cnonce \tag{4.2}$$

Whenever a S-CSCF receives an authenticated request (final response to the challenge), the S-CSCF analyzes it: Firstly, extracts the authentication parameters from the user's registration message and by using its AV and nonce, calculates the expected response. If the expected response matches user's response (included in the response header of the registration request) then the latter is successfully authenticated and the registration is completed. The S-CSCF obtains from HSS the user's profile and calculates the *rspauth* value that enables the user to authenticate the network (proof that that the server knows user's password). This value is included in the "200 OK" final response generated by the server. However, computed credentials are not included in the other parameters of the request instead of the AV, so the message is not protected against unauthorized modification and Man-in-the-Middle attacks (see Section 5.3.2).

4.2.2 SIP Digest with TLS

The SIP digest can be also employed in conjunction with TLS. The main difference in the TLS authentication scheme is that the second REGISTER message (which contains the UE's authentication string) is protected through the integrity and confidentiality mechanisms provided by the specific protocol. Initially, the UE indicates that supports TLS using the "Security-Client" in the registration message [79] while the server responds with a "401 Unauthorized" message that contains its security mechanisms in "Security-Server" header. After the reception of the 401 response, the UE initiates the handshake required for the TLS connection. P-CSCF and UE exchange their supported cipher suites and certificates as described in [17,18]. The authentica-

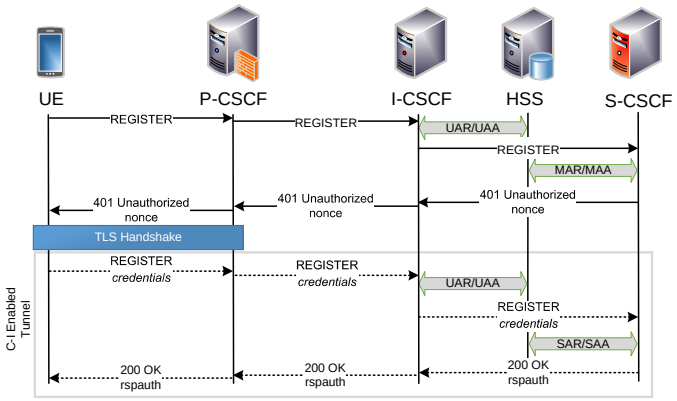


Figure 4.2: SIP Digest over TLS authentication in IMS.

tion of the server side is achieved through the use of valid certificates and then the security tunnels are established. Then the UE sends the response to the second REGISTER message, encrypted, and the authentication procedure continues in exactly the same way as in the SIP digest. All the traffic following the second registration message is encrypted and integrity is protected. The SIP Digest over TLS authentication procedure is depicted in Fig. 4.2.

4.2.3 IMS AKA with IPSec

According to the IMS specifications, the IMS AKA [18] is considered as the strongest authentication scheme that can be deployed in cases where a UE embeds an ISIM. Up to a point, this scheme is similar to the SIP digest; however, the IMS AKA establishes a secure tunnel between the UE and P-CSCF (see Fig. 4.3) that provides integrity and confidentiality services and also protects communication messages against attacks like the Man-in-the-Middle, eavesdropping (see Section 5.5), etc. Furthermore, all subsequent messages are protected through the same security tunnel. The selection of this mechanism is indicated by the "ipsec-3gpp" value in the "Security-Client" header of the registration message. Additionally, the first registration message embodies the "Security-setup" header which includes one Security Parameter Index (SPI) for every inbound (server and client modes of the specific UE) Security Association (SA), the IPSec modes and a list of security and integrity mechanisms supported.

As depicted in Fig. 4.3, in the IMS AKA, the S-CSCF retrieves from the HSS the AKA AV, instead of the SIP Digest AV, which is utilized to establish a secure tunnel between the UE and P-CSCF. The AKA AV consists of the concatenation of a random number (RAND), the expected response (XRES), the cipher key (CK), the integrity key (IK), and the authentication token (AUTN). When the P-CSCF obtains the AKA AV (without the XRES value) from the I-CSCF, which is included in the "401 response", it stores locally the IK and CK. At this point, P-CSCF allocates two unique SPIs for the Security Associations (SA) and forwards them to the UE through the "401 unauthorized" response without including the IK and CK. The SAs are established by selecting the encryption functions, supported by both UE and P-CSCF, giving priority to the ones that provide better security level.

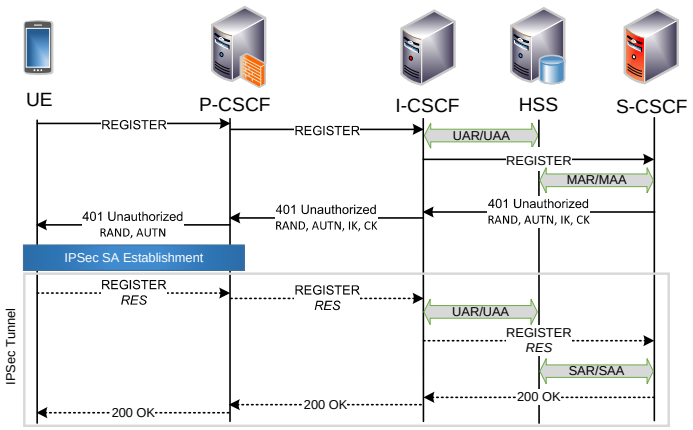


Figure 4.3: IMS AKA with IPSec authentication in IMS.

The UE, in turn, authenticates the server through the validation of the AUTN (note that AUTN includes a Message Authentication Code (MAC)). Afterwards, the UE generates the corresponding response message (RES) and computes the IK and CK to establish the SA and subsequently a secure tunnel (through the IPSec in ESP) between the UE and P-CSCF, which provides integrity and confidentiality services in the subsequent messages. The P-CSCF forwards the new authenticated request to the S-CSCF in order to check its validity. The S-CSCF authenticates the UE by comparing the RES and XRES. In case of a successful authentication (RES=XRES), the S-CSCF continues with the user's profile retrieval and terminates the procedure with a "200 OK" response. In contrast to TLS with SIP Digest, the final 200 OK does not contain the "rspauth" value because the UE has already authenticated the server during the validation of the AUTN. Further information about IMS AKA authentication and the generation of AVs can be found in [18, 82].

Chapter 5

IMS Security Level: Threats and Attacks

5.1 Introduction

IMS is an overlay IP network that is based on SIP for the administration of multimedia sessions. According to the literature [10, 11, 83–86], both the IP and SIP suffer from various attacks. In IMS, security mechanisms (see Chapter 4) are deployed in order to mitigate the impact of malicious behaviors on the messages' confidentiality, authenticity, and integrity, as well as on the architecture's availability and user's privacy. However, even the employment of the state-of-the-art security mechanisms cannot guarantee a satisfactory protection level against all type of threats. In the following subsections, IMS security flaws that compromise security requirements are presented, considering that providers have employed the security mechanisms described in the 3GPP's specifications [18] and presented in Section 4.2. The focus is on assessing the current security level of IMS infrastructures and on identifying the implications that such malicious actions can have for the users and the core network.

To facilitate the evaluation of the IMS security level, security mechanisms' capabilities are considered by taking into account the time frame of the attack. Since security mechanisms are not applied from the first phase of authentication, their efficiency against the same attack is evaluated, when the latter is developed during session and registration procedures. Additionally, the role of the attacker is taken into consideration since he can either be: (a) an External Attacker (EA) who launches attacks against the IMS infrastructure without having authorization to access the network, or (b) an Internal Attacker (IA) who is a legitimate user and thus authorized to access network services but acts maliciously. The basic difference between these two entities is that the latter has a legitimate subscription to the server and can authenticate the crafted requests. This fact enables an internal attacker to establish security tunnels with the proxies, authenticate malicious packets, and gather important information to launch attacks. On the other hand, an external attacker in many cases, is deterred when state-of-the-art authentication mechanisms are employed. Nonetheless, this is not the case when

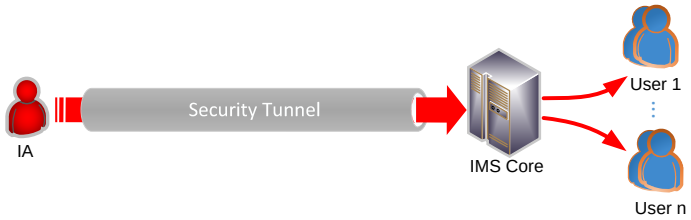


Figure 5.1: Example of an Internal Attacker (IA) in IMS.

the SIP DIGEST authentication is employed or the attack is launched during the registration procedure since the first message is sent unprotected. Taking Skype [87] as an example, an internal attacker could be a subscriber who tries to overload with calls and cause denial of service to a person in his contact list, while an external one could be a passive eavesdropper who tries to obtain a legitimate user's password by monitoring the communication with the Skype server.

Table 5.1 summarizes the current security state with respect to confidentiality, integrity, availability, accountability, and privacy from an attacker's perspective. The green color has been chosen for indicating the inability of the corresponding security mechanism to mitigate a specific attack while the red color denotes the robustness against that attack. Finally, the susceptibility to an attack due to partial conformance of the security mechanism to security specifications or due to the fact that the attack has been based on a series of assumptions is denoted with the yellow color. The objective of Table 5.1 is to provide a representation of what security and privacy requirements remain uncovered by either the specifications or other researcher's proposals. Moreover, it illustrates the possibility of security breaches and security requirements violation when different types of authentication mechanisms are utilized. The access level of the attacker is a critical point due to the fact that external entities may correspond to a larger set of potential attackers but they are deterred more easily than the internal ones. Another point which is evident from Table 5.1, is that the stronger authentication mechanisms cannot always offer a greater level of protection against internal attackers (users who at least are legitimately capable of authenticating requests). The specific attacks which can penetrate the core network's security deployments are also depicted in the specific table. Thus, it can provide a valuable insight on what security and privacy requirements remain uncovered (what is vulnerable, what vulnerabilities can be exploited, and by whom).

5.2 Attacks Against Integrity, Authentication and Availability

5.2.1 IMS SIP Signaling Attacks

The SIP specification [3] describes methods such as the BYE, CANCEL, and UPDATE, which are responsible for multimedia session termination, cancellation and modification respectively and which are utilized in the IMS as well. However, malicious users may manipulate such messages in order to cause denial of service to some specific session. The manipulated messages are identical to legitimate ones, instead of the fact that they have been generated by a spoofed

source. Consequently, the attacker compromises message authenticity as he "impersonates" a legitimate user. On top of that, the processing of such a (spoofed) message will cause illegal termination, cancellation or modification of a session that actually leads to loss of service availability. More specifically, an attacker can utilize the following SIP methods to compromise authentication and cause unavailability:

5.2.1.1 BYE Request

A SIP BYE method is a request message that a network entity or a UE utilizes for terminating a session. Nevertheless, from the attacker's perspective, this method offers a great opportunity for illegal session tear down.

This type of attack cannot be launched in the IMS architecture *during registration*, as register "termination" is accomplished through a specific REGISTER message (spoofed "Expires" header in REGISTER requests; see Section 2.3.1.3). On the other hand, *during session* and when the **(a) SIP Digest** is employed, every incoming request sent to the S-CSCF requires authentication; thus neither an internal nor an external attacker can extract the appropriate credentials unless they know a legitimate user's password. However, both of them could send the BYE request directly to the victim's UE that will force to close the open session while the UE may not authenticate this message. Furthermore, an internal attacker could also send a BYE request through the S-CSCF using his legitimate credentials (as a subscribed user), but spoofing the appropriate message headers and his public ID with the corresponding call initiator (see Fig. 5.2). Particularly, these headers include the "Call-ID" denoting the ID of the session that the attacker wants to terminate and the "From" that corresponds to the victim's ID. Note that the "From" header can be spoofed with the ID of the proxy when the malicious request is sent directly to the victim. As a result, the CSCF will successfully authenticate the attacker's BYE message and forward it to the target. This example can be mitigated by IP-check-tables, an optional mechanism that a P-CSCF can utilize to correlate the IP with the ID of every request [88].

When the **(b) IPSec** or **(c) TLS** is employed, a secure tunnel is established between the UE and P-CSCF, and consequently, an external attacker cannot eavesdrop or modify a message. So a legitimate user can encrypt/decrypt and validate only requests correlated with a specific session. This means that if an external attacker sends a direct BYE message to some UE it will be discarded as the latter does not know the IPSec cryptography parameters in order to participate in the established secure tunnel.

An internal attacker may forward the BYE requests to innocent users through his legitimately established IPSec tunnel (such an example is depicted in Fig. 5.1). In such cases, the internal attacker impersonates the ID of the call initiator (by spoofing the "From" header) and consequently terminates the session with the specific spoofed request. The S-CSCF can, however, check the correlation between the IP and the given public ID of the messages (cross layer SA-SIP correlation) from the SAs that have been established during the AKA in the registration procedure. Thus, it can reject such spoofed requests.

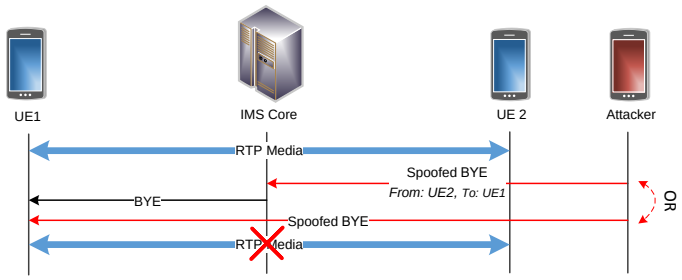


Figure 5.2: A basic example of an illegal session termination (BYE Attack).

Nonetheless, the SIP parser has also responsibility in this type of attacks because the attacker may bypass the SA-SIP correlation. This can be achieved by an internal attacker who uses his own public ID and IP and spoofs only the headers "From", "Call-ID", and "CSeq" in the BYE request and who forwards the message to the P-CSCF through his legitimate IPSec tunnel. An attacker cannot obtain directly the "Call-ID" to launch BYE attacks while the IPSec or TLS tunnel is established. However, he could try to predict one by gathering information regarding the victim's UE (type, model, revision, etc.) from the "User-Agent" header from the registration procedure and thus consider the procedure utilized by the specific UE for the generation of the "Call-ID" For example, some UE implementations choose the "Call-ID" value by computing the function: H(datetime)@host, where H is a hash function. Hence, an attacker can predict the specific time of a request and consequently the "Call-ID" value through traffic analysis. This message can pass the SA-SIP correlation because the internal attacker has used his legitimate IP and public ID. Finally, the message reaches the SIP parser. In many implementations the SIP parser correlates the headers "Call-ID", "CSeq", and "From" which denote a unique session. Therefore, the parser will decide to terminate the call which has the given "Call-ID" and "CSeq" headers.

5.2.1.2 CANCEL Request

According to SIP specification [3], the CANCEL method is used for cancelling (as its name implies) pending requests. It usually follows an INVITE request when the call initiator would like to cancel the session. A malicious user, similar to the BYE request attack, could craft a CANCEL request in order to illegally terminate a pending request, which compromises message authenticity and service availability. On the one hand, a CANCEL attack in the IMS cannot be launched *during registration* as the register "cancellation" is accomplished by sending a specific REGISTER message that includes an "Expires" header with zero value. On the other hand, a CANCEL attack could be launched for every other request (that could be cancelled according to SIP specification) *during a pending session*.

When the **(a) SIP Digest** is employed for the communication between a UE and the P-CSCF, according to the specifications, every new incoming request must have an increased "CSeq" header value as compared to the respective header value of the previous request. On the contrary, the CANCEL request must have the same value of the "CSeq" and "Call-ID" included in the pending request (e.g., INVITE) that is going to be cancelled. Therefore, the CANCEL

request cannot be authenticated unless credentials from the very previous request are included in it. As a result, either internal or external attackers can manipulate or generate the victim's IP, the "From", "Call-ID", and the "CSeq" headers of the CANCEL request to launch such an attack. Furthermore, an attacker can also send the CANCEL request directly to the target exactly as in the case of the previously described BYE attack. When the **(b) IPSec** and **(c) TLS** are employed the situation is exactly the same with the BYE attack (see Section 5.2.1.1).

5.2.1.3 Re-INVITE and UPDATE Request

The difference of the re-INVITE from the INVITE method is that the first one is used to modify existing session parameters instead of initializing a new session. It should be noted that the Re-INVITE's "CSeq" header value is incremented denoting a session's parameter modification. So an attacker can send a second spoofed INVITE (i.e., a re-INVITE) impersonating a legitimate user or a core network entity (by spoofing the "From" header), in order to illegally modify the session's critical parameters such as the port number, the address ("Contact" header) or even the media options (e.g., muting the session) and consequently causing DoS [13]. The UPDATE request has exactly the same usage as the Re-INVITE but the former is only used to modify an already established session while the first one may, for example, place a call on-hold before reception of the final response.

This type of attack cannot be launched *during registration* procedure. On the other hand, *during session* the attacker can launch such an attack even if the **(a) SIP Digest** is employed. As have been already explained (see BYE-Section 5.2.1.1), every incoming request can be authenticated by this mechanism. Consequently, a re-INVITE message should be authenticated, so an external/internal attacker can only launch such an attack in case he knows a legitimate user's password. For the **(b) IPSec** or **(c) TLS** employment, the only way for an external attacker to launch such an attack is through his legitimately established IPSec tunnel, impersonating (IP and public ID) as described in the BYE attack. These attacks can be launched in environments that do not apply strictly the SA-SIP binding check.

5.2.1.4 REGISTER Request

The REGISTER request is used by the UE in order to obtain access to the network services, which provides simultaneously its location and availability. Whenever the user would like to log out, he sends a specific REGISTER message (the value of "Expires" header is set to zero) to declare its intention, so the network de-registers the specific UE. An internal or external attacker might spoof such a REGISTER request to cause DoS or to gain unauthorized access to the provided service. This type of attack could be launched either *during registration* in which the internal or external attacker acts as the Man-in-the-Middle (MiM) (for more details about this attack please refer to Registration Expiration attack in Section 5.3.2.1) or *during session* in which the malicious user can launch a replay attack by sending an old REGISTER message after modifying the "Expires" header to zero value.

Even if the **(a) SIP Digest** is employed, as described in [89], the attacker can launch a replay attack by capturing the credentials of a previously authenticated message and crafting a REGISTER message in which the value of the "Expires" header has been set to zero (De-Register attack). Consequently, the network that is responsible to process such a request would authenticate it successfully. This attack can be achieved if the nonce value of the captured credential is valid. However, in the IMS, these attacks can be mitigated by the use of a nonce counter header [20]. The authenticating network is responsible for storing the previously used nonce to reject such attempts.

The employment of the **(b) IPSec** or **(c) TLS** provides integrity, authentication, and confidentiality services, as well as anti-replay protection. Thus, a malicious user (internal or external) could not capture a REGISTER message for the launch of REGISTER replay attack. Besides, anti-replay protection techniques, which are included in the IPSec and TLS, can protect the IMS architecture against replay attacks.

5.2.2 SIP Tampering Attacks

The SIP tampering attacks are based on the concept of injecting a part of malicious code in well-formed and syntactically correct messages. The consequences of processing such a message may be serious for the availability and the data integrity of the targeted system. The attacker can fake the authentication, drop tables, or even shut the HSS (HSS is the IMS database infrastructure) server down by inserting a malicious code in the "Authorization" field as presented in Table 5.2. The IMS inherits these types of vulnerabilities due to the utilization of SIP and its interconnection with database systems. Similar to SQL injection attacks in web services [90], as well as in SIP [19], a malicious user might accomplish such an attack also against the IMS.

More specifically, a SIP message may carry such malicious code and by traversing all the CSCF's entities can finally reach the database server. This can be done because the SIP parser of CSCF ignores the SQL statements, as this is in the authority of the HSS database. An internal or external attacker can break the authentication by stealing user passwords, or he can cause DoS by deleting database tables or by shutting down the server. Every header being processed by the database can be appropriate for malicious SQL code injection. The main concept is based on the fact that the quotation marks, like apostrophe and the interrogative mark, denote the end of a statement or a query in SQL. For example, the statement "SELECT password FROM Subscriber WHERE username = `'nvra'`", returns the password that corresponds to the username "nvra". The question here is how a malicious user can achieve such an execution in the database: In a field that requires the user name, the `"'"` will close the current statement and the SQL parser will wait for another command execution (in a normal situation the code execution should be terminated). Thus, considering the example of statement 5.1, the parser will execute the UNION command which has the harmful code. The double hyphens are of

Table 5.1: IMS Security Mechanisms from the Attacker's Perspective

				IMS Sec. Mechanisms					
				Digest		IPSec		TLS	
Req.	Threat Category	Time Frame	Attack	IA	EA	IA	EA	IA	EA
Int./Auth. & Availability	SIP Tampering	During Registration	SQL Injection						
		During Session	SQL Injection						
	SIP Signaling	During Registration							
		During Session	BYE						
			CANCEL						
			Re-INVITE						
			UPDATE						
			De-Register						
Integrity / Authenticity	Impersonation	During Registration	IMS Impersonation (IP/ID Spoof)						
		During Session	IMS Impersonation (IP/ID Spoof)						
	Replay	During Registration	-						
		During Session	-						
	MiM	During Registration	Registration Expiration						
			Bid Down						
			Generic MiM - Authentication Abuse						
		During Session	Bid Down						
			Generic MiM - Authentication Abuse						
			Conference Interception						
Availability	SIP Parser	During Registration	SIP Malformed, Sequence Disorder						
		During Session	SIP Malformed, Sequence Disorder						
	Flooding	During Registration	REGISTER, 401 Response						
		During Session	INVITE, INVITE Reflection, BYE						
			Response, Response Reflection						
Privacy / Confidentiality	Eavesdropping	During Registration	Session Parameters Disclosure						
			Privacy/Identity Disclosure						
		During Session	Session Parameters Disclosure						
			Privacy/Identity Disclosure						
Account-ability	Toll Fraud	During Registration							
		During Session	Bypass IMS - Cancel						
			Bypass P-CSCF						

The employed authentication mechanism is assessed considering the access level of the attacker (Internal Attacker [IA] or External Attacker [EA]). The security and privacy requirements which remain uncovered are illustrated by what is vulnerable, what can be exploited, and by whom. The green color indicates the inability of the corresponding security mechanism to mitigate a specific attack while the red color denotes robustness against that attack. The yellow color denotes the susceptibility to an attack due to partial conformance of the security mechanisms to security specifications or due to the fact that the attack has been based on a series of assumptions.

Table 5.2: Examples of SQL Injection During Registration

REGISTER sip: CSCF.unipi.gr SIP/2.0 Via: SIP/2.0/UDP [1050::::0005::120c:354b];branch=kfda5ss5d3 Max-Forwards: 70 From: <sip:nvra_private@unipi.gr>;tag=5d31 Call-ID: ak5fj49fhujDUuf0AjU9 CSeq: 1 REGISTER *Authorization:* *Digest username="nvra_private@unipi.gr '**; DROP TABLE Subscribers--**", realm="unipi.gr",nonce=" ", uri="sip:unipi.gr",* *response=" ", security-client: ...*
REGISTER sip:CSCF.unipi.gr SIP/2.0 Via: SIP/2.0/UDP [1050::::0005::120c:354b];branch=kfda5ss5d3 Max-Forwards: 70 From: <sip:nvra_private@unipi.gr>;tag=5d31 Call-ID: ak5fj49fhujDUuf0AjU9 CSeq: 1 REGISTER *Authorization:* *Digest username="nvra_private@unipi.gr '**; shutdown--**", realm="unipi.gr",nonce=" ", uri="sip:unipi.gr",* *response=" ", security-client: ...*

major importance and are used to comment out the rest of the code.

"Username: ' UNION SELECT password
FROM Subscribers WHERE username = 'nvra' --" (5.1)

An SQL injection can be launched either *during registration* or *during session*. *During registration* and when the **(a) SIP Digest**, or **(b) IPSec**, or **(c) TLS** is employed, this attack can be successful due to the fact that the first registration message is unprotected. This can be either performed by an internal or an external attacker. On the other hand, *during session* the employment of the **(b) IPSec** or **(c) TLS** can prevent these behaviors that are initiated from external attackers because they do not have access to the confidential data that are exchanged inside the tunnels and they can not establish one. Nonetheless, the internal user could act maliciously by injecting his SIP messages inside the tunnels.

5.3 Attacks Against Integrity/Authentication

5.3.1 Impersonation and Spoofing Attacks

In impersonation and spoofing attacks, a malicious user acts "on behalf" of a legitimate user, which bypasses the authentication mechanism. In the IMS architecture, this type of attack can be achieved either by crafting the SIP spoofed messages or in the IP layer during the IP allocation from the GGSN (Gateway GPRS Support Node). An *IMS User Impersonation* attack can be launched if a malicious user "discovers" a legitimate user's ID. The former can obtain the ID of a subscriber by eavesdropping through the communication channel.

The impersonation attack can be launched either *during registration* or *during session*. The employment of the **(a) SIP Digest** does not offer efficient protection due to lack of integrity mechanisms. For instance, as described below, an internal or external attacker may impersonate a legitimate user to receive the SIP Digest AVs in order to launch a Man-in-the-Middle attack (see Section 5.3.2). The employment of the **(b) IPSec** or **(c) TLS** deters such attacks because the response to the server's challenge is sent after the tunnel establishment, and consequently, it is encrypted. Moreover, during the AKA, the CK and IK are not given to the user and thus the attacker will not receive any critical cryptographic data if he challenges the server on behalf of his victim.

During session and even if the **(b) IPSec** or **(c) TLS** is employed, an internal attacker can impersonate the victim to access services free of charge: The attacker allocates legitimately his IP from GGSN. Afterward, he establishes a new IPSec or TLS tunnel though a successful registration procedure. From inside of this tunnel, he crafts a SIP request by changing only the public ID with the victim's one. As a result, the network will charge the victim for the provided services and not the attacker. These attacks can be launched in environments that do not strictly apply the SA-SIP binding check where the IP of the packet and the ID of the SIP message are correlated. An external attacker cannot launch this particular attack because he is not able to establish or access a security tunnel.

At this point, it should be stressed that even the tunnel, which is established after the user's registration, does not provide an effective solution if the cross layer binding between the ID and IP (public ID in SIP message and IP address in the IP packet) is ignored.

5.3.2 Man-in-the-Middle Attacks

In a MiM attack, the malicious users act in the middle of the communication path between the user and the server, without any of the participants being able to identify them (acting transparently). Various incidents based on this technique have been published [12, 91, 92]. The use of the Address Resolution Protocol [12] or Domain Name System [93] poisoning techniques enables the attacker to act as an intermediate. Particularly, the attacker changes the IP-MAC or the domain-IP associations correspondingly in order to redirect the traffic through him (acting as gateway) and gathers communication channel's data.

In fact, in VoIP/IMS infrastructures, after an ARP or DNS poising attack follows a SIP based attack, where the messages are manipulated, imposing further damage to the system. In conjunction with an impersonation attack (see Section 5.3.1) in SIP messages, the attacker can masquerade as a user or server at the same time. In this type of attack, the attacker bypasses both integrity and authenticity security requirements of the reference system. Consequently, a malicious user can not only impersonate the user and the network elements (as already explained), but can also gain unauthorized access to the provided services or even worse, cause denial of service.

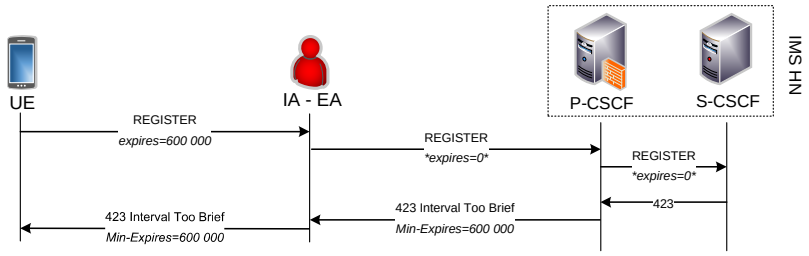


Figure 5.3: Register expiration attack mitigation in IMS.

5.3.2.1 Registration Expiration

The REGISTER message, among others, determines the registration's time period for a specific service (through the header "Expires"). Whenever the time period expires, the UE should repeat the registration procedure. Under this context, an attacker acting as an intermediate between the P-CSCF and the user, can modify the "Expires" header to zero value to de-register a legitimate user *during registration*.

Since the **(a) SIP Digest** lacks integrity mechanisms, either an internal or an external attacker could modify the "Expires" header field to the lowest possible value (e.g., zero), which will force the legitimate user to log out from the corresponding service. However, in order the IMS architecture to mitigate such type of attacks, the introduction of the "Min-Expires" header has been suggested. This header defines the lowest possible time interval for the "Expires" header according to the home network policy (for the initial registration procedure). As a result, if an internal or external attacker (since both have access to the unprotected REGISTER messages) modifies the "Expires" header in a lower value than the predefined by the network, the P-CSCF will respond with a 423 response message ("Interval too brief") including the "Min Expires" header (see Fig. 5.3), to inform the subscriber about the minimum acceptable value for the duration of the registration. The default minimum value (for the first REGISTER message) that is defined in specifications [94] equals to 600,000 seconds.

Although **(b) IPSec** and **(c) TLS** provide integrity, authenticity, and confidentiality services, an internal or external attacker could launch the attack during registration because the first REGISTER message is sent in clear text without any protection. Note that the secure tunnel is established after the UE sends the first REGISTER message. Nevertheless, to discourage such behaviors, the expiration of registration is defined in the second REGISTER request which takes place inside the security tunnel.

5.3.2.2 Bid-down

During the authentication procedure, the UE informs the IMS network about its capabilities with respect to the security mechanisms that can be utilized for the rest of the handshake. This type of information is included in the "Security-Client" header [79] by the UE. On the other side, the S-CSCF will parse this specific header to choose the strongest algorithm and accomplish the authentication procedure. However, a malicious user can act as intermediate in this negotiation

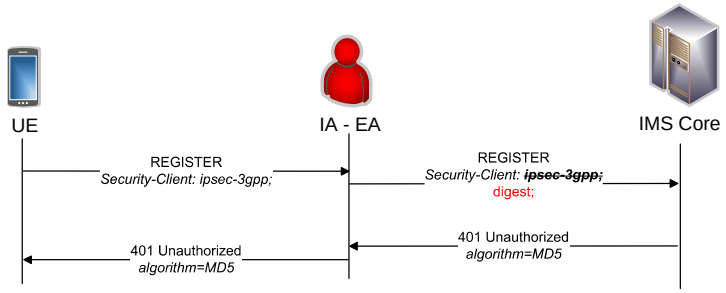


Figure 5.4: Security level degrading attack. The stronger security mechanisms are removed during the negotiation with the proxy.

between the UE and P-CSCF by removing the strongest security mechanisms [19] from the "Security-Client" header, downgrading the security level. An example of a bid down attack where an intermediate degrades the security level from IMS AKA to SIP Digest is depicted in Fig. 5.4. This type of attack could be launched either *during registration* or *during session*.

During registration, even when the **(a) SIP Digest**, **(b) IPSec**, or **(c) TLS** is employed, an internal or external attacker can accomplish this attack as the first registration message is in clear text form without any integrity protection. For this reason, the malicious user can remove completely the preferred, by the UE, security mechanisms (e.g., IPSec or TLS) included in the "Security-Client" header to downgrade the supported security level. For instance, the attacker may force the utilization of only the SIP Digest or weaker hash functions. Exactly the same applies *during session*. A possible solution against bid-down attacks could be the predefinition of the security mechanism: during the very first authentication procedure (initial subscription), the P-CSCF stores the strongest security mechanism that a UE supports. Thus, in all future registration requests, the P-CSCF shall know which mechanism to use.

5.3.2.3 Generic MiM: Authentication Abuse

The majority of MiM attacks can be launched in IMS environments by mainly exploiting the lack of integrity protection in the communication protocols. The utilization of existing security mechanism cannot always adequately handle MiM attacks.

More specifically, *during registration* and in case where the **(a) SIP digest** has been employed, an internal or an external attacker can impersonate both a legitimate user and the S-CSCF, abusing the Digest authentication [22]. Particularly, an attacker may spoof the legitimate user's IP/private ID and consequently send a registration request to the P-CSCF. Then the S-CSCF processes the incoming request and responds with a challenge message. The attacker, whenever he receives this response, acts on behalf of the P-CSCF and forwards this message to the legitimate user whose IP address/private ID has been spoofed "forcing" him to compute the appropriate credentials for this specific challenge response message. As the malicious user has an authenticated message based on challenge response sent by the P-CSCF, the malicious user is authenticated as a legitimate user. In case that the **(b) IPSec** or **(c) TLS** have been employed, an internal or external attacker cannot capture the authentication response from the UE since

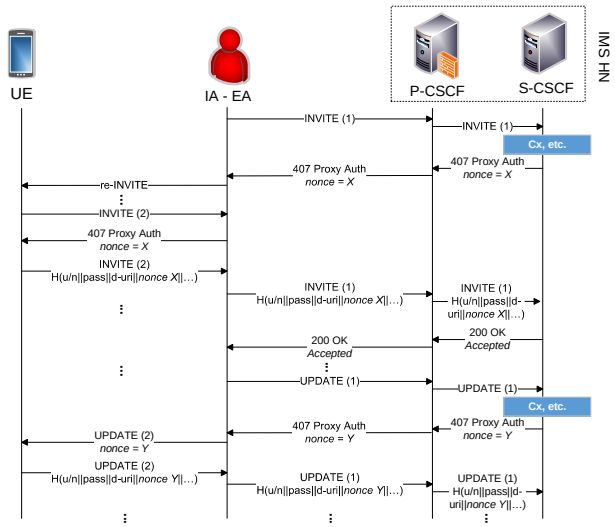


Figure 5.5: SIP Digest abuse in IMS.

the specific response is sent after the secure tunnel establishment. Therefore, the attacker will not establish such a tunnel between him and the victim's UE. Thus, the tunnel properties are mitigating this type of attacks.

During Session, the situation is similar but there are some restrictions due to the extended invitation handshake required in IMS infrastructures. When the **(a) SIP Digest** is employed, an internal or external attacker gets a challenge from the P-CSCF through an INVITE request. The attacker impersonates the P-CSCF and sends a re-INVITE request to a victim UE which starts a new INVITE negotiation to update the session. The authentication takes place between the victim and the attacker. The former captures an authentication response for his nonce value and uses it to complete his pending authentication to the real server. A spoofed re-INVITE is mandatory for the attack because the previously obtained authentication string (from the REGISTER procedure) is useless for the following reason: A valid authentication response in the HTTP Digest is computed by [20]: $H(H($username ":" realm ":" pass), nonce ":" $H($Method ":" digest-uri-value)), where H is a hash function, so the attacker cannot unbind the digest-uri from the authentication response, and consequently, he cannot use the specific response in any other SIP method than the one that was initiated (note that the attacker's spoofed re-INVITE has the same digest-uri as an INVITE message). The attack continues in the same way for all subsequent requests of the handshake (UPDATE and PRACK). The attack is depicted in Fig. 5.5. As far as the **(b) IPSec** and **(c) TLS** are concerned, the establishment of security tunnels before the initiation of the INVITE procedure provides strong protection against this type of attack.

5.3.2.4 Conference Interception

The SIP REFER method is a non-default request described in RFC 3515 [57]. Particularly, SIP REFER is used by an authorized entity (referrer) in order to request some other entity to access a resource on behalf of the "referrer". Note that the resource to be accessed, is identified by the

corresponding URI included in the SIP "Refer-To" header and can be any type of existing URIs such as SIP and HTTP. A malicious user can avail of this request by inviting himself or another UE of his choice to participate (illegally) in the session [37]. In this case the attacker spoofs a legitimate REFER request of a valid user by adding his UE's URI/public ID in the "Refer-To" or "To" header, depending on the type of conference invitation.

When **(a) SIP Digest** is employed, an attacker can act as an intermediate (MiM) between the P-CSCF and the UE, utilizing well-known attack techniques such as DNS [93, 95] and ARP poisoning [12]. It is assumed that a legitimate UE has already established a multimedia conference room (i.e. attack can be launched *during session*) and would like to invite one more user (UE3) to join. At the very first stages, a malicious user changes DNS binding in order to force the traffic passing through his domain. Consequently, whenever a legitimate UE sends a SIP REFER message, the DNS resolution procedure will force the CSCF components to forward traffic towards the attacker's domain. Afterwards the malicious user poisons the ARP correlating legitimate user's IP with his own MAC address in order to receive the responses directed to a legitimate UE.

As soon as the malicious user obtains a SIP REFER, spoofs the "To" header and the "Request-URI" value with his URI/public ID, while the remaining message is retained as is, and forwards it to the P-CSCF. Afterwards, the SIP REFER request is processed by the S-CSCF, which by its turn sends it to the destination that the "To header" points to, namely the internal attacker. The latter responds with a "202 Accepted" to the S-CSCF, while the former sends a spoofed "202 Accepted" towards the UE. Subsequently, the attacker sends a "legitimate" SIP NOTIFY message to the P-CSCF, since he is the "legitimate" referee. The internal is able to authenticate successfully the NOTIFY request as he holds a valid subscription (considering that the internal attacker is an internal user that holds a legal subscription).

After successful authentication, the P-CSCF sends a NOTIFY to the UE through the attacker who acts as MiM, while the latter spoofs the included headers that points him ("From" and "Contact") with the corresponding of UE3. The UE accepts it by sending a 200 OK response message. In the same way, the attacker spoofs and forwards it to the P-CSCF. Finally, the attacker executes an invitation handshake in order to establish a media session with the MRFP that will enable him to participate as a legitimate user in the conference room. For further information refer to Section 3.5.2. The entire attack procedure is depicted in Fig. 5.6. The green color denotes that the internal attacker is able to fulfill the specific request or generally to bypass a security mechanism. Note that an external attacker will not be able to launch such an attack because of lack of valid credentials to authenticate SIP NOTIFY message.

The security tunnel establishment provided by the employment of **(b) IPSec** or **(c) TLS** can deter the illegal modification of the REFER message and thus can prevent this attack.

5.3.3 Replay Attacks

In IMS replay attacks a malicious user initially acts passively by observing and capturing the signaling data between a legitimate user and its home network. He focuses on capturing

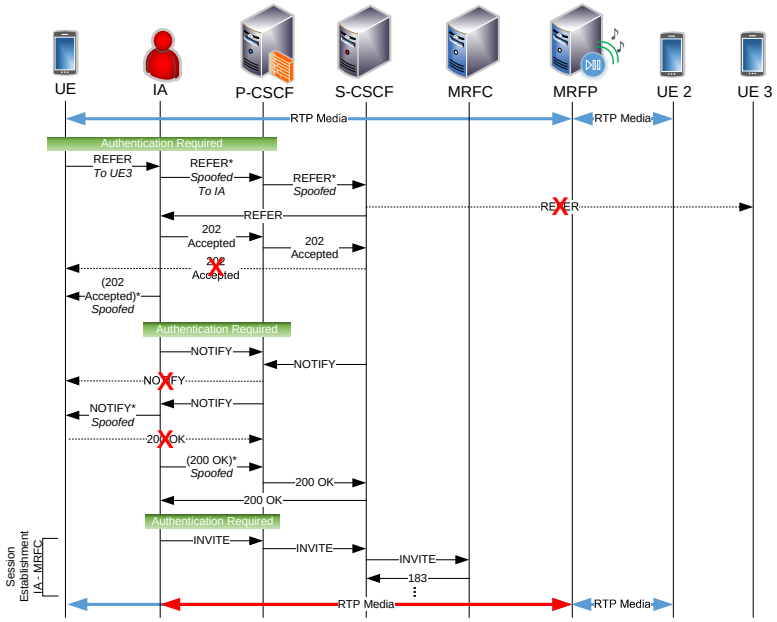


Figure 5.6: Conference interception attack in IMS.

authenticated messages to craft spoofed requests/responses that include the authentication vector of a legitimate user and thus can impersonate him and get access to his services. Replay attacks can be launched either *during registration* or *during session* in exactly the same way. In fact, the existing security mechanisms do not provide appropriate protection: the **(a) SIP digest** does not offer confidentiality, and consequently, an internal or external attacker could eavesdrop on the authentication vector and include it in a spoofed REGISTER or INVITE message. However, a nonce counter could be utilized to mitigate such attempts. For instance, a replay attack can be launched by crafting a REGISTER message that changes the nonce and the response values in the authentication header with the victim's one (used in a previously established session). Every nonce value has a limited lifetime so the malicious user must launch the replay attack before its expiration. In addition to the nonce counter, the IP-check-table can prevent these attacks. The **(b) IPSec** and **(c) TLS** protect against eavesdropping through secure tunneling. As a result, the internal or external attackers could not craft a spoofed message that would include a previously used authentication vector.

5.4 Attacks Against Availability

5.4.1 SIP Parser

A message parser is the core of any network service. It is responsible for decomposing any incoming/outgoing message into an internal structure suitable for further processing. In most cases, the parsers can only process well-formed messages. Malicious users may exploit this fact and craft a malformed message to constitute the parser unreliable or even unavailable. Protection mechanisms that offer confidentiality, authenticity, or integrity services cannot shield parsers against this type of attack.

Table 5.3: Examples of SIP Malformed Messages

INVITE NULL Via: SIP/2.0/UDP [1050::::0005::120c:354b] Max-Forwards: 70 From: <sip:nvra_private@unipi.gr>;tag=589f Call-ID: ak5fj49fhujDUuf0AjU9 CSeq: 10 INVITE Contact: <sip: [1050::::0005::120c:354b] > User-Agent: Sipp v3.2-TLS, version 20101116 Allow: ACK, BYE, CANCEL, INVITE, OPTIONS, INFO, REGISTER, NOTIFY, UPDATE, SUBSCRIBE Content-Type: application/sdp Content-Length: 146
INVITE tel: + ∗©†®†∗©∗©†††•®®£££ ???? ??? ??? Via: SIP/2.0/UDP [1050::::0005::120c:354b] Max-Forwards: 70 CSeq: 10 INVITE Contact: <sip: [1050::::0005::120c:354b] > Call-ID: ak5fj49fhujDUuf0AjU9 User-Agent: Sipp v3.2-TLS, version 20101116 From: ¿¿*©†¿*†¿©*©†©©©©©©©©©©©©©©©©©©†®†¿¿*¿¿¿¶®†*¿¿¿¿***¶¿¶®†¿¿¿¿?????? ?? ?? ???? ?? ?? ???? ?? ????????? ? ?? ?? ?? ??;tag=589f

5.4.1.1 SIP Malformed

IMS is based on the SIP which is a text-based protocol with loose message syntax "offering" to malicious users innumerable attack opportunities. Specifically, a malicious user can syntax a message that is not compliant to RFC 3261 [3] by simply modifying a well-formed one. For example, the malicious user may insert multiple or NULL headers, as depicted in Table 5.3, in order to introduce delays in routing decisions or other unexpected incidents affecting the system's availability. Furthermore, a malicious user may craft a malformed message by inserting the UTF8 (see Table 5.3) characters in header fields or by overflowing the messages with sequences of special characters. A detailed analysis of malformed messages attacks can be found in [96]. This type of attack can be launched either *during registration* or *during session*.

The employment of the **(a) SIP Digest** does not provide any kind of protection either during registration or session. Although the **(b) IPSec** and **(c) TLS** provide confidentiality, authenticity, and integrity, they protect the IMS infrastructure only from an external attacker, as an internal one could utilize the secure tunnel to send a malformed message to the corresponding server. On top of that, it should be mentioned that the lack of encryption during the first registration message enables an internal or even an external attacker to launch such an attack.

5.4.1.2 Sequence Disorder Attacks

Sequence disorder attacks could also cause loss of availability by creating a large amount of pending requests, while the server is in stateful mode. Such a behavior could burden the

system with a significant delay while the parser tries to match these requests/responses to the corresponding sessions. For instance, the attacker may have sent ACK messages before the reception of "200 OK" responses [97]. As for the security considerations, the situation is the same with the SIP malformed case that was described above.

5.4.2 SIP Flooding

Without loss of generality, the flooding attacks focus on a target's system resources (CPU, memory, and bandwidth) exhaustion to cause DoS. During such an attack, a malicious user addresses numerous requests to the provided service (target system). As the resources of the target system are limited, it is just a matter of time to cause DoS. Besides, it should be mentioned that all the services, independently from the utilized protocol, are vulnerable to resource consumption attacks since none of the existing security mechanisms (e.g. IPSec or TLS) provide resource consumption protection.

The majority of flooding attacks affecting Internet's availability have been targeted against TCP servers. In this case, a malicious user initiates numerous half-open connections to the proxy server (by sending a TCP-SYN request). The server for every incoming TCP-SYN request allocates, for a specific period of time, memory resources to handle the connection and it will release them after it gets a response from the client or as soon as the specific period of time expires. Consequently, if the malicious user forwards a spoofed TCP-SYN request, the server will never receive a response message to establish the connection and release the allocated resources (Fig. 5.7). Due to this fact, the server will retain the half-open sessions until they expire. Thus, such attacks force the server to reserver large amount of memory for a relatively long period of time, causing DoS.

Similarly, in the IMS, a malicious user can generate and target numerous spoofed SIP requests (e.g. INVITE, REGISTER, OPTIONS, etc.) to the IMS servers to cause DoS. For instance, the attacker may spoof the contact header of an INVITE request. The server (or P-CSCF in IMS) will allocate memory resources for session handling. While the contact header points to another IP, the server will not receive the SIP ACK request in order to release the memory resources. A flood of such requests may lead to DoS as described above. This is called SYN syndrome attack since it is based on the TCP's SYN vulnerability. The attacker can also launch a combined forged message and flooding attack: He can forge different messages with randomly chosen IP addresses (Fig. 5.8) enhancing the impact of the SYN syndrome attack (new memory allocation for every SIP URI/IP).

Another case of flooding attack can be launched by forwarding an enormous amount of requests to a specific network entity in order to cause large delays in active sessions or in the session establishment procedures. The target can be the P-CSCF or the MRFC/AS or even a UE. The latter can be easily flooded due to their limited CPU and memory resources and the maximum incoming traffic that they can handle. The Distributed Denial of Service (DDoS) attacks are launched against the more hardcore network entities. The attacks are originated by multiple sources which forward SIP requests with high rates, draining the memory and

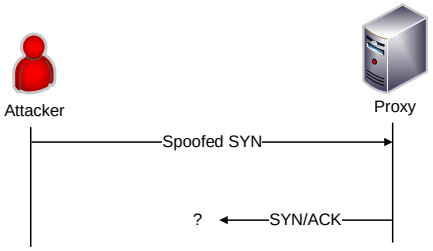


Figure 5.7: The TCP syndrome flooding attack against proxy.

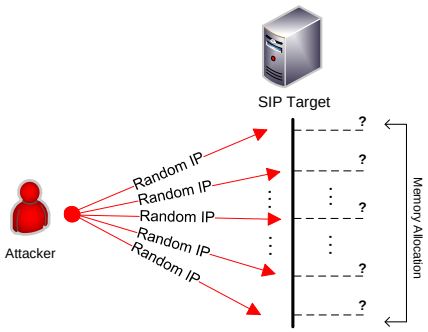


Figure 5.8: Resource allocation during spoofed SIP flooding.

CPU resources of the target system. The attackers can be innocent UEs/servers or attacking networks. In the case of innocent UEs or servers the attacker may have infected them with malicious software turning them into zombies (called slaves or zombies because they execute orders as dictated by the attacker). Then, the attacker can deploy all of them at the same time in order to flood a target machine introducing large delays in sessions.

5.4.2.1 REGISTER and 401 Flooding

The internal or external attacker can force the IMS core to execute cryptographic functions, which are considered computationally expensive to validate all these incoming requests (i.e., for every new registration message the IMS core must accomplish at least: the detection of an appropriate S-CSCF and the computation of new/fresh AVs). The specific attacks, in addition to the quick memory consumption due to numerous half-open connections, mainly target to consume the target system's processing resources. Especially in IMS environments, they can introduce delays to the network due to the heavy weight of employed security mechanisms and the large number of network entities that are involved in every registration request (i.e. opens many diameter connections over Cx, Dx with HSS, and Gq interfaces with PDF). Flooding attacks can also be launched against a specific UE through the 401 response. The attacker can easily overwhelm the low resources of the UE. The employment of the **(a) SIP Digest**, **(b) IPSec**, or **(c) TLS**, cannot deter an internal or an external attacker from launching such flooding attacks, *during registration*, since the first REGISTER message is in clear text form. Fig. 5.9 depicts the REGISTER flooding attack in IMS environments.

5.4.2.2 INVITE and BYE Flooding

The INVITE and BYE flooding attacks are considered together due to their similarities in the way they are addressed by the security mechanisms. A malicious user can generally launch flooding attacks utilizing any SIP request that can engage the IMS core network to correlate these requests to a call/session. Utilizing such requests, the attacker tries to consume memory resources of the targeted network. Another case of flooding which involves innocent entities is the INVITE reflection syndrome [27]. The attacker sends many INVITE requests to different

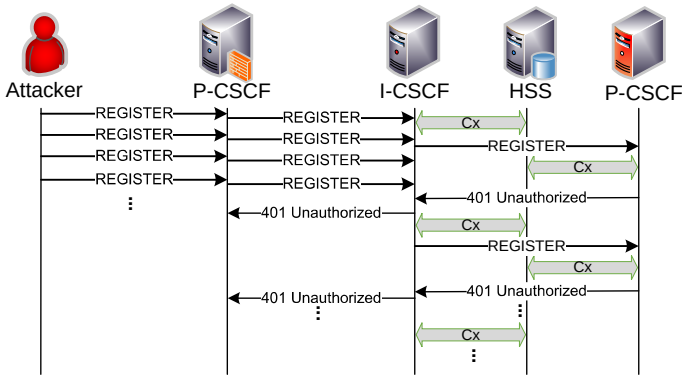


Figure 5.9: REGISTER flooding in IMS.

servers/UEs with spoofed the SIP contact header. Thus, all the involved servers will respond to the given IP of the "Contact" header, namely the target machine, causing DoS circumstances. Moreover, flooding attacks that utilize INVITEs can easily cause the DoS to a UE taking into account that many UEs are not capable of handling more than 6 requests at a time.

The specific two SIP requests are sent during a session and the employment of the **(a) SIP Digest** prevents (partially) such attacks from an external attacker due to the fact that the requests must be authenticated. However, on the one hand, in environments such as the IMS, the mandatory authentication of SIP messages can be harmful with respect to the system's availability due to the heavy overhead introduced by the mechanisms that take part in every session. On the other hand, an internal attacker can include in every request his legitimate credentials to avoid the 401/407 unauthorized responses from the P-CSCF. The establishment of the **(b) IPSec** or **(c) TLS** tunnels discourages the external attackers from launching such attacks. An internal attacker, however, can launch an INVITE or other SIP method floods by utilizing his established tunnel.

5.4.2.3 Response Flooding

The responses can be also employed for launching flooding attacks against a UE. The INVITE response flooding can be also utilized from an attacker to discover the passwords of a "Password Based Authentication" [98]. Specifically, exhaustive search can be executed by the attacker to the gathered authenticated vectors in conjunction with the nonces that the latter has chosen during the flooding procedure. Further, a reflection SIP flooding can be launched with response messages. In such a case, the attacker may forge SIP requests with the "Contact" pointing to the victim (home network entity or UE). Afterwards, he forwards them to a number of different proxies which, in turn, respond to the address derived from "Contact" header, flooding therefore the victim's device (see Fig. 5.10).

When the **(a) SIP Digest** is employed, the responses are not authenticated. To this end, internal and external attackers can both launch such flooding attacks. On the other hand, the establishment of the **(b) IPSec** or **(c) TLS** tunnels prevent response flooding attacks from external but not from internal attackers.

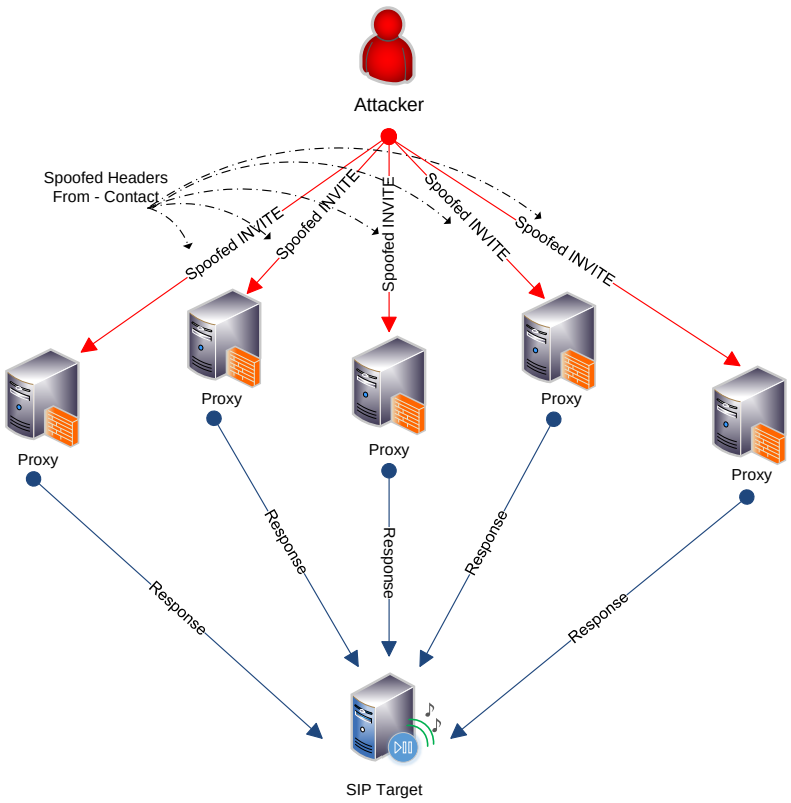


Figure 5.10: REGISTER reflection flooding in IMS.

5.5 Attacks Against Privacy/Confidentiality

5.5.1 Eavesdropping

In IMS architectures, a SIP message contains information concerning not only the parameters of a session but also the users and their devices. Besides, the text-based form of SIP messages combined with existing tapping tools (e.g., Wireshark [99]) constitutes eavesdropping a straightforward task for malicious users. For instance, a malicious user may sniff the communication between a UE and P-CSCF to gather the session's parameters for later attacks or to disclose a victim's private information such as the type of accessed services, persons/services contact, and more. Under this context, the IMS security framework specifications [18] allow the use of "*anonymous*" values in the "From" header, or the removal of the critical headers that reveal identity information (i.e. "P-asserted-Identity", "Via", "Route", "Contact", "User-agent", "Call-ID") before the message reaches an untrustworthy domain (RFCs 3323 [31] and 3325 [100]). Furthermore, an internal or external attacker may take advantage of a possible lack of confidentiality services that would reveal "private" information related to a session or to the user's identity.

5.5.1.1 Session Parameters

Disclosing the headers' values of SIP messages leads not only to privacy violation but also to security incidents such as MiM or SIP signaling attacks (see Sections 5.3.2 and 5.2.1). For

example, a malicious user could gather information about the specifications and model of a particular UE, from the agent-info header. As a result, the latter gains knowledge how this particular UE handles the session's requests and responses (e.g. can be informed about how the Call-ID is initiated by a specific UE to launch the BYE attack described in Section 5.2.1.1).

Disclosing the parameters of a SIP message can be accomplished either *during registration* or *during session*. The employment of the **(a) SIP Digest** does not provide confidentiality services, and consequently, internal and external attackers can eavesdrop on signaling parameters either during registration or during session. For example, an attacker may capture the AVs (from INVITE or REGISTER messages) to launch an MiM attack or to guess the password of the legitimate user. The employment of the **(b) IPSec** and **(c) TLS** provide the appropriate confidentiality services against both an internal and external attacker. However, an external or internal attacker can capture message communication parameters during the registration procedure, due to the lack of confidentiality services in the very first message, with the same consequences as described above.

5.5.1.2 Privacy/Identity Disclosure

As already mentioned, communication (signaling) messages, among others, carry the user's personal information that a malicious user may gather thus violating a legitimate user's privacy. Note that disclosure of the private or public identity can relate users with their "electronic" preferences (services, calls, contacts, etc.). The use of "anonymous", instead of the real user identity, remains an option even though it cannot provide adequate privacy protection since the malicious user may obtain the private ID from other header fields. Besides, as described in RFCs 3323 [31] and 3325 [100], the first hop (from the UE to P-CSCF) lacks any privacy protection mechanisms and thus the identity (private or public) of the user is exposed. It is also worth noticing that former RFCs focus on obscuring the subscriber's identities when communication messages come through untrusted domains (e.g., removing specific headers such as P-asserted-Identity). A malicious user can identify the legitimate users' "private" information either *during registration* or *during session*.

The employment of the **(a) SIP Digest**, as already mentioned in *Session Parameters* (above), does not provide any confidentiality services, and consequently, a malicious user could disclose either the public identity included in "From", "Contact", "P-Preferred-Identity" headers or the private identity located in the "Authorization" header either *during registration* or *during session*. The use of a meaningless URI value instead of the public ID during a privacy-enabled session will not block internal or external attacker's attempt to obtain the private ID from the "Authorization" header which is important for the authentication procedure of the request. When the **(b) IPSec** or **(c) TLS** is employed, the situation is exactly the same but *during the session* the provided confidentiality services prevent eavesdropping and consequently protect user's privacy.

5.6 Attacks Against Accountability

5.6.1 Toll Fraud

In a toll fraud attack, a malicious user attempts not to be charged for the use of a service. In IMS environments, an internal attacker can initiate a session without being charged for the provided service either bypassing the P-CSCF or the IMS core (by contacting a UE directly). In the IMS, the billing procedure is linked to two main identifiers: (a) the "IMS Charging Identifier" (ICID), inserted by the P-CSCF in the "P-Charging-Vector" header, which is responsible for correlating billing information when communication messages pass through various networks elements, and (b) the "Inter Operator Identifier" (IOI) that denotes the participating networks in a specific transaction. These two identifiers are important for the Charging Data Function (CDF) to start the billing procedure. Consequently, a malicious user may try to modify or remove the aforementioned identifiers to avoid being charged.

5.6.1.1 Bypass P-CSCF

A malicious user could bypass the P-CSCF and consequently the inclusion of the ICID in the "P-Charging-Vector" header by sending the request (e.g., INVITE) directly to the S-CSCF. Due to the lack of ICID, the billing service would consider the session a toll-free session.

A malicious user can launch such an attack only *during a session* since providers charge only the established sessions. The employment of the **(a) SIP Digest**, **(b) IPSec**, or **(c) TLS** does not provide adequate protection against internal attackers while they send messages directly to the S-CSCF. However, the IMS specifications advises the restriction of access to I-CSCF and S-CSCF to everyone but core IMS entities.

5.6.1.2 Bypass IMS

A similar attack is described in [101], where the internal attacker determines the IP address of the callee from the "180 ringing" response. Afterward, the malicious user cancels the current session (by issuing a CANCEL request) and sends a new session request (by issuing an INVITE request) toward the callee bypassing the IMS. Equivalent to the "Bypass P-CSCF" case, such an attack could be launched only *during session*. As far as the protection mechanisms (**SIP Digest**, **IPSec**, and **TLS**) are concerned, it should be stressed that none of them provides protection against such attacks because every entity in the IMS that initiates a session will get back the 180 ringing response and consequently the callee's contact address.

Chapter 6

IMS Protection Schemes in Literature

6.1 Introduction

Beyond and above the security mechanisms proposed in IMS specifications, various researchers have been working on the improvement of IMS security. Most of the effort was on flooding detection methods while other security concerns, such as private data disclosure and anonymity, didn't attract much attention. The majority of the proposed IMS protection frameworks are presented in this chapter. Their evaluation is based on the security requirements that they intend to protect in a way similarly to Chapter 5: (a) defending the Integrity and Authenticity of messages and data, (b) preserving the Availability of the services and user devices and (c) protecting the Confidentiality of data and users' Privacy. Table 6.1 outlines the aforementioned security mechanisms that have been proposed in the literature, highlighting their contribution to the protection of the IMS architecture. A more thorough and detailed comparison of their features and drawbacks is provided in Chapter 7.

6.2 Frameworks Protecting Integrity and Authenticity

The SIP spoofing attacks mainly threaten Integrity and Authentication requirements. Spoofing includes impersonation, MiM, replay and tampering attacks. Towards SIP tampering detection, the authors of [102] propose an SQL injection detection framework that utilizes signatures for detecting possible attempts, by the incoming requests, to modify in an unauthorized way the user's database. A set of keywords such as the SQL specific commands INSERT, UPDATE, or UNION, which must not be contained in the SIP message header has been defined, as well as additional restrictions such as "Authorization" header length "limitation", to shield SIP message against injection attacks.

In [103], the authors propose a mechanism for integrity protection in SIP messages. Specifically, the contents of the SIP messages are hashed, including the header and the body of the message, with the user's password. Also, this model proposes the embodiment of an extra header (Verify-Body) that contains all the necessary information (e.g. username, hash function, realm, etc.) for

the server to verify the integrity of the message. This mechanism can deter signaling attacks, replay, and MiM attempts in the SIP layer, but it cannot prevent flooding and eavesdropping attacks.

In [24], a cross protocol mechanism for detecting SIP signaling attacks in VoIP environments is described. The authors introduce an IDS that statefully correlates the RTP and SIP traffic. The idea is based on the fact that a faked SIP request comes with an orphan RTP flow from the callee. For instance, a BYE attack can force the caller to release the media stream while the other calling member, without being notified about this event, will keep forwarding the RTP stream. Such orphan RTP streams enable the specific IDS to detect signaling attacks. On the other side, the CANCEL and UPDATE attacks cannot be identified by this method because they both occur before an RTP session.

In [104], the transaction dialog IDs of every message are extracted and compared with the corresponding IDs of the incoming messages. Messages that cannot be matched and correlated with any of the previous IDs are dropped. The main drawback of this method is that it cannot prevent a direct CANCEL attack.

A mechanism that monitors REGISTER messages is described in [105]. A successful registration takes place when a 200OK is received by the server which includes the REGISTER value in the CSEQ header. Then a tuple is created in a table containing information gathered from the SIP message: the user's identity (UID), the UE's IP address, a timestamp and the duration of the specific registration in seconds as it is derived from the "expires" header. The UID acts as primary key in the table. This table is actually a white list and therefore only the stored UIDs will be processed. This mechanism does not offer any protection against IAs which have a legitimate subscription and can authenticate all their requests. Moreover it is unable to deter INVITE flooding attacks and also it cannot be deployed in IMS infrastructures in conjunction with security tunnels because the messages do not contain SIP authentication except for the first REGISTER request. Finally, messages with forged UIDs can bypass the mechanism and also the authors do not provide any description for handling the DDoS attacks.

Another approach for detecting spoofed calls is presented in [106]. The authors propose a method for profiling networks by extracting characteristics from audio streams. The extracted data are a combination of noise characteristics and packet loss which are adequate to build call/provider profiles. Their proposal is based on the observation that while the audio stream traverses through different networks, it is re-encoded by each network using different codes and bit rates (different network technologies use different audio codec specifications). As the audio is re-encoded, its quality is degraded providing enough information for the mechanism to make fingerprints from these artifacts. This approach can efficiently detect the actual call origin but only when the audio stream is included. Thus, all signaling attacks cannot be deterred. Moreover, it cannot detect spoofing attacks when the attacker comes from the same physical location and, finally, it does not address flooding attacks.

6.3 Frameworks Protecting Availability

The detection of resource consumption attacks is considered in [25]. A DoS mechanism that extends the detection to the transport layer is proposed. Specifically, the mechanism detects traffic abnormalities by calculating the *Hellinger distance* between requests and responses, taking into account not only the SIP traffic but also the transport protocol's traffic. This scheme does not provide a methodology to detect and deter the attackers but is only limited on triggering alarms when a flooding attack is under way. Moreover, it is focused on the INVITE request flooding attack case and cannot sufficiently detect INVITE reflection DDoS attacks where the distance between responses/requests remains unchanged.

In [26] a flooding detection model is introduced which is based on priority queues. More specifically, it deploys two queues, one of high and one of low priority. All the INVITE messages are inserted in the low priority queue while the responses are inserted in the high priority one. The messages in low-priority queue will be processed only when the high priority queue is empty. The result is that the legitimate requests are the ones handled first, while their responses are in higher priority. On the other hand the INVITE messages are processed with an increased delay but the server can still be on-line, avoiding DoS consequences. Moreover, the illegitimate INVITEs are discarded faster since they do not come with responses so the high priority queue remains empty. Nevertheless, this model does not actively deter or block the attacker but only mitigate the effects of the flooding attack. Furthermore, the attacker can bypass the mechanism by flooding the server and consequently the two queues, both with INVITE requests and responses (e.g. 100, 200, 180, etc.).

Another flooding attack prevention mechanism is presented in [107]. The authors propose a detection mechanism that is able to detect DoS including some SIP signaling attacks. Specifically, Honey Pot architecture is deployed in order to provoke attacker's interest and thus gathering useful data towards the detection of such attacks. Utilizing anomaly and signature based detection techniques; the mechanism creates profiles of "normal behavior" for users and network entities and signatures of known attacks. Any deviation from the normal behavior standards can be considered as an attack. An attack is detected by correlating different events through specific rules. For instance, the BYE signaling attack can be detected by spotting orphan RTP flows after a period of time (only from one participant while the other has received the BYE message) utilizing signature-based correlation with attack patterns.

In [108], a framework towards the detection of malformed messages is proposed. Every incoming message is checked for its conformity with the RFC 3261 [3] SIP syntax through predefined rules based on signatures describing the well-formed messages. Furthermore, in [109], a self-learning system that can detect not only known malformed messages but also new ones is introduced. The specific mechanism extracts a model of normality observing the SIP traffic to detect anomalous content in messages.

6.4 Frameworks Protecting Privacy

The research work related to users' privacy is considered limited. Particularly for identity protection, the SIP protocol's specification [3] describes the provision for an anonymous identity by introducing a meaningless URI, such as "*sip:abc@anonymous.invalid*", which can replace users' identity only in the "From" header field. However, when this feature is used during a signaling session, authentication mechanisms cannot be utilized since mechanisms like IMS AKA with IPSec, TLS with SIP Digest and SIP Digest (see Chapter 4) require the user's public and private identities. Finally, it cannot conceal caller's identity which is included in the "To" header field while the proxy must be aware of the other contacting point.

A better privacy level for SIP-based sessions is achieved through the service described in [31]. This solution facilitates the users to conceal their identity inside a domain. A new header is employed for this reason which can have the following values:

- "header": Indicates the intension of the user to obscure his private information. The home network is responsible for concealing the corresponding headers.
- "session": This value is used by the UE when a user wishes to conceal private information included in the SDP.
- "user": It is used in registration messages and indicates that the privacy services will be provided by the network when the UE is incapable of this.
- "none": Indicates that none of the privacy services will be applied to the message.
- "critical": It denotes the criticality of asserting the privacy functions. The request is rejected when the network cannot provide the required service.

However, this mechanism is vulnerable to bid down attacks where the attacker can strip out the private headers and thus the messages will be processed as normal ones without the desired privacy protection. Moreover, it does not protect user's identity at the first hop and it cannot hide usernames (correspond to private user identity) included in the "authorization" header and required for user authentication.

A small improvement to [31] is introduced in RFC 3325 [100]. Specifically, the user who requires protection of his identity sets the "id" value to the "Privacy" header. Afterwards, the user sends a SIP request with a meaningless URI instead of his real identity. When the server receives the request, requires authentication form the user in order to determine his set of public/private identities. The user responds again with the meaningless URI but includes his username in the "Authorization" header. After the successful authentication of the user, the server gathers specific user's public identity from the database. Afterwards, the server inserts the user's public identity in the "P-Asserted-Identity" header and forwards it to a trusted domain. This scheme extends the functionality of RCF 3323 [31] while it supports authentication and privacy services at the same time. However, the authorization header can disclose the user's private identity to eavesdroppers. Moreover, as it is implicitly stated in RFC 3325, it provides only a

partial solution to the problem and requires the existence of at least one trusted proxy while it proposes the use of TLS or IPSec mechanism.

A simple mechanism is presented in [35] where the received messages are anonymized by a server. The latter keeps the associations between real and anonymous identities. This scheme cannot be used in conjunction with an authentication mechanism and also the identities are revealed during the handshake at the first hop. Additionally, the callers' identity cannot be concealed.

[33,34] propose a framework focused on using encrypted header values, instead of the real users' identities. The specific scheme utilizes public key cryptography. Even though it can prevent the disclosure of identities and preserve users' anonymity, it introduces significant overhead as a result of the PKI infrastructure. This approach also raises concerns related to the storage and management of the certificates similar to the S/MIME. In the same work symmetric cryptography (Advanced Encryption Standard (AES) [110] is also applied for protecting users' identities. Even though the use of symmetric cryptography offers significant performance gains, it is necessary to modify the 4-way SIP handshake in order for the communicating parties to share the secret key.

The utilization of Mist architecture [111] is proposed in [112]. Specifically, the SIP messages are routed through Mist's lighthouses. A PKI infrastructure is required to encrypt the SIP messages which afterwards are encapsulated in Mist packets. The caller sends the request in a Mist packet to the lighthouse. The latter determines the callee's destination and sends a new Mist packet to the corresponding lighthouse. Callee's lighthouse creates a new Mist packet that includes the INVITE request and sends it directly to the callee. The callee receives the Mist packet and extracts the encapsulated SIP INVITE request. Afterwards, inserts his location in a SIP response and encrypts it with caller's public identity. Next, he encapsulates the encrypted response message in a Mist packet and sends it back to the caller through the lighthouses (the Mist's encapsulation procedures also take place as before).

When the caller receives the Mist packet that include the response with caller's location, initiates a session establishment handshake in order to communicate with the callee. This scheme does not only require the employment of many Mist lighthouse, something that introduces significant amount of delay, but also requires the modification of SIP's transactions. Moreover, it requires the deployment of a PKI infrastructure for encrypting the SIP messages. Finally the session establishment procedure, which takes place after the Mist message exchange, remains unprotected and therefore a passive eavesdropper may disclose users' identities.

Table 6.1: Security Protection Frameworks in Literature (From The Attacker's Perspective)

		Proposed Mechanisms in Literature																			
Threat Category	**Attack**	**[25]**	**[26]**	**[106]**	[107]	**[102]**	**[24]**	**[104]**	**[108]**	**[109]**	**[105]**	**[103]**	**[33]**	**[34]**	**[3]**	**[100]**	**[31]**	**[112]**	**[32]**	[35]
SIP Tampering	SQL Injection																			
SIP Signaling	BYE																			
	CANCEL																			
	Re-INVITE,UPDATE																			
Impersonation	User Impersonation (IP and ID spoof)																			
MiM	Generic MiM - Authentication Abuse																			
	Conference Interception																			
SIP Parser	SIP Malformed																			
Flooding	REGISTER, 401 Response																			
	INVITE, INVITE Reflection, BYE																			
	CANCEL																			
	Response, Response Reflection																			
Eavesdropping	Session Parameters Disclosure																			
	Privacy/Identity Disclosure																			
Toll Fraud	Bypass IMS / CANCEL																			

The vulnerabilities which are covered by mechanisms in literature are in red color. The green color denotes that the attacker is not deterred by the specific mechanism and the system is susceptible to the corresponding vulnerability. The yellow color indicates that the vulnerability is partially covered.

The S/MIME [32] can also be used as an alternative solution for protecting SIP's signaling messages. Using S/MIME, the SIP headers are sent encrypted by utilizing digital certificates when confidentiality is desired by the caller. Every message consists of two parts. The first part contains the SIP headers that are in clear text for message routing and handling. The second part bears the encrypted headers and the media that correspond to the actual request. In cases where message integrity should be also protected, a signature of the message is also included. The server by decrypting the encrypted part obtains the actual SIP header. Utilizing this mechanism, a user can conceal sensitive information such as his identity and the type of the UE ("user-agent" header) since the correct values are included only in the encrypted part. However, the digital certificates reveal information about the identity of the caller since they bear related information important for the encryption/decryption procedure. Moreover, the certificates must be signed by a trusted certificate authority and thus the UE should support different root certificates, because different users trust different root authorities. In cases of subscribers which have multiple UEs, the same certificate must be installed, updated and managed in all user's devices independently [19].

Chapter 7

Proposed Security Mechanisms for IMS/VoIP Environments

7.1 Introduction

The numerous vulnerabilities that can be exploited in IMS environments have been considered by the 3GPP which responded by including a thorough security policy in its specification. The employment of well-known internet security mechanisms (IMS AKA with IPSec, TLS, SIP digest) as described in Chapter 6 has definitely raised the protection level up to a point. However, numerous unprotected malevolent actions against the provided services and users are totaly deterred since either they do not consider the case where an internal user can act maliciously or they do not focus on SIP signaling's specific characteristics.

As derived from Table 5.1 provided in Chapter 5, none of the existing security mechanisms can shield IMS against DoS or adequately deter signaling, spoofing and MiM attacks. Moreover, the inability to protect initial registration messages and secure identity information, can lead not only to privacy violation but it also offers opportunities for additional attacks.

On top of that, other scientific works either require the employment of an additional infrastructure, such as a PKI, or utilize strong and resource demanding encryption algorithms. Thus, these solutions will inevitably introduce a significant overhead to an already burdened architecture and also cannot be utilized by low end user equipment. Besides, there is no effective protection against DoS attacks, since existing techniques stay only on detection. Regarding securing identity information, even RFCs 3323 and 3325 do not adequately preserve the user's anonymity against passive eavesdroppers.

Taking into account the aforementioned facts, it is clear the need for a security framework that can mitigate and deter these issues. Concerning acts that threaten architecture's availability, a Bloom filter based framework has been developed against flooding attacks. The specific mechanism focuses on signaling protocol's characteristics and by utilizing an application layer metric, can avoid resource demanding calculations while it is able to efficiently detect DoS and DDoS against servers and UEs.

Additionally, a cross layer intrusion detection and prevention architecture against a wide range of attacks that threaten system's authenticity and availability is introduced. The mechanism utilizes two modules: the first is responsible for forged/spoofed messages while the second for flooding attacks detection and prevention. The information processed, is gathered from layers 2, 3 and 5 of the protocol stack and correlated according to their current SIP authentication state. A statistical model is also utilized in conjunction with a Bloom filter for detecting deviations from normal behavior.

Finally, the proposed security architecture protects user's Privacy by introducing a scheme that provides the user with an one-time identity, even from the first step of communication. This scheme does not require the deployment of additional infrastructures or modification of the SIP transactions. Besides, it provides anonymity services both to callers and callees.

7.2 Protection Against Threats Violating Availability

7.2.1 General Description

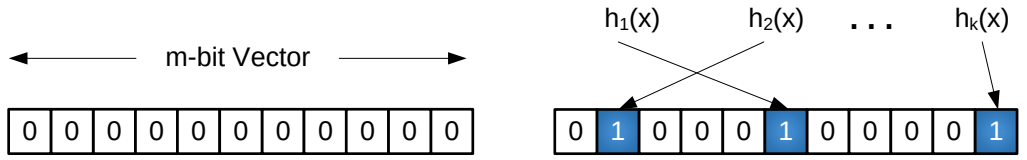
To address flooding attacks (see Section 5.4.2), a mechanism based on the Bloom filters concept [1] is proposed. The mechanism has been developed to monitor and detect flooding attacks against SIP based-proxies (i.e. P-CSCF) and end users (i.e. UE).

The Bloom filter is a data structure used to represent a data set in a way that facilitates someone to efficiently determine the existence of an element in that set. The disadvantage of such a structure is the small percentage of false alarms, since it is based on probabilistic functions such as the hash functions. However, in such an environment the impact is almost negligible. For instance, if a call is incorrectly identified as malicious, the caller will immediately understand it by receiving the appropriate response (e.g. busy signal) from the server and he would try to make a new one. In contrast, the impact of a faulty assessment of a legitimate e-mail as spam will be greater, since the sender might not ever find out if this message was delivered or not

The Bloom filter was introduced in 1970 with main objective to minimize the delay in data searching. The limited amount of space required to represent a data set by the utilization of Bloom filters offered the opportunity of migrating that data from storage devices to the much faster system's main memory.

7.2.1.1 Standard Bloom Filters

Suppose that a set $S = \{a_1, a_2, \ldots, a_n\}$, of n elements has to be represented by a Bloom filter. The filter comprised of a vector V of m bits. All bits of vector V are initially set to zero. For every element of the set S, λ hash functions $h_1, \ldots, h_\lambda$ in $\{1, \ldots, m\}$ are applied. For all $x \in S$, the $h_i(x)$ where $1 \leq i \leq \lambda$, points to a specific bit in the filter. That bit is set to 1. Every bit can only be changed once, namely from 0 to 1. If a $h_i(x)$ points to a bit that has already been set to 1, that bit remains unchanged.



(a) After its initiation. **(b)** During the insertion of an element x.

Figure 7.1: Bloom filter states.

An element y exist in set S if every $h_i(y)$ filter's bit is equal to 1. There is also a probability this claim to be wrong: There is probability P, where even if all the $h_i(y)$ are equal to 1 but the $y \notin S$. This is called false positive. Since that P is almost negligible, the Bloom filter's utilization is acceptable, depending on the environment where the filter is employed. The probability of a specific bit V_i to be equal to 0 is:

$$P = (1 - 1/m)^{\lambda n}$$

Thus, the probability of a specific bit to be equal to 1 is [113]:

$$P = Pr\{V_i = 1\} = 1 - (1 - 1/m)^{\lambda n}$$

The claim that an element y exists in the set S can be a false positive. In such a case the specific $V_1, \ldots, V_\lambda$ bits of the vector are all equal to 1:

$$V_{h_1(y)} = V_{h_2(y)} = \cdots = V_{h_\lambda(y)} = 1$$

The probability for this to happen is:

$$P = Pr\{V_{h_1(y)} = 1 \; and \; V_{h_2(y)} = 1 \; and \; \ldots \; and \; V_{h_\lambda(y)} = 1\}$$

Thus,

$$P = (1 - (1 - 1/m)^{\lambda n})^{\lambda} \tag{7.1}$$

The optimal number of the employed hash functions with respect to the false positive rate can be derived through the differentiation the Function 7.1 [114]. Since $(1 - 1/m)^{\lambda n} \approx e^{-\lambda n/m}$, the derivative of (7.1) is:

$$\ln\left(1 - e^{-\frac{\lambda n}{m}}\right) + \frac{\lambda n}{m} \frac{e^{-\frac{\lambda n}{m}}}{1 - e^{-\frac{\lambda n}{m}}} \tag{7.2}$$

The derivation of 7.2 equals to 0 when $\lambda = \ln 2(\frac{m}{n})$, which is also the global minimum. A Bloom filter of m bit size is depicted in Fig. 7.1a. Every bit is set to 0 during the filter's initiation. In Fig. 7.1b, the colored blocks bits represent an element x that entered the Bloom filter by changing the value from 0 to 1.

Considering Fig. 7.2, the element y does not exist in the filter since the output of the $h_1(y)$ function point to a zero bit on the vector. On the other hand, the element z exists in the set, with a probability P since it can be a false positive.

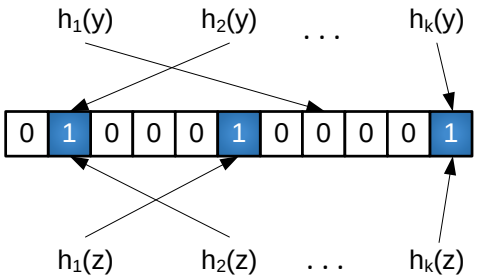


Figure 7.2: Checking the existence of elements y, z in the Bloom filter. Element y does not exist while z exists.

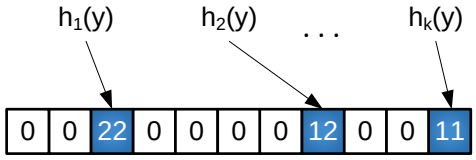


Figure 7.3: Counting Bloom filter.

7.2.1.2 Counting Bloom Filters

A more advanced Bloom structure is the counting filter. The conventional filter does not allow any other operations (e.g. subtraction or addition) than simply turning the zero bits to 1. Consequently, a set that dynamically changes over time, cannot be represented by the conventional Bloom filter.

The counting Bloom filters are capable of counting how many times a hash output points on specific table position by utilizing counters instead of single bits. That filter in depicted in Fig. 7.3.

7.2.2 Monitoring System

The proposed detection mechanism utilizes Bloom filters with counters. Specifically, in order to monitor, and consequently identify, flooding attacks against SIP proxies and end users, a two-part Bloom based monitor has been employed. Actually, the monitor's main task is to record the state (open, in progress, established) of any incoming session, information that will be utilized during the detection phase. Furthermore, it should be stressed that a single tracking memory solution can only be used as an indication of an attack, whereas the proposed monitor's information can be used, among others, for identifying malicious messages. On top of that the Bloom filter is considered as a cost effective and efficient method for recording/logging large amount of information in "compact" data stores.

7.2.2.1 Part I: Session Establishment Monitoring System

The first part monitors all new incoming requests, while the second one monitors the requests that have been directed to a specific end user. The first part of the monitoring system features three distinct Bloom filters in order to keep track of: (a) the INVITE requests, (b) the

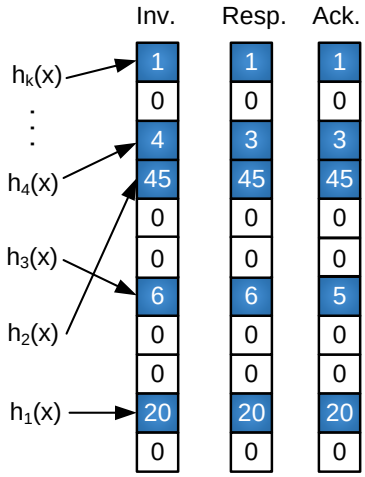


Figure 7.4: Snapshot of the session establishment monitoring system.

corresponding responses and (c) the final ACKs. The key input to the hash functions is the concatenation of the data included in the "Call-ID" and the "From" header (these headers can identify a session uniquely). Fig. 7.4 illustrates the proposed monitor whereas the Algorithm 7.1 presents the proposed logging procedure only as far as the first part is concerned. It is important to highlight that the employment of the three Bloom filters allows the detection of flooding attacks using either SIP responses or final ACKs messages.

Algorithm 7.1: Monitoring Algorithm

```
1  for each incoming message do
2      if type is request then
3          Check the method
4          if method is INVITE then
5              Update the invite Bloom filter(increase by one the appropriate entries of the
               filter)
6          end
7          else if the method is ACK then
8              Update the ack Bloom filter(increase by one the appropriate entries of the filter)
9          end
10     end
11     if type is final response then
12         Update the response Bloom filter(increase by one the appropriate entries of the
           filter)
13     end
14 end
```

7.2.2.2 Part II: End User Traffic Monitoring System

The second part of the monitor keeps track of the sessions directed to a specific end user. The main differences with the first part of the monitor are:

1. The key input to the hash functions is the "To" header of the incoming request since only

the traffic to specific entities is considered.

2. Only the "INVITE" segment (see Fig. 7.4) is employed, since the other segments are not required.

The monitoring algorithm of this part of the monitor is that for any new INVITE message the corresponding entries of the monitor are increased by one. Table 7.1 illustrates the differences between the two parts of the monitoring system.

Table 7.1: Characteristics of the monitoring system.

Segment	Input Headers	# Bloom Filter	Bloom Filter
Part I	"From", "Call-ID"	3	Invite, Response, Ack
Part II	"To"	1	Invite

7.2.3 Detection Method

In general, a flooding attack is associated with numerous incomplete sessions. On the contrary, in three way handshake protocols, like SIP, for any valid session there is a unique one-to-one mapping among INVITEs - responses - ACKs. Therefore every INVITE message should be matched with one and only one response and acknowledgement.

The proposed detection mechanism also takes into account (a) the average network delay (Nd) and (b) the average user response time (URT). Recalling that each INVITE message should be matched with the corresponding response and ACK, with a one-to-one mapping, the term **session distance** is defined through the following metric:

$$dist = Num\ of\ INVITEs - 0,5 * (Num\ of\ OK + Num\ of\ ACK) \tag{7.3}$$

Clearly, in a well "behaved" environment the *session distance* metric for any established session is equal to zero. Based on the proposed mechanism, the existence of a distributed flooding attack can be identified by calculating, at specific intervals, the *session distance* metric for all the entries of the filter. The frequency of the checks, corresponding to a time interval T_{ddos}, depends on the SIP based system capabilities and resources.

The threshold for DDoS can be specified by "observing" the average value of the *session distance* metric during a specific period of system's operation (described by Algorithm 7.2). This can be considered as the "training period" of the DDoS module. For instance, consider a case

Algorithm 7.2: DDoS threshold estimation procedure

```
for all elements in the monitor do
    Session_distance_i = #invite_i − 0.5 ∗ (#resp_i + #ack_i)
    Threshold_value = +session_distance_i
end
```

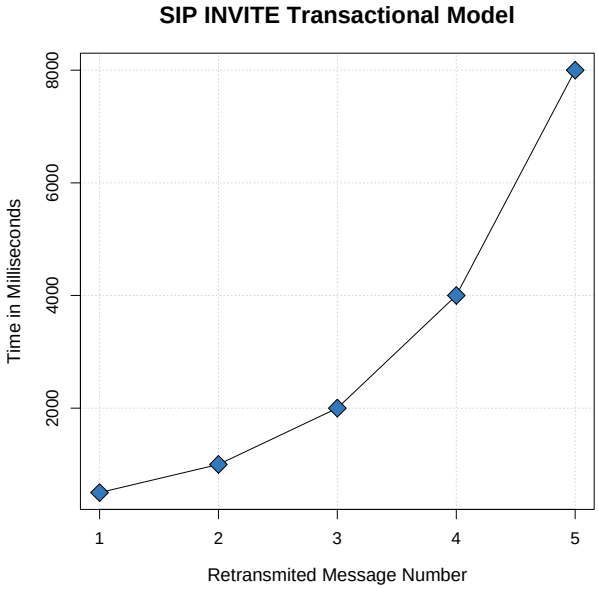


Figure 7.5: SIP - INVITE transactional model.

where the administrator of a particular realm observes the SIP traffic during peak hour and by utilizing Function 7.4 finds out that the average value of the *session distance* is T_{sd1}.

$$T_{sd1} = \left(\sum_{i=1}^{n} threshold\_value_i \right) \Big/ n \tag{7.4}$$

where n, is the number of different executions of the threshold estimation procedure.

Likewise, the network delay (Nd_1) and user response time (URT_1) parameters are calculated. The appropriate threshold value that should be adopted for identifying a DDoS attack is calculated through the formula:

$$T_{alarm} = T_{sd1} + Nd_1 + URT_1 + \delta \tag{7.5}$$

where δ is a parameter reflecting the specific SIP system's capabilities and resources. A DDoS alarm is triggered if the observed traffic exceeds T_{alarm}.

The first part of the monitor can be also utilized for detecting single-source flooding attacks and particularly cases where the attacker uses exactly the same message to launch an attack against a proxy or a specific end-user. In this case, a specific entry of the INVITE segment of the first part of the monitor keeps increasing, causing the increase of the corresponding session distance values; even though they do not exceed the T_{alarm} value. A second threshold value, $T_{single1}$, is therefore necessary in order to detect flooding attacks that utilize the same message against a specific end-user. Consequently, if a computed $session_distance_i$ value exceeds the threshold $T_{single1}$, an alert is triggered and the message is discarded.

However, if the attacker utilizes different SIP messages in order to launch flooding attack against a specific end-user, he may evade the detection mechanism since the incoming INVITEs will be distributed uniformly in the corresponding part of the monitor and thus the calculated

session $distance_i$ will remain under the threshold $T_{single1}$. Nevertheless, the requested URI and the corresponding "To" header will be the same as the attacker targets against a specific end user. Thus, in this case, the utilization of the second part of the monitor is necessary. Each entry of this part of the monitor corresponds to the incoming request targeting to a specific user. If any of these entries exceeds the threshold $T_{single2}$, an alarm is triggered. This part of the monitor is checked every T1 sec, where T1 is defined in RFC 3261 [3] as the retransmission time of the INVITE request until it receives a response. The default value for T1 is 0.5 seconds while the retransmission rate of a message is calculated by the function $T1 = 2 * T2$ (Fig. 7.5). Since the maximum number of retransmission is eight (8), the duration of that procedure cannot exceed the 32 seconds. Thus, the threshold $T_{single2}$, should not exceed the number of 8 retransmissions during 32 seconds.

The proposed detection mechanism can trigger specific administrative tasks, according to the security policy that has been employed, for mitigating the attack. For example, a service provider "PDR-A", may employ a policy according to which when an incoming connection is recorded by the proposed monitor and the corresponding threshold is exceeded, it means that this specific session has already been "recorded" (by the mechanism) and thus it should be dropped. A different service provider "PDR-B", may employ a different security policy according to which as long as the computed session distance is above the corresponding threshold value, all incoming messages are dropped.

7.2.4 Detection Example

Fig. 7.6 presents a snapshot of the mechanism and how the thresholds are being checked checked during sessions. Specifically, the Function 7.3 is applied on the tables of Part I of the monitoring system. A DDoS attack can be detected by comparing the sum of the counters included in the DDoS table with the T_{alarm} threshold. Since no alarm is triggered after the first comparison, every single record of DDoS table is checked. If a specific record exceeds the $T_{single1}$ threshold, a flooding attack that employs same message is under way. Simultaneously, the records of the end-user monitoring system (Part II of the monitoring system) are compared with the $T_{single2}$ value. When a record exceeds $T_{single2}$, it is concluded that a flooding attack against a user is under development.

7.2.5 Evaluation

The evaluation of the flooding detection mechanism has been done in terms of the overheads introduced (end-to-end delay, resources consumption) and the effectiveness of the scheme. For the evaluation, an experimental test-bed architecture has been utilized. Specifically, it consists of the following components (Fig. 7.7):

- IMS domain: The OpenIMS platform has been utilized.
- Two legal call generators Alice and Bob.

- A malicious request generator Eve: A Proprietary SIP flooding attack generator has been utilized.

For the scope of the evaluation, different scenarios have been implemented as presented in Table 7.2. The attacker floods the server with 3 million messages through S4 and S5 while in S2 and S3 he forwards 1.1 million messages.

As far as the measurement of the detection time and of the overheads introduced, Fig 7.8 to 7.12 depict the results for scenarios S1 to S5 correspondingly, while Table 7.3 presents the statistical characteristics for each of them with 95% Confidence Interval (CI). The average detection time is under 40 μs in all different scenarios.

Table 7.4 and Fig. 7.14, illustrate the end-to-end delay introduced by the mechanism. Moreover, Fig. 7.14 depicts the necessity of such a detection mechanism since the attacker "introduces" a significant amount of delay to the end user communication in flooding attacks with high message flow rate.

Additionally, the CPU load has been measured for all the scenarios (S1 to S5) with and without the proposed scheme, in order to identify the CPU load-overhead introduced by the proposed scheme. The results are presented in Fig. 7.13. Specifically, when the system is not under attack (S1) the CPU load is the same for both configurations. On the other hand, when the system is under attack (S2-S5) the CPU load increases approximately 2-3 times as compared to a system that does not implement the proposed solution. This increase was expected since:

(a) in the case of an attack the system process more traffic including "costly" operations like hashing, file opening and writing; note that these operations are introduced by the proposed scheme and

(b) for the purposes of the tests, in scenarios S2 to S5 whenever an attack has been identified the system was re-initialized (to continue the evaluation procedure), instead of simply generating an alert.

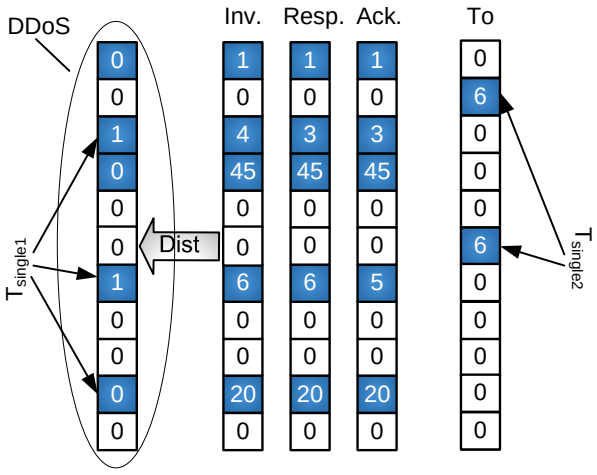


Figure 7.6: Threshold value representation.

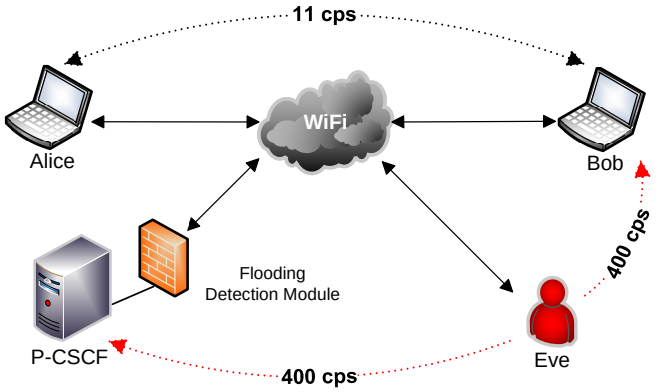


Figure 7.7: Test-bed network architecture.

Table 7.2: The Scenarios Employed for Evaluating the Proposed Flooding Detection Mechanism

Scenario Name	Scenario Description
Scenario 1 (S1)	In this scenario there is only legal traffic. Specifically the "*legal request generator*" generates requests (at a pace of 11 req/second), while the "*legal response generator*" generates the corresponding responses.
Scenario 2 (S2)	In this scenario the malicious user generates requests (at a pace of 150 req/second) that are addressed to a(n) (specific) innocent user who tries to respond to all of them with the existence of background traffic of 11/calls per second (cps).
Scenario 3 (S3)	In this scenario the malicious user generates requests (at a pace of 1 req/10 microseconds) that are addressed through the proxy to various clients belonging to non-existing domains with the existence of background traffic of 11/cps.
Scenario4 (S4)	In this scenario the malicious user applies more effort to his attack increasing the generated requests at a level of 400 req/second. Apart from that, the other properties of the attack remain the same as in S2.
Scenario 5 (S5)	In this scenario the malicious user applies more effort to his attack increasing the generated requests at a level of 400 req/second. Apart from that, the other properties of the attack remain the same as in S3.

Table 7.3: Statistical attributes (in µs) with 95% CI.

Scenario	Max.	Min.	Avg.
S1	29.13	29	29.06
S2	33.99	33.5	33.74
S3	29.03	28.9	28.96
S4	32.7	32.4	32.6
S5	30	29.8	29.9

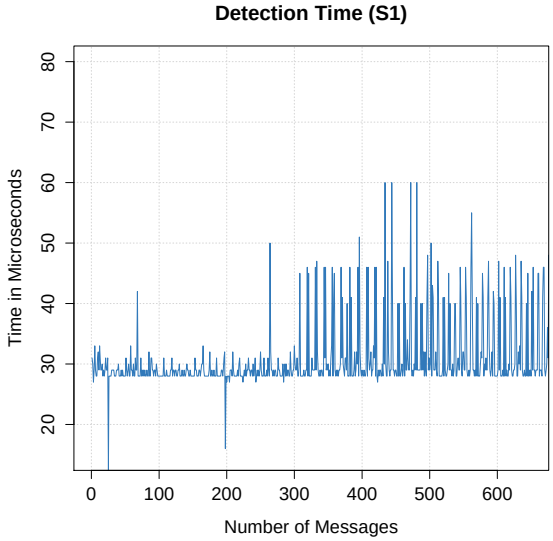


Figure 7.8: Detection time in S1.

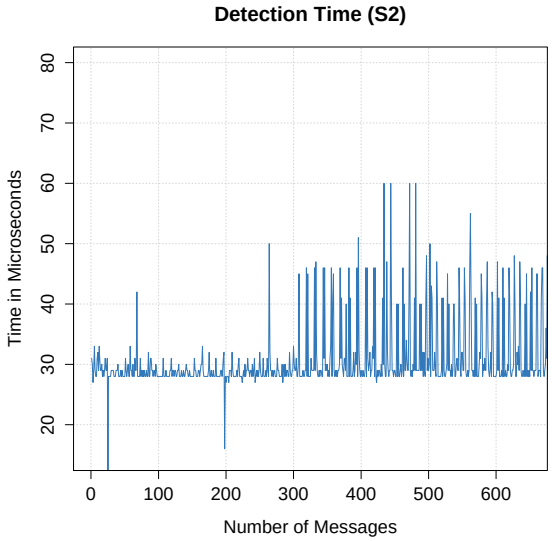


Figure 7.9: Detection time in S2.

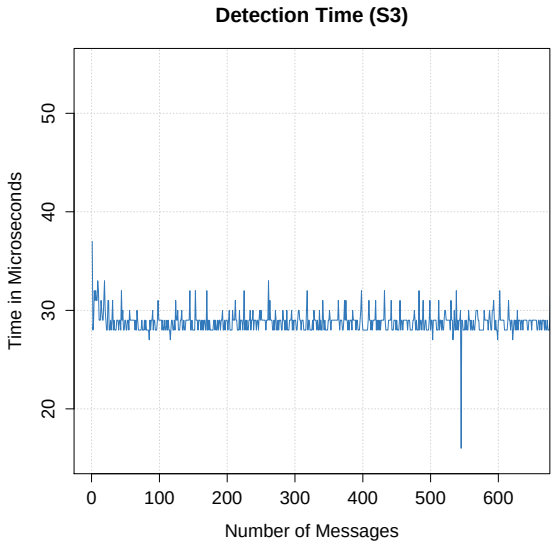


Figure 7.10: Detection time in S3.

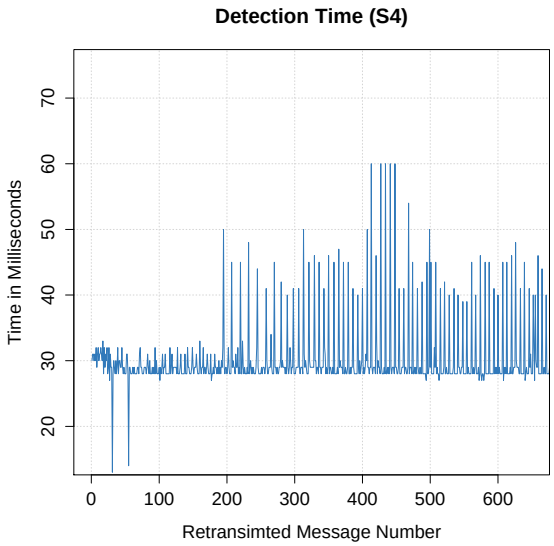


Figure 7.11: Detection time in S4.

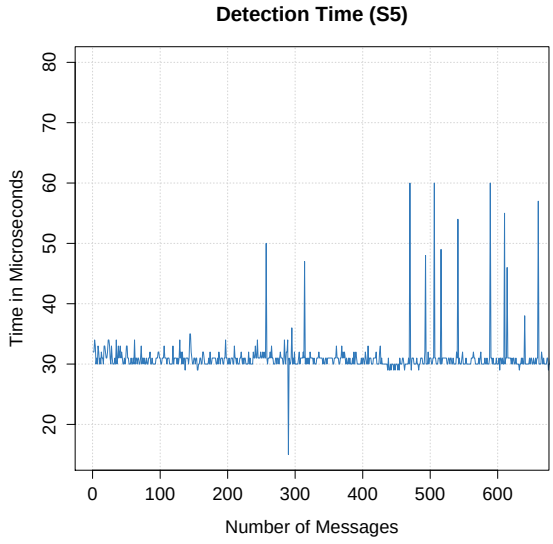


Figure 7.12: Detection time in S5.

Table 7.4: Average End-to-End Delay

Filter State	S1	S2	S3	S4	S5
Disabled	15.112	19.552	17.923	378	440.56
Enabled	14.91	19.708	18.521	417	446.01

(c) during the evaluation the attack traffic has not been blocked, in order to demonstrate the robustness of the proposed mechanism and measure the end-to-end delay between the terminals of Alice (caller) and Bob (callee).

Considering the aforementioned results in conjunction with the fact that Bloom filters can be implemented in hardware and thus minimize further the overheads introduced, it can be deduced that the detection time/overhead introduced by the proposed mechanism is negligible. The most important fact, however, is that flooding attacks have a significant impact to memory resources and therefore a small amount of overhead is acceptable in order to protect the system from a DoS event.

The effectiveness of the proposed scheme, as happens with most detection methods, depends on the rate of false alarms (negatives and/or positives). To this direction the thresholds (T_{alarm}, $T_{signle1}$, $T_{single2}$) were defined according to the "procedure" presented in Section 7.2.3.

Particularly, in order to train the proposed mechanism (thresholds definition), the first scenario (normal traffic) has been executed for one hour. If the proposed mechanism was employed by a different provider, the training phase should be repeated in order to take into account the different traffic characteristics (e.g. peak hour). Using the DDoS threshold estimation procedure (see Algorithm 7.2 and Function 7.3), the threshold T_{alarm} for the tested system was estimated to: $T_{alarm} = 180$, where $T_{sd1} = 150$, $Nd1 + URT1 = 30$ and $\delta = 0$, while the values of $T_{signle1}$ and $T_{signle2}$ were set to 6 and 8 respectively.

Concerning detection accuracy, the results have shown absence of false alarms. However, in cases where the thresholds have not been set to the appropriate values, the proposed scheme generated false alarms, either negative or positive. For example, if for some specific reason (e.g. mother's day) the normal traffic exceeds the thresholds, the proposed scheme would generate false positives alarms. Nevertheless, in such cases the triggered (false) alarms could have a "positive effect" as they provide a warning about the excessive use of system resources.

7.2.6 A comparative Study with Proposals from Other Researchers

Other researchers' proposals for detection of DoS attacks are described in Chapter 6. Summarizing, the majority of these mechanisms are not focused on SIP's specific properties and they either adopt resource demanding cryptographic techniques or they cannot handle attacks against specific entities. It is important to stress that the proposed mechanism differs from other solutions in the following aspects: (a) the monitoring/recording of the incoming traffic is performed through an efficient and cost effective mechanism based on Bloom filters and (b)

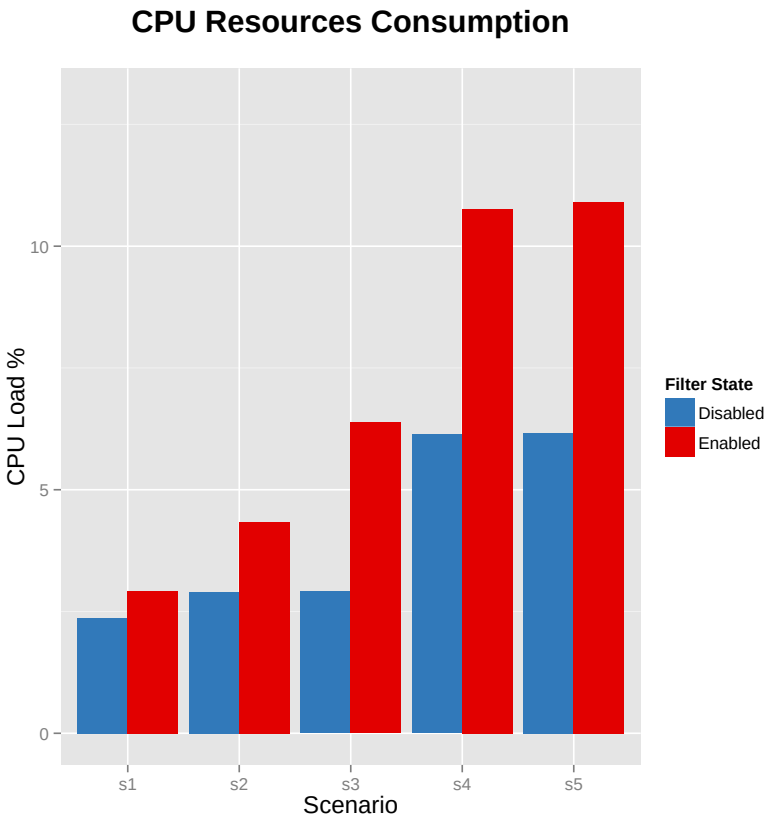
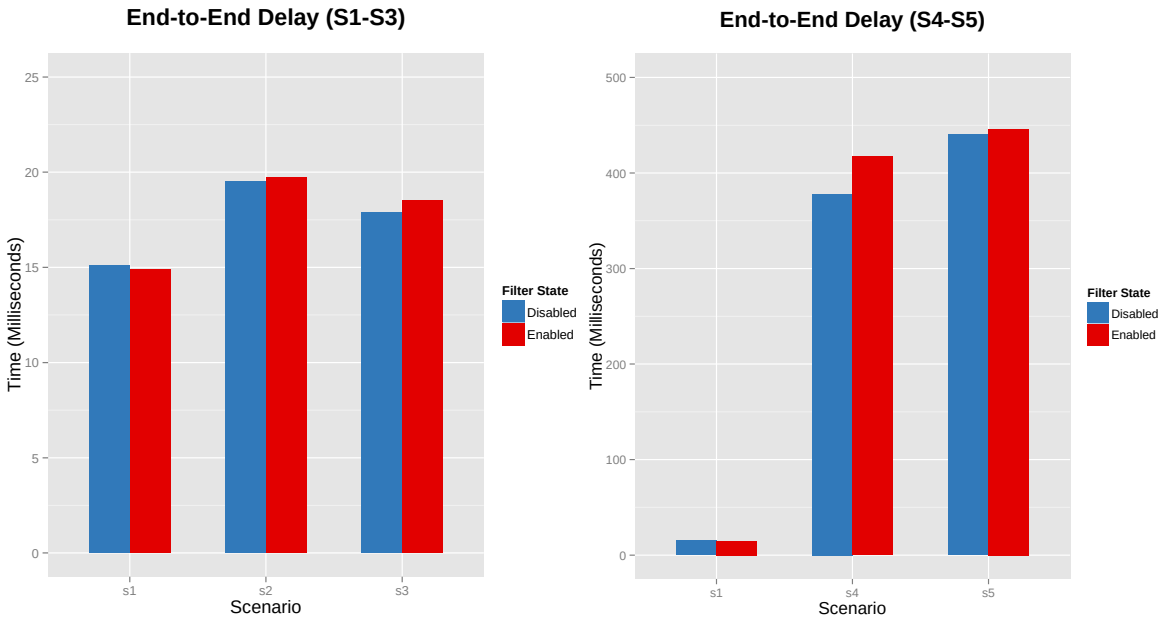


Figure 7.13: CPU resources consumption.



(a) Scenario 1 to 3.

(b) Scenario 4 and 5.

Figure 7.14: End-to-end delay.

the introduction of the SIP specific "session distance" metric is less computational demanding as compared to the solutions described in Chapter 6.

7.3 Protection Against Threats Violating Authenticity and Availability

7.3.1 Introduction

According to Chapter 5, acts that violate message's authenticity and system's availability include SIP signaling, impersonation/spoofing, replay, MiM and flooding attacks. Since the employment of 3GPP's proposed security mechanisms cannot effectively deter an internal and in some cases an external attacker from launching the specific range of attacks, the development of a detection/prevention scheme is more than necessary.

The development of a comprehensive cross layer mechanism for the detection and prevention of a wide range of attacks against SIP-based environments is proposed in this section. The main objective during the development of this scheme was to provide a lightweight mechanism that does not employ resource demanding security protocols nor expensive mathematical calculations. Thus, it can support low end UEs and it does not introduce significant overhead to the home network's deployments. Besides, it is transparent to system's and user's operation, it does not impose any modifications on them and it can be easily deployed in both IMS and VoIP infrastructures.

Specifically, all the messages are first checked for their authenticity using data gathered from layers 2, 3 and 5 of the internet protocol stack. If the SIP registration message is authenticated then a bind is created which correlates information of the aforementioned encapsulated protocols. This bind is stored into a table with the help of a Bloom filter. All the incoming messages are checked against the entries of this table thus detecting all spoofed messages. However, non-spoofed messages are not always legitimate since there are flooding attacks which can be launched without forged SIP requests. Thus, a second statistical module is utilized that detects deviations among all bindings with respect to their traffic behavior. When a binding satisfies a rule that has been created during the training period of the mechanism, then the corresponding messages are dropped and their sources are blacklisted.

It is worth noting that the mechanism is not restricted in detecting only INVITE or REGISTER message floods but it can also effectively deter constant and increasing rate floods launched with any of the SIP's available request methods. A classification of flooding attacks in VoIP/IMS environments is also presented in order to highlight all the different cases that the intrusion detection and prevention systems have to confront.

7.3.2 A Classification of Flooding Attacks in VoIP/IMS Environments

This section provides a classification of flooding attacks and thus facilitates the analysis of the attacks in conjunction with the proposed Intrusion Detection and Prevention System (IDPS).

The behaviors have been classified in terms of (i) the access level of the attacker and (ii) the type of the messages (registration or other type of request). Additional parameters that have been considered are whether the attacks (iii) are launched from single or multiple sources, (iv) use messages that are spoofed with fixed or different IP addresses and (v) include or not authentication vectors. Through this analysis it is possible to deduce the attacker's objective and also the range of the attacks that must be mitigated.

Concerning the access level, an attacker can be Internal (IA) or External (EA). Especially in IMS environments, they can launch attacks through the security tunnels when AKA with IPSec is employed. On the other hand, the EA does not have any legitimate subscription to the server. However, the latter can launch flooding attack since the first registration request is sent without any protection both in VoIP and IMS environments according to specifications [3] and [2] respectively. Furthermore, a discrimination between registration messages and the rest of SIP requests is introduced, due to their crucial difference: The registration requests are sent without security protection while almost all others require authentication such as IMS AKA with IPSec, SIP Digest or SIP Digest with TLS. Such attacks can be launched with spoofed or legitimate messages.

Identifying whether an attacker uses a fixed SIP IP or not (the same attacker utilizes the same SIP message with variable "From" header), enables the determination of the type of the attack and its target (e.g. resources exhaustion, brute force, end-user flooding). These different categories can be further divided relating to whether a message contains a valid authentication string or not. Such a discrimination facilitate the identification of the attackers' target (e.g. CPU resources exhaustion, end-user flooding, etc.). Finally, there are two main types of DoS attacks: the distributed (DS) and the single source (SS). This final classification is of major importance because the attacker can "silently" flood the servers utilizing lower rate flooding attacks from different machines (zombies) at the same time (see Section 5.4.2). Fig. 7.15 depicts the above mentioned classification tree of flooding attacks. The circles denote the target and the specific type of the attack.

7.3.2.1 Internal Attacker

A legitimate subscriber may act maliciously (as an IA) by launching flooding attacks from SS. The utilized messages can be either registration or non-registration requests. If the attacker uses different spoofed IPs in the malicious messages, then he probably launches a SYN syndrome attack or a resources exhaustion attack (see Section 5.4.2) by forcing the server to open an enormous amount of encrypted connections (over Dx, Cx) with the HSS and challenge string calculations. This observation is based on the fact that the attacker avoids the reception of all responses and also tries to allocate server's memory resources per different IP address.

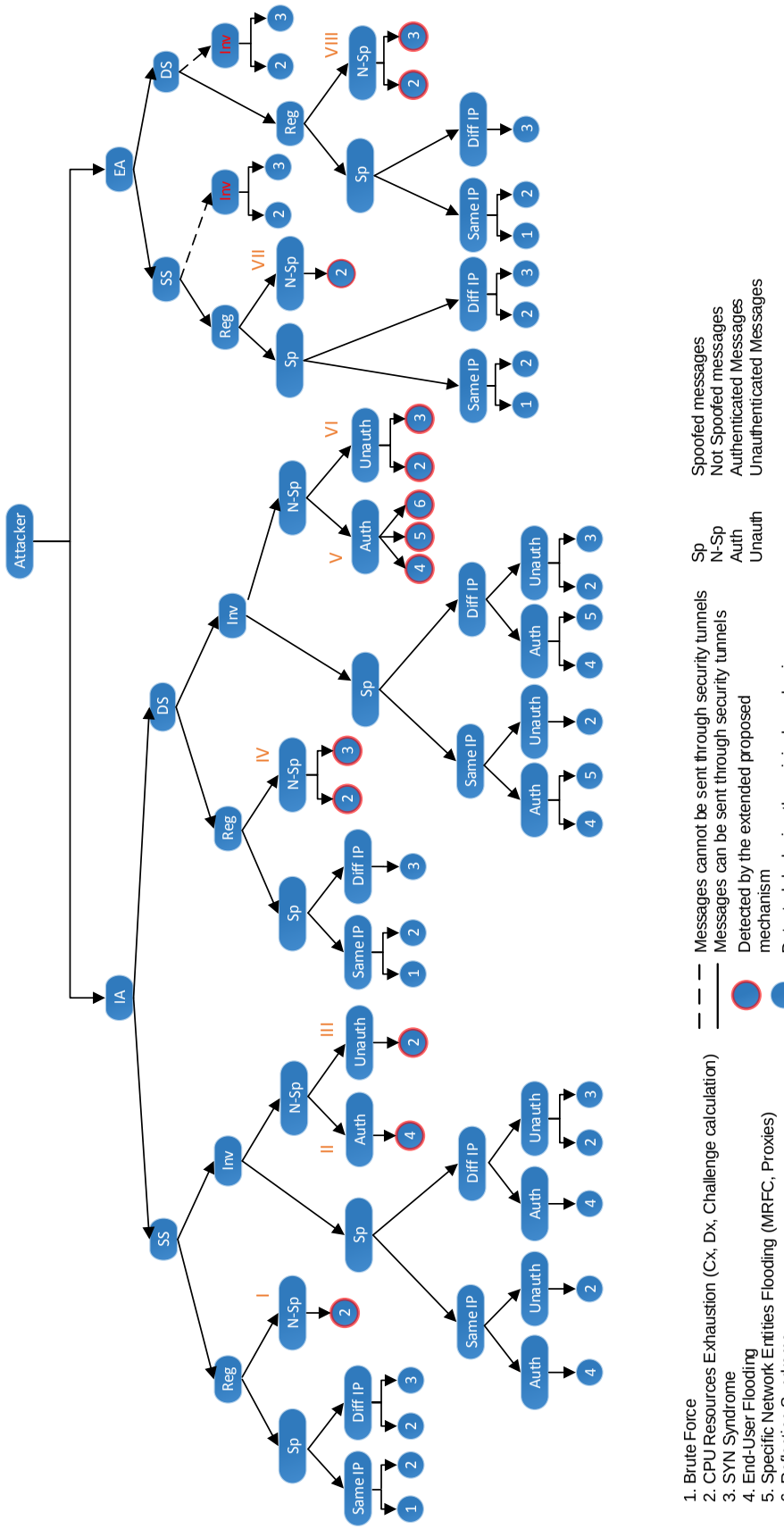


Figure 7.15: Flooding attack classification tree.

In the case where the attacker utilizes spoofed registration messages with the same IP address in all messages, it can be deduced that he launches either a brute force attack trying to break a user's password or a resource exhaustion attack as formerly noted, utilizing a powerful attacking server. This is based on the fact that the attacker needs to gather the server's 401 and 200 OK responses.

Another SS attack can be launched by an attacker utilizing other SIP signaling messages such as the INVITE request. Depending on the type of the DoS, the attacker decides to spoof or not the specific INVITE message. When an end-user is the target of the attack, the malicious signaling traffic must be authenticated so the attacker does not spoof the IP addresses. Therefore, an enormous amount of apparently legitimate (the originator is a subscriber and the messages have been authenticated) traffic reaches the user's UE causing DoS. If the messages are not authenticated they can also lead to DoS as a resources exhaustion attack.

All the above mentioned cases of attacking possibilities exist and can be launched from multiple sources either as an invite reflection or zombie attacks (see Section 5.4.2). In terms of classification, SS and DS differ only in the effects they can induce to the network/servers due to the massive amount of traffic that the latter can handle. Therefore, when the attacking entities forge registration messages they can achieve not only a resources exhaustion attack but also a SYN syndrome attack. The reason is the enormous amount of different requests (i.e. different IP address per request) that reach the servers.

Also in DS attacks, where invite or other non-registration requests are utilized, the core network entities are also threatened due to the large volume of traffic and resources deployed by the attackers. If the requests are deliberately unauthorized they can cause resource exhaustion to the core servers, as in SS attack, and also a SYN syndrome effect (many IP addresses). The same is true for not spoofed SIP messages with random IP addresses. Moreover, a reflection syndrome can be achieved with not spoofed messages because the innocent servers/UEs are involved without any spoofed headers; note that only the originator (namely the attacker) spoofs the "Contact" header. This is also the case when the messages are spoofed with fixed IPs with the exception of unauthenticated ones: the SYN syndrome in this case is not possible because numerous messages reach the servers but with the same IP, hence no more memory resources will be allocated.

7.3.2.2 External Attacker

An EA can launch flooding attacks both from SS or DS. Basically, an EA does not have a subscription to the server and thus many of the attacks cannot be launched. As a matter of fact, when AKA IPSec is employed, the tunneling is mandatory for every session between the P-CSCF and UE except for the first registration message, as the IMS's specifications [18] defines. Therefore, non-registration requests can be forwarded to the proxies only when SIP Digest, GIBA [80] or NIBA [81] are employed. These cases are represented with dotted arrows in Fig. 7.15. Moreover, the non-registration requests originated by an EA can only be spoofed and unauthenticated. The authenticated ones fall into the IA's category. Therefore, either SS or DS

flooding attacks can lead to DoS by exhausting CPU resources or by a SYN syndrome if the IP addresses have been chosen randomly.

As described in Section 5.3.3, an EA can flood a target with stolen authentication strings (replay attack) when SIP Digest is employed, since the calculation of responses on this security protocol does not depend on IP addresses. If the replay attack is not launched or it is not successful, the large volume of unauthenticated traffic can lead to CPU resources exhaustion in the case of fixed IP addresses. Otherwise, the randomly chosen IP addresses add the probability of a SYN syndrome attack through a large amount of half-open connections (server maintains a specific amount of memory per IP for a predefined and fixed period of time).

However, registration messages can be forwarded from an EA and processed by the signaling core due to the lack of authentication requirements. Taking into account the aforementioned facts about the registration messages, it can be deduced that their effects on the architecture are exactly the same in both cases, irrespective of whether the attacker is internal or external.

7.3.3 The General Concept of the Proposed Mechanism

Threats in SIP/IMS environments may originate from different layers, since the attacker attempts to exploit more than one of the protocol's vulnerabilities. It is therefore important for the IDPS to be able to detect such behaviors before they become a threat for the higher layers and affect multimedia services. The proposed IDPS employs a table in order to maintain specific information for every registration request from layers 2, 3 and 5 of the protocol stack with every row maintaining the specific requests' values after their successful registration. Table 7.5 presents a simplified instance of such a table. The columns "C" and "Variables/Threshold" are utilized for detecting flooding attacks. A detailed description can be found in the following sections. A decision tree has been also introduced in order to facilitate the discrimination between fake and legitimate messages.

Table 7.5: A Simplified Instance of the Mechanism's Cross-Layer Correlation Table

	C	UE	IP Address		IMPI/IMPU	Method	Variables/Threshold
E0	Counter	MAC_{12}	IP_{12}	SIP_{12}	ID_{12}	REGISTER	Flooding Detection: Variables/Thresholds
E1		MAC_{24}	IP_{24}	SIP_{24}	ID_{24}	REFER	
⋮		⋮	⋮	⋮	⋮	⋮	
En		MAC_n	IP_n	SIP_n	ID_n	INVITE	
		Layer 2	**Layer 3**		**Layer 5**		

An initial version of the proposed mechanism was presented in [38]. The extended version presented here consists of two distinct modules: the first one is responsible for registration requests while the other module for all remaining requests. It has been designed not only to prevent spoofing attacks, such as SIP signaling, UE and User ID impersonation and MiM, but also attacks against system's availability such as flooding attacks. It is true that many flooding attacks utilize spoofed messages and they can therefore be mitigated. However, they can be

also launched without spoofed messages, for instance though a security tunnel. Considering Fig. 7.15, we can pinpoint (red circles) eight cases of flooding attacks that the initial version of the mechanism [38] and the conventional firewall are not able to deter:

(I) If a UE and a user ID have never been registered before, i.e. it is the pre-registration message, this message will be dropped. This happens because there is not any record for this specific combination. Thus, the UE cannot be registered. On the other hand, if the IDPS gives unprotected access only for registration messages, the architecture will be vulnerable to flooding attacks.

(II) In case of an IA who authenticates all his messages without forging them, neither a conventional firewall nor a security mechanism such as IPSec can detect them during a flooding attack.

(III) These requests are not spoofed but they are unauthenticated. Taking into account that the IA is already registered he can utilize his legitimately established security tunnel to send them. According to specifications [18], after the tunnel establishment, authentication vectors are not included in SIP messages. Hence, flooding attacks in such a case cannot be deterred.

(IV) This case is similar to (I)

(V) This case is similar to (II)

(VI) This case is similar to (II)

(VII) and (VII). These cases are similar to (I)

The above mentioned attacks are addressed by the proposed IDPS. More specifically, all cases depicted in Fig. 7.15, can be detected and mitigated through the deployment of the second module (Module II).

7.3.4 Mechanism's Architecture

The proposed mechanism's architecture is presented in Fig. 7.16. The first module handles every incoming message, takes decisions about its originality and if it will be routed to its destination or not. The main concept is based on cross layer binding between six values that can be gathered from layers 2, 3 and 5 of the network protocol stack. These values correlate a specific UE with a session, a set of IP addresses and the identities of the subscribers. For instance, the frames located at the data link layer (layer 2) bears the network or MAC address of the utilized UE. Furthermore, the binding among the IP address of the 3rd and 5th layer and the MAC address must be unique at a specific point of time. For every incoming message, a tuple E_i is generated $\forall i \in \{0, \ldots, n\}$, where n is the number of incoming messages and $n \in \mathbb{N}$. Every E_i passes through the spoof checking module which decides if the message is legitimate and thus if it will be forwarded to the second module or it will be dropped according to some rule of the Policy Enforcer (PE). The latter holds a blacklist of known malicious E_i. An incoming

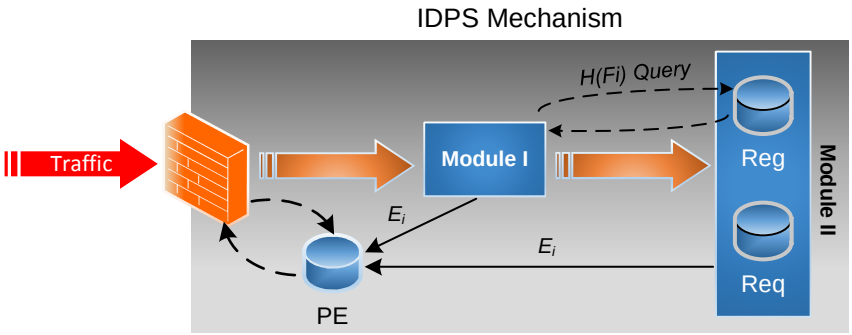


Figure 7.16: Proposed IDPS's architecture.

message is firstly checked for existence in the PE's list. Only the non-listed messages should be forwarded and handled by the next modules.

Afterwards the legitimate E_i are fed to the second module which consists of two tables: the registration table for holding registration messages' data and the request table that holds the data of all the other requests. Every E_i is stored to one of these tables, depending on the type of the request.

The position of the table where a tuple must be stored is calculated according to the Bloom filters theory [115] described also in Section 7.2.1.1. The counting Bloom filter is employed in order to avoid searching and sorting through the second module's tables: The first column in both tables is a counting Bloom filter. The input to the hash functions is the tuple $F_i = MAC_i, IMPI_i, IP_i$, where $F_i \subset E_i$. These three values denote a unique combination that is extracted from every E_i.

Therefore, the precise position for every user per UE/IP on the tables is the value $H(F_i)$. Both tables have eleven different columns as depicted in Table 7.6: C is the counting Bloom filter that provides the mechanism with information about the number of messages per subscriber/UE and eliminates the time needed for detecting a specific tuple's position. The values MAC, IP, SIP-IP, denote the corresponding address at layer 3, 4 and 5 of the network protocol stack that have been involved in a specific request. The IMPI/IMPU holds the private/public ID of the corresponding incoming message. The TS holds a timestamp and the $T.Dist$ the time distance between the last two timestamps that can be calculated by: $T.Dist = TS_1 - TS_{i-1}$, where i is the number of messages that were stored in a specific row. The $Init.D\_Avg$ and $Curr.D\_Avg$ denote the initial and current T.Dist value respectively. Finally, the Trs value is a threshold for alarm triggering in single source flooding attacks. The purpose and use of these values is described in the next paragraph. The monitoring method is depicted in Fig. 7.17, while Algorithm 7.3 provides the pseudo-code of the monitoring procedure and of the way the incoming messages are handled by the components of the proposed mechanism.

Spoofing Detection Method

In order to provide a more accurate description of the proposed mechanism's spoofing detection procedure, the following sets are defined for every one of the network layers involved: $M = \{\text{the set of MAC addresses}\}$, $I = \{\text{the set of IPs}\}$, $S = \{\text{the set of SIP-IPs}\}$, $D =$

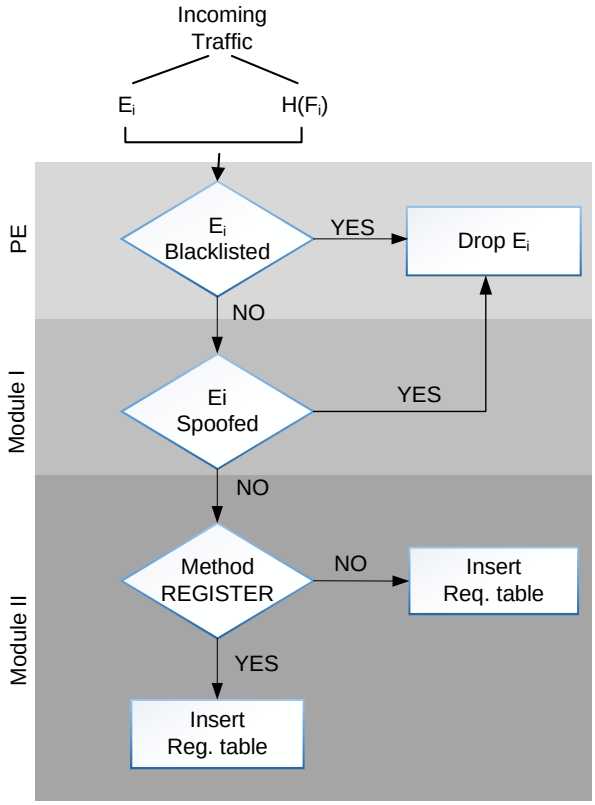


Figure 7.17: IDPS's monitoring method.

Algorithm 7.3: IDPS's message handling pseudo-code.

Input: The extracted E_i tuple from incoming traffic
Output: An E_i that shall be blacklisted or a legitimate E_i that shall be inserted to the second module's table

```
1  modulei(Ei);
2  for each incoming message do
3      if module1 = 1 then /* module1 returns 1 when message shall be routed and 0
       otherwise */
4          Hash = H(Ei.MAC||Ei.IP||Ei.ID);
5          if Ei.method = REGISTER then
6              insert_reg_table(Hash, Ei);
7          else
8              insert_request_table(Hash, Ei);
9          end
10     else
11         drop_packet(Ei);
12         upadate_PE(Ei);
13     end
14 end
```

{the set of IMPIs and IMPUs}, $A = \{$the set of SIP methods$\}$. Moreover, if $R = \{$REGISTER$\}$ then $R \subset A$. Therefore, $E_i = \{m_i, i_i, s_i, a_i, d_i\}$, where $m \in M$, $i_i \in I$, $s_i \in S$, $a_i \in A$ and $d_i \in D$. Also, $K_i = \{m_i, i_i, s_i, d_i\}$ and $O = \bigcup_{i=1}^{n} K_i$ and finally $W = \bigcup_{i=1}^{n} E_i$.

For every incoming message the values E_i and $H(Fi)$ are generated. The spoofing module (module I) detects the position $H(Fi)$ on the registrations' table and retrieves the corresponding C_i's data. Let E_c be the corresponding tuple. The mechanism proceeds with the execution of the following procedure between tuples E_i and E_c (Fig. 7.18).

If $E_i \cap E_c = \emptyset$ and $a_i \in R$, then E_i corresponds to a new registration procedure, therefore there is no identical set in W. If the specific message has been authenticated and $i_i + S_i = s_i$ (intra-packet check) then the corresponding tuple will be forwarded to module II. There, it will be stored in the registration table, denoting the first registration and the binding of the specific UE-subscriber for this specific period of time.

If $E_i \cap E_c = \emptyset$ and $a_i \notin R$, then there is no identical set in W, thus the message has been spoofed and shall be dropped. The PE has to be updated as well with the E_i tuple. The UE is actually not yet registered and this is derived from the fact that the two (E_i, E_c) sets do not have common elements.

If $E_i \cap E_c \neq \emptyset$, then at least one of the m_i, i_i, s_i, a_i, $d_i \in E_c$.

i. Let only $m_i \in E_c$, then i_i, s_i, a_i, $d_i \notin E_c$. Therefore the corresponding message shall not be processed because this case corresponds to an identity theft attempt (see Section 5.3.1) or the IP addresses have been spoofed. The specific E_i must be forwarded to the PE.

ii. Let d_i, $m_i \in E_c$, given that i_i, $s_i \notin E_c$, then if $a_i \in R$ and $i_i = s_i$, the corresponding tuple ($H(Fi)$) shall be updated only when the message has been successfully authenticated. This registration message comes from a UE that has changed location and has been given a new IP address.

iii. Let a_i, $m_i \in E_c$ given that i_i, s_i, $d_i \notin E_c$. If $a_i \in R$ then the corresponding registration message has been initiated from the same UE but the subscriber has changed. After the successful registration the corresponding sc has to be updated with the incoming s_i. For instance, a user swaps the Universal Integrated Circuit Board (UICC) with another one utilizing the same UE and proceeds to a new registration procedure. The case where $a_i \notin R$ is covered by (i).

iv. Let only s_i, $m_i \in E_c$, then there is application or network layer spoofing attempt and the corresponding message shall not be processed. The PE must be updated with the specific E_i.

v. Let only i_i, $m_i \in E_c$, then it is straightforward that $s_i \in E_c$ and thus there is an application layer replay or SIP signaling attack and the message shall be dropped. The PE must be updated. The attacker has reused a previously gathered SIP message from another subscriber. The attacker's objective is to bypass authentication mechanisms.

vi. Let i_i, m_i, $s_i \in E_c$, the message includes an IMPI/IMPU from another subscriber. This identity theft attempt comes from an IA and the message shall be dropped. The E_i is forwarded to the PE. We can assume that the attacker is an insider because a registration for his UE already exists in the table (the only element that does not belongs to E_c is the d_i). This behavior may enable the attacker to charge the provided service to the actual IMPI/IMPU owner.

vii. Let m_i, i_i, s_i, $d_i \in E_c$, then also m_i, i_i, s_i, $d_i \in K_c$. Then the sets K_i and K_c are identical ($K_i = K_c$) and the message is a legitimate one and shall be processed. When $K_i = K_c$, the message is legitimate irrespectively if $a_i \in E_c$.

viii. Let only $i_i \in E_c$ then m_i, s_i, a_i, $d_i \notin E_c$. The message that corresponds to this specific tuple is spoofed and shall be dropped. The PE function must be updated.

ix. Let i_i, s_i, $d_i \in E_c$. If $a_i \in R$ then the corresponding message shall be processed because it can be considered as a legitimate one. The subscriber has initiated a registration procedure utilizing a new UE and the tuple has to be updated (with the new MAC address) after a successful authentication. If $a_i \notin R$, then the IPs of both protocols (SIP and IP) are spoofed or there is an ARP poisoning attempt. The correspondence between MAC and IP has changed and the sets $E_i \neq E_c$ and $K_i \neq K_c$.

x. We know that $K_c \subset E_c$ because $a_c \notin K_c$ and consequently neither $a_i \notin K_c$. Therefore if $K_i \cap K_c = \emptyset$, given that $E_i \cap E_c \neq \emptyset$, it is derived that only $a_i \in E_c$. Then, if the specific $a_i \in R$ and $i_i = s_i$ the message comes from a UE that has initiated a registration procedure for the first time. The tuple that corresponds to the specific E_i has to be updated ($i_i \rightarrow i_c$ and $m_i \rightarrow m_c$) and stored to the registration table after a successful authentication. Otherwise, if $a_i \notin R$ or $i_i \neq s_i$ the message is spoofed while an unregistered UE tries illegally to forward a message or to be registered with forged IP.

xi. Let only the $s_i \in E_c$ then m_i, i_i, a_i, $d_i \notin E_c$. The message is spoofed at the application layer. This can be a replay attempt or a SIP signaling attack initiated from an external user. Particularly, the attacker is and outsider since none of m_i, i_i belong to E_c, thus the specific UE has not been registered until then. The PE is informed for that action.

If the crosschecking between E_i and E_c concludes that the message is not spoofed, then the E_i is forwarded to module II. Even though the specific message may not be spoofed, it is not known if it is also a legitimate one (it may be part of a flooding attack).

Flooding Detection Method

The second module is fed with the E_i that has successfully passed the cross-correlation checking of module I. Every E_i has always its own position on the table and thus the values are overwritten except for $Init.D\_Avg$. The latter value is calculated during the initial handshakes of a specific UE with the server. If the $T.Dist$ between two consequent messages is smaller than the average ($T.Dist\_Avg$), the Trs value is incremented by 1. The $T.Dist\_Avg$ is calculated

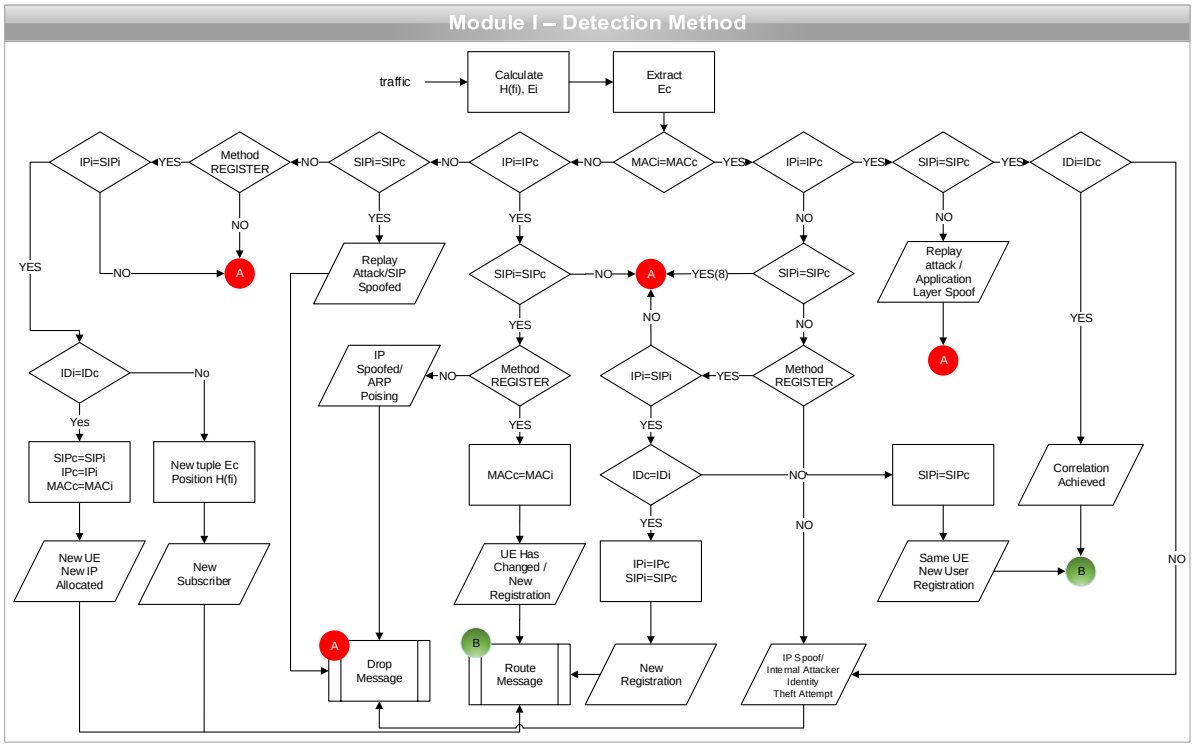


Figure 7.18: Module I - Spoofing detection method.

during a training period of 30 minutes in a normal traffic environment:

$$T.Dist\_Avg = \sum_{i=1}^{m} T.Dist_i \Big/ q \tag{7.6}$$

where q is the number of $C_i \neq 0$. An alarm is triggered (SSalarm) when Trs exceeds a predefined number of messages that are below the $T.Dist\_Avg$. The specific E_i tuple is send to the PE in order to deny access to the corresponding UE.

In case of distributed flooding attacks, the attacking entities may flood the server with low rate of consequent messages so as not to exceed the $T.Dist\_Avg$ value. Such a behavior can be detected utilizing the $Init.D\_Avg$ and $Curr.D\_Avg$ values in conjunction with the average of the incoming messages ($Init.C\_Avg$) and the sum of them ($Init.C\_Sum$). The $Init.C\_Avg$ and the $Init.C\_Sum$ can be measured during the training period by utilizing the functions 7.7 and 7.8 correspondingly:

$$Init.C\_Avg = \sum_{i=1}^{m} C_i \Big/ q \tag{7.7}$$

$$Init.C\_Sum = \sum_{i=1}^{m} C_i \tag{7.8}$$

$$Curr.C\_Sum = \sum_{i=1}^{m} C_i \tag{7.9}$$

$$Init.C\_Sum\_Avg = \sum_{i=1}^{m} init.C\_Sum_i \Big/ k \tag{7.10}$$

where k is the number of executed training procedures.

If Function 7.9 grows with an unusual rate, an alarm is triggered (DSalarm1). A tolerance rate (tr) should be estimated according to the server's capabilities during high traffic periods. Thus, if the $Curr.C\_Sum > Init.C\_Sum\_Avg + tr$ then a DS flooding attack is under way. As already mentioned, the attacking entities cannot be detected by calculating only T.Dist since it may not be exceeded and the alarm not triggered. Therefore, the suspicious subscribers are those with current value of C_i greater than $Init.C\_Avg + tr2$. Also attacks can be prevented by detecting variations between $Init.D\_Avg$ and $Curr.D\_Avg$ of every row. An alarm is triggered when the fraction in Function 7.11 tends to a predefined number x.

$$\lim_{Curr.D\_Avg \to 0} Curr.D\_Avg / init.D\_Avg = x \tag{7.11}$$

Therefore the decrement of the average response time of the specific tuple is calculated. If that happens, for instance, with a rate of 60%, namely $x = 0.4$, during the first DS flooding alarm (DSalarm1), the specific E_i has been involved in the attack (DSalarm2) and thus the PE must be informed in order to restrict its access to the server. Another attack case that can be detected by 7.11 is the increasing rate (IR) flooding attack [116].

Table 7.6: Requests Table

	C	MAC	IP	SIP-IP	IMPI/ IMPU	Method	TS	T. Dist	Init. D_Avg	Curr. D_Avg	Trs
$H(F_{74}) \longrightarrow$	54	MAC_{11}	IP_{11}	SIP-IP_{11}	clam	INVITE	100	15	14	10	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$H(F_{42}) \longrightarrow$	20	MAC_4	IP4	SIP-IP_4	nvra	NOTIFY	10	5	7	5	5
	60	MAC_7	IP_7	SIP-IP_7	dgen	INFO	1500	12	8	4	7

In such situations the attackers try to bypass a traffic rate-based detection mechanism by gradually increasing the attack rate. Therefore the distance average is decreased gradually until very low values. This can be detected employing Function 7.11 that calculates slight or enormous deviation in UEs' traffic behavior. The detection is illustrated in Fig. 7.19. The SSalarm can detect only constant rate (CR) attacks by calculating Function 7.6, where the attackers send a huge amount of messages per time unit. On the other hand in IR attacks, the gradual increment of flooding rate slowly decreases the $T.Dist\_Avg$ value and thus Function 7.6 tends to be incapable to cope.

Another point of major importance is that Function 7.11 provides the IDPS with the appropriate information for distinguishing between legitimate users and attackers when both trigger the DSalarm in cases that the filter lacks adequate training. In such cases, the distributed source alarm is triggered while the sum of C_i grows bigger over the time. The mechanism may trigger

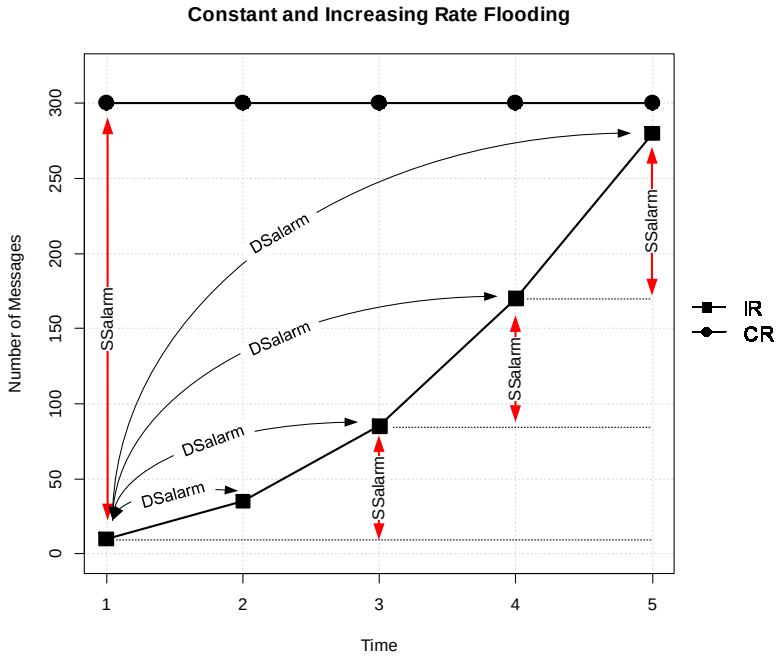


Figure 7.19: Detection of increasing and constant rate attacks.

a false alarm for both legitimate messages and malicious ones but when it will seek for the specific C_i that has been involved in the attack, it will only detect the attackers. This occurs due to the fact that Function 7.11 will still tent to 1 (note that $Curr.C\_Avg \approx Init.C\_Avg$) for the legitimate users and to 0 for the actual attackers.

7.3.5 Evaluation

The proposed mechanism has been evaluated in terms of memory consumption and efficiency in attack detection. The test-bed architecture (Fig. 7.20) consists of the Open IMS platform [117], three legitimate call generators (LCG) and one malicious call generator (MCG) for initiating flooding and spoofed message attacks. The IMS server with the proposed mechanism has been installed on a dual core machine at 2.4 GHz with 4 Gigabytes of RAM, while the call generators have been initiating messages through two 2 GHz dual core machines with 1 and 2 Gigabytes of RAM correspondingly. Finally, the attacker has been installed on a single core machine at 2.2 GHz with 1 Gigabyte of RAM.

Memory Consumption

Three traffic scenarios have been employed for evaluating Bloom filter's memory consumption with two different vector sizes per scenario. The scenarios are presented in Table 7.7. Considering Fig. 7.21a, the allocated memory resources, when the Bloom table employs 2000 registers (which corresponds to 2000 subscribers), is relatively low in all traffic scenarios. However, there is a difference of 140% between the 20 calls per second (cps) and 50 cps. On the other hand, a small increase (33%) in memory consumption is observed when the traffic grows from 50 cps to 70 cps. Almost the same results are observed when the Bloom's size is increased to

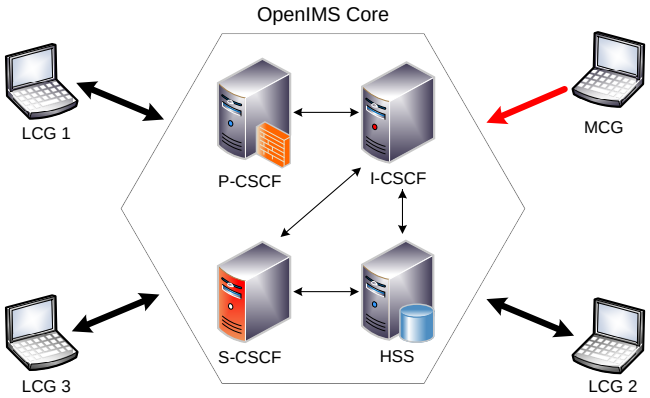


Figure 7.20: The employed test-bed architecture.

Table 7.7: The Scenarios Employed for Evaluating the Proposed IDPS's Memory Consumption

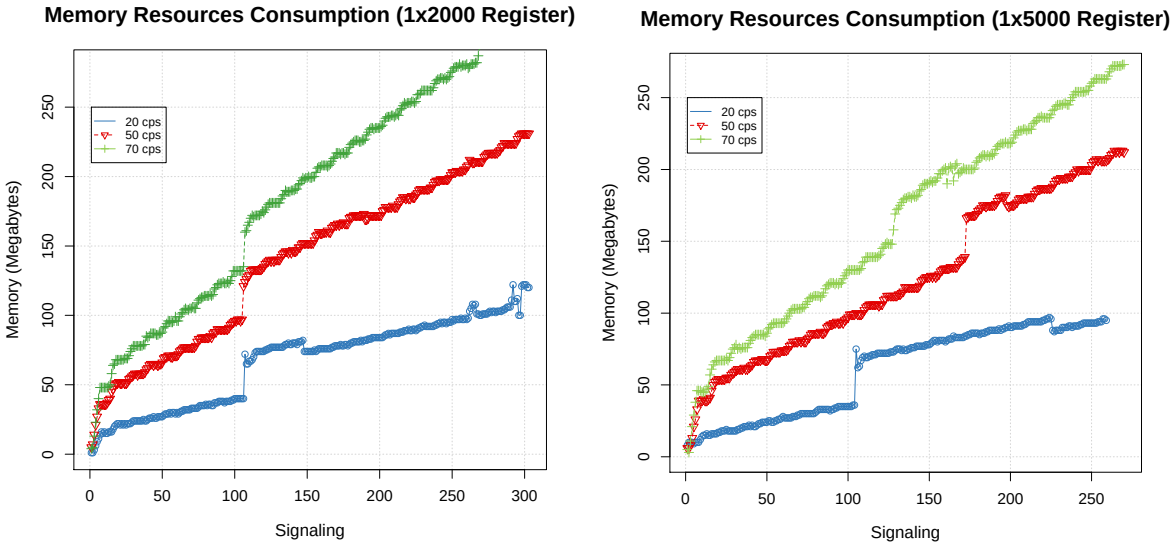
CPS	Size	Description
20 cps	2k	In this scenario the Legitimate Call Generators (LCG) communicate through the IMS Core at a pace of 20 calls per second (cps). The Bloom filter has a size of 2000 and 5000 cells.
	5k	
50 cps	2k	In this scenario the LCG communicate through the IMS Core at a pace of 50 cps. The Bloom filter has a size of 2000 and 5000 cells
	5k	
70 cps	2k	In this scenario the LCG communicate through the IMS Core at a pace of 70 cps. The Bloom filter has a size of 2000 and 5000 cells.
	5k	

5000 registers in order to host the corresponding amount of subscribers. There is a difference of 144% from 20 to 50 cps and 38% from 50 to 70 cps (Fig. 7.21b). It is also illustrated that the different sizes of the Bloom filter do not significantly affect the amount of the allocated memory resources as much as the increment of traffic does. The bar chart in Fig. 7.22 includes a comparison of the average memory consumption between the two different Bloom sizes. In 20 cps scenario both Bloom sizes consume the same amount of memory while the same is true for the 50 cps scenario. Only a slight increment of 6% can be observed in 70 cps scenario between the small and the larger Bloom table.

Accuracy in Attack Detection

For an IDPS it is extremely important to produce the lowest possible number of false positive alarms. A large number of false positives degrades the quality of the provided services while the system causes a denial of service to its own environment and consequently to the legitimate users. On the other hand, a mechanism that produces a large number of false negatives does not affect the legitimate users but it cannot efficiently detect attacks.

For visualizing the performance of the proposed mechanism, the receiver operating characteristics (ROC) graphs are utilized. To design such a graph the following four values are required:



(a) Bloom size of 2000 registers.

(b) Bloom size of 5000 registers.

Figure 7.21: Memory resources consumption in 3 different traffic scenarios.

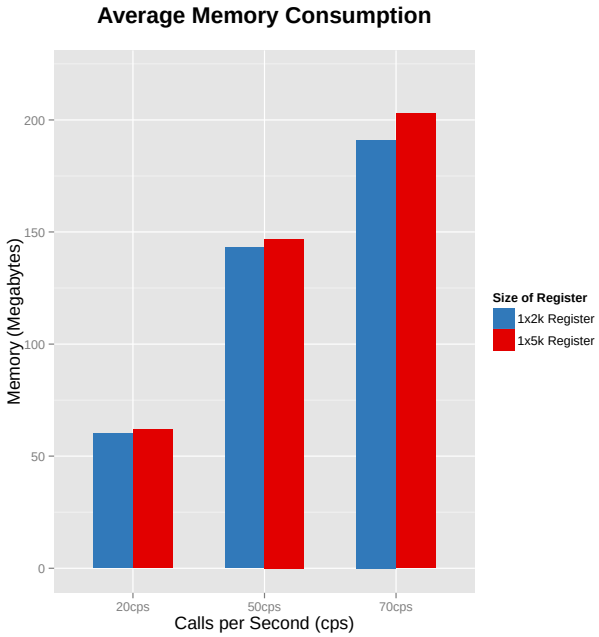


Figure 7.22: Comparison of average memory consumption between different sizes of Bloom filter.

the number of false positives (FP - Legitimate messages which are misclassified as attacks), the number of true positives (TP - Malicious messages which are detected as attacks), the number of false negatives (FN - Malicious messages that passed as legitimate ones) and finally the number of true negatives (TN - Legitimate messages that are correctly classified as legitimate traffic).

The true positive rate (TPR), also called *sensitivity* or *hit rate*, is calculated by Equation 7.12 [118].

$$TPR = \frac{TP}{TotalPositives} = \frac{TP}{TP + FN} \tag{7.12}$$

The false positive rate (FPR), also called *false alarm rate*, is calculated by Equation 7.13.

$$FPR = \frac{FP}{TotalNegatives} = \frac{FP}{FP + TN} \tag{7.13}$$

Legitimate and malicious traffic scenarios have been run on the same test-bed architecture (Fig. 7.20) in order to calculate the TPs and FPs rates. All scenarios have been run with normal background traffic of 10 cps. The flooding and the spoofing detection modules are tested with three and five scenarios correspondingly (Table 7.8). Considering the ROC graph of Fig. 7.23a, we can deduce that the spoofing module almost produces zero number of false alarms and at the same time it has detected all the malicious spoofed messages initiated by the MCG in scenarios S B2, S B3 and S B4.

False negative rates are slightly increased in increasing rate flooding attacks (Fig. 7.23b) due to the fact that the specific attack requires many messages to develop. However, it does not have any impact to the system until it reaches the threshold values. At this point the specific alarm is triggered and the attacker is deterred. The same graph illustrates that the false positives tend to zero in constant rate attacks with a high detection rate (high TPR). On the other hand, considering S B8 we can pinpoint that the mechanism produces a large number of false positives and true positives at the same time. This is due to the inadequate training of the mechanism. This occurs when the scenarios exceed the time of the training period (30 minutes) and thus the collected mean values are much smaller.

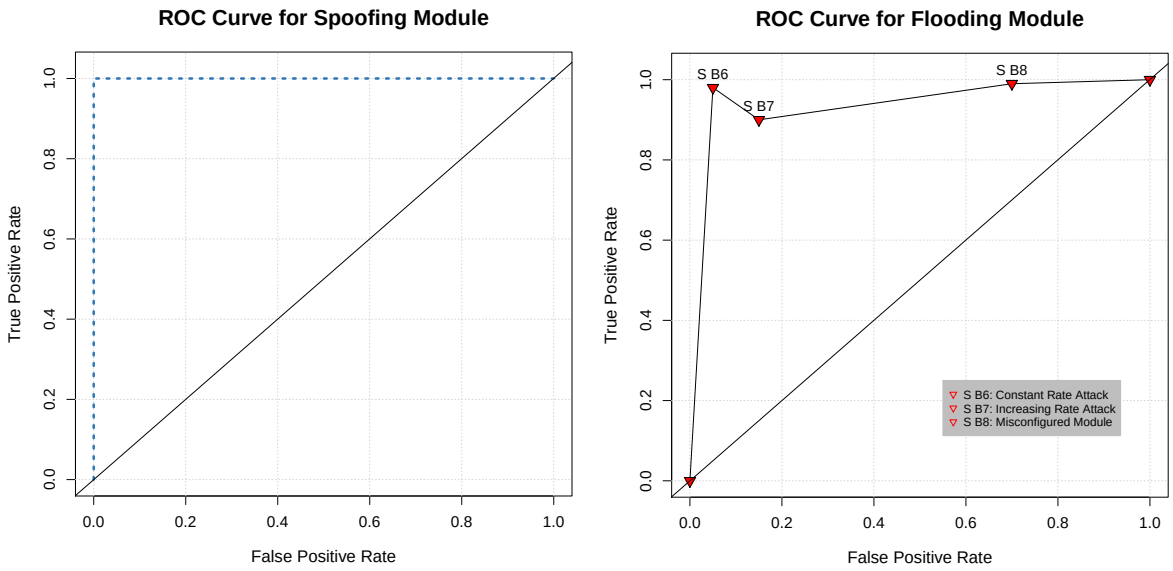
Therefore, only the distributed source alarm is triggered while $\sum C_i$ grows bigger over time. However, the mechanism may inaccurately trigger the DSalrm for both legitimate messages and malicious ones but when it will seek for the specific C_i that has been involved in the attack it will only detect the attackers. This happens due to the fact that Function 7.12 tents to 1 for the legitimate users (while the $Curr.C\_Avg \approx Init.C\_Avg$) and to 0 for the actual attackers.

7.3.6 A Comparative Study with Proposals from Other Researchers

Various different approaches and proposals of how VoIP and IMS environments can be hardened against attacks, can be found in the literature. However, most of them are focused on the detection without being able to actively deter the attackers, while there are only a few that can discourage specific cases of the above mentioned attacks. These protection

Table 7.8: The Scenarios Employed for Estimating the Proposed IDPS's Detection Accuracy.

Testing Point	Sc. No.	Description
Module I	S B1	Legitimate Traffic between the LCG #1 and LCG #2 through the OpenIMS core at a pace of 10 cps.
	S B2	MCG launches 1500 messages with spoofed ID ("From" header) while S B1 is being carried out.
	S B3	MCG launches 1500 messages with spoofed MAC address while S B1 is being carried out.
	S B4	MCG launches 1500 messages with spoofed IP address while S B1 is being carried out.
	SB 5	LGC #3 launches 3000 legitimate messages while S B1 is being carried out.
Module II	S B6	MCG launches constant rate flooding attack when S B1 is being carried out.
	S B7	MCG launches increasing rate flooding attack when S B1 is being carried out.
	SB 8	S B7 while the proposed IDPS mechanism is poorly trained and configured.



(a) Spoofing module's ROC graph for scenarios S B1 to S B5.

(b) Flooding module's ROC graph for scenarios S B6 to S B8.

Figure 7.23: ROC analysis of the IDPS mechanism.

Table 7.9: Comparison of Security Mechanisms

			Prop.		[26]		[23]		[107]		[25]		[105]		[103]		[106]	
Layer	Threat Category	Attack	D	P	D	P	D	P	D	P	D	P	D	P	D	P	D	P
L 5	**SIP Signaling**	BYE	•	•	x	x	•	x	•	•	x	x	•	•	•	•	x	x
		CANCEL	•	•	x	x	•	x	•	•	x	x	•	•	•	•	x	x
		Re-INVITE	•	•	x	x	•	x	x	x	x	x	•	•	•	•	x	x
		UPDATE	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
	Masquerade / ID Theft	UE Impersonation (SIP IPs)	•	•	x	x	x	x	x	x	x	x	•	•	x	x	x	x
		User Impersonation (SIP IDs)	•	•	x	x	x	x	x	x	x	x	•	•	•	•	•	•
	MiM	Registration Expiration	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
		Bid Down	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
		Generic Authentication	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
		Conference Interception	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
	Replay	SIP Replay	•	•	x	x	x	x	x	x	x	x	•	•	•	•	x	x
	Flooding	Non-INVITE	•	•	x	x	x	x	x	x	x	x	x	x	x	x	x	x
		Invite	•	•	•	•	x	x	•	•	•	x	x	x	x	x	x	x
		Single Source	•	•	•	•	x	x	•	•	•	x	x	x	x	x	x	x
		Distributed Source	•	•	•	•	x	x	•	•	•	x	x	x	x	x	x	x
L 3	**Masquerade (L3)**	UE IP spoof	•	•	x	x	x	x	x	x	x	x	x	x	x	x	•	•
L 2	**MiM (L2)**	ARP Poison	•	•	x	x	x	x	x	x	x	x	x	x	x	x	x	x

With **D** is denoted that the mechanism provides only attack detection while with **P** that it can also prevent the attack.

frameworks have been already described in Chapter 6. A comparison among the above mentioned proposals, with respect to their efficiency in detecting and preventing the spoofing and flooding attacks is provided by Table 7.9.

7.4 Protection Against Threats Violating Privacy

7.4.1 Introduction

In VoIP and IMS environments, legitimate users are recognized through pseudonyms issued by the service provider. These pseudonyms can be either the IMS private or public identity (IMPU and IMPI correspondingly). Hereafter, the term private identity (PI) and public identity (PU) will be used for denoting the user's identity in the IMS architecture. These identities are used for message routing and users' identification and authentication.

As described in Chapter 4, the UE sends the user's identity in clear text, during the authentication procedure. If an attacker eavesdrops the communication channel he can easily obtain a legitimate user's identity and link it with his calls and other actions and general preferences. Further to this, an attacker can link the identities with other social networks and real names using for instance "people search engines" such as the www.pipl.com [119]. Such an exposure does not only raise privacy concerns but may also lead to other attack types (e.g., Spam Over Internet Telephony-SPIT or DoS).

VoIP [3] and IMS specifications [18] do not propose any countermeasures for the protection of user identities while the contribution of other research work in this field is considered limited and on top of that requires the deployment of additional infrastructures and key management related techniques that the low-end devices cannot support.

Since the privacy requirements in such environments can be easily violated by eavesdroppers, a mechanism has been developed for protecting users' privacy even from the first hop of the communication and keeping the identity concealed even in untrusted domains. Further, it is free of all the aforementioned requirements (PKI deployments) that introduce overhead and security related concerns (key storing, certificates managment/revocation etc.) to the infrastructure.

By utilizing the privacy enhancing mechanism developed, UEs are able to generate, unlikable, one-time identities for every new session, in a way transparent to the users. It also provides unlinkability and anonymity services to all users participating in a communication session (e.g. caller, callee, call conference participants), provided that the caller member desires a privacy enhancing session.

The developed identity protection mechanism is based on Shamir's key-less scheme [120] and utilizes the commutative functions in order to obscure and conceal users' real identities included in the SIP signaling messages. The commutativity property enables the communicating parties to exchange data without a key or prior knowledge while the computational complexity of the modular exponentiation is equivalent to the discrete logarithm problem (DLP) [121].

7.4.2 The General Concept of the Proposed Mechanism

The basic idea of the specific scheme is based on the commutative property: Let S a non-empty set and an operation between all the elements that exist in S. The operation $\sim$ satisfies the commutative property if $x \sim y = y \sim x$, for all $x, y \in S$. Examples that satisfy the commutative property are: $x * y = y * x$ and $(x^y)^z = (x^z)^y$.

In cryptography, a commutative cryptographic function enables parties to decrypt their own part of the ciphertext without affecting the initial message: Let us assume that Bob, with a key (b), and Alice, with a key (a), need to share a piece of information (r). The commutativity of an encryption function (E) satisfies equation 7.14.

$$E_a(E_b(r)) = E_b(E_a(r)) \tag{7.14}$$

A scheme based on this property has been proposed by Shamir [120] and as illustrated in Table 7.10, it enables two parties to share a secret without the need of a key (key-less scheme).

Alice and Bob pick a random key, *Kalice* and *Kbob* respectively. Alice sends the r encrypted with her key *Kalice*. Bob receives the ciphertext and double encrypts it with his key *Kbob* and sends the result back to Alice. Since function E is commutative, Alice extracts $E_{Kbob}(r)$ by applying E^{-1}_{Kalice} and sends it to Bob. Finally, Bob extracts the r by applying E^{-1}_{Kbob} to the received ciphertext.

The Diffie-Hellman key exchange (D-H) was considered as a possible solution for this scheme but it was not transparent to the underlying SIP infrastructure. Particularly, the employment of D-H requires the exchange of three more messages in addition to SIP's session establishment handshake. Specifically, the D-H provides the communicating entities with a common secret that is generated during the last step of the algorithm:

- After choosing a prime p and a generator g of the $\mathbb{Z}_p$, User A calculates a secret a under $\bmod(p)$ and sends the publics p, g and the encrypted $g^a \bmod(p)$ to User B.
- User B calculates a secret b under $\bmod(p)$ and sends back to User A the $g^b \bmod(p)$.
- Users A and B conclude to the same string K by calculating the $K = (g^b)^a \bmod(p)$ and $K = (g^a)^b \bmod(p)$ correspondingly.

The produced common secret can be used as an encryption key for protecting the communication afterwards (e.g. encrypting the media). Consequently, the employment of D-H in NGN context requires an extended modification of the underlying infrastructure (e.g SIP's establishment handshake) in order for the two entities to share the same secret and inevitably it introduces additional overhead. Furthermore, during the initial steps of the D-H the communicating entities identities remain unprotected.

On the contrary, the Shamir's scheme is used for sharing a secret "message" without the need to exchange a secret key in advance, since it is based on the commutative property. This means that no additional messages are required while users' identities can be protected from the very first step of the session.

7.4.3 Obscuring Identity Scheme Description

In order to "hide" legitimate users' identities from eavesdroppers, a commutative encryption function is utilized for generating one-time meaningless identities that will replace real identities. The proposed protocol uses the commutativity of modular exponentiation and its computational complexity is equivalent to the DLP [121].

Table 7.10: Shamir's key-less scheme.

Protocol Initiation
Alice wants to send a message r to Bob
Alice picks a random key *Kalice*
Bob picks a random key *Kbob*
Message Exchange
1. Alice $\rightarrowtail$ Bob: $E_{Kalice}(r)$
2. Alice $\leftarrowtail$ Bob: $E_{Kbob}(E_{Kalice}(r))$
3. Alice $\rightarrowtail$ Bob: $E_{Kbob}(r)$

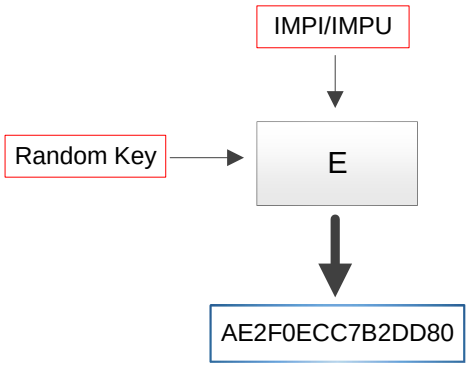


Figure 7.24: Obscured identity generation. The output is a one-time meaningless identity.

Particularly, in order to generate such identities the commutative encryption function is fed with the real identity of the user and a random key (see Fig. 7.24). The output is a meaningless string that is being used as the user's identity. Since the communicating parties, for every newly established session, use a different random key it is possible to generate one-time identities for every established session.

The encryption protocol requires the exchange of three messages between the UE and the proxy and thus the SIP session establishment handshake does not need any modifications in order to incorporate the proposed scheme. For simplicity, it is assumed that the user has the same PI and PU and it will be denoted as "ID". An overview of the proposed scheme is presented in Table 7.11.

Particularly, the UE chooses a prime number p and generates a random number kue such that $1 \leq kue \leq p-2$ and $\gcd(kue, p-1) = 1$. At the same time, the inverse of the key, $kue^{-1} \bmod (p-1)$, is calculated under $p-1$. Then, it raises the user's identity (ID) to the power of kue under $\bmod p$. Afterwards, the UE sends a SIP REGISTER message with the ciphertext $ID^{kue} \bmod p$ as an identity, instead of the real ID, and informs the proxy about the number p. Proxy receives $ID^{kue} \bmod p$ and p, and generates a random ($kproxy$) such that $1 \leq kproxy \leq p-2$ and $\gcd(kproxy, p-1) = 1$ and also the inverse key, $kproxy^{-1} \bmod (p-1)$, calculated under $p-1$. Then, the proxy raises the received ciphertext to the power of $kproxy$ and sends back to UE a "401 unauthorized" response that includes the result $((ID^{kue})^{kproxy} \bmod p)$.

A fresh nonce value is also included in the latter response (it is used by the user in order to generate his SIP Digest credentials). The UE raises the double encrypted ciphertext to the power of kue^{-1}. The inverse modulo of UE's key is computed by $kue^{-1} \bmod (p-1)$. According to the commutative property $(ID^{kue*kproxy}) \bmod p = (ID^{kue*(-kue)})^{kproxy} = ID^{kproxy} \bmod p$.

Then, UE sends a new REGISTER message (according to RFC 3261 [3]) to the proxy including user's credentials and the $ID^{kproxy} \bmod p$. The proxy raises the $ID^{kproxy} \bmod p$ to the power of $kproxy^{-1} \bmod (p-1)$ and extracts the actual user's ID: $(ID^{kproxy})^{-kproxy} \bmod p = ID$ and authenticates the user with the SIP digest [20] since it has all the required values to validate the response (user's identity and credentials included in "Authorization" header).

In case of a successful authentication, the proxy responds with a "200 OK" message including a

Table 7.11: Proposed ID Obscuring Scheme

Protocol Description
UE will register to VoIP/IMS server without disclosing user's identity (ID).
Protocol Initiation
▪ UE generates a random prime p. ▪ UE and Proxy pick a random kue and $kproxy$ correspondingly, such that $keu, kproxy \in [1, p-2]$ and $\gcd(kue, p-1) = 1$, $\gcd(kproxy, p-1) = 1$. ▪ UE and Proxy calculate the inverse of keys kue^{-1} and $kproxy^{-1}$ under $\mod (p-1)$ correspondingly.
Message Exchange
1. UE $\rightarrowtail$ Proxy: $ID^{kue} \mod p$ 2. UE $\leftarrowtail$ Proxy: $(ID^{kue})^{kproxy} \mod p$ 3. UE $\rightarrowtail$ Proxy: $ID^{kproxy} \mod p$

header value ("rspauth") that enables UE to authenticate the server (mutual authentication − server proves that knows user's password). In cases where the UE needs to protect the IDs of all users involved, it executes the same protocol under the same prime number for every different user identity. A new random key is not required since the input to the encryption function is different and consequently the generated ciphertext.

Another point of major importance is the callee's side. Specifically, a malicious user can intercept the callee's communication channel. Thus, obscuring an identity only during the first hop (between UE and proxy) is not enough for ensuring that ID will not be disclosed until the signaling message reaches its destination (Fig. 7.25). To this end, when a user does not trust the callee's channel, may request an end-to-end obscured identity. In this case, the proxy is responsible for initiating the proposed protocol for the communication with the callee, as described earlier in this section.

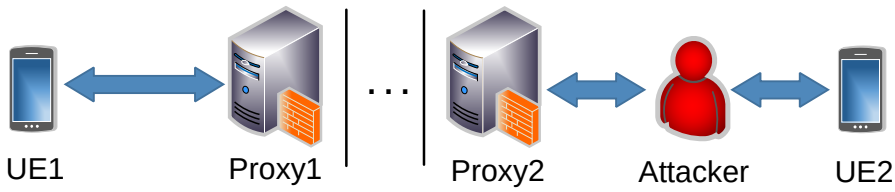
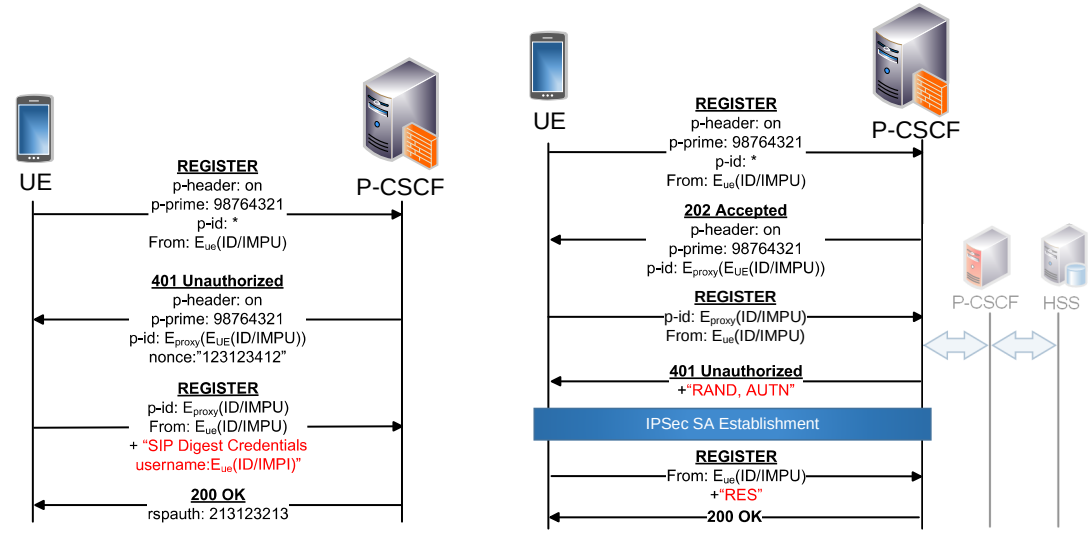


Figure 7.25: The insecure channel is on the callee's side. The identities can be disclosed between proxy2 and UE2 if the UE does not take advantage of the end-to-end privacy option.



(a) Obscuring identity in conjunction with SIP Digest authentication.

(b) Obscuring identity in conjunction with the AKA IPSec authentication.

Figure 7.26: Obscuring identity in VoIP/IMS environments.

7.4.3.1 Extending IMS and SIP to Support the Proposed Scheme

As described in Section 7.4.3, the proposed scheme requires a publicly known prime number. Since UE is the initiator of the registration handshake, it is responsible to choose and send the number to the proxy. Moreover, the latter shall be informed that the upcoming communication will be enhanced with privacy services for the user's ID. Therefore, to incorporate this solution in real NGN services, the SIP's functionality has to be extended by introducing additional headers that enable end users and proxies to utilize the mechanism:

- *p-header*: This header is a flag that indicates whether the privacy scheme will be disabled ("off"), enabled for obscuring caller's identity ("on") or all of the involved users identities ("full"). Finally, the value "ete" is available when the caller desires to protect his identity in an end-to-end fashion.
- *p-prime*: That header holds the mandatory prime number which indicates the field that the computations will take place. It is generated by the handshake initiator.
- *p-id*: It is responsible for bearing the ciphertext until the server extracts the user's real id.
- *p-ids*: The same as "p-id" but for the callee's identity.

A successful registration handshake is illustrated in Fig. 7.26a where the proposed scheme is used in conjunction with the SIP digest authentication. It is worth noting that the new mechanism does not impose any modifications on the original 4-way handshake. On the contrary, when IPSec with AKA (see Section 4.2.3) is used as the authentication protocol, two additional SIP messages are required. This is mandatory due to the identity-wise Authentication Vector retrieval from the HSS in IPSec authentication. Thus, an additional REGISTER request is necessary prior to the normal registration procedure.

Table 7.12: SIP REGISTER Request with the One-Time Obscured Identity

Register Request - Privacy Options Enabled
REGISTER sip:testbed-ims.gr SIP/2.0
Via:SIP/2.0/UDP1 92.168.2.2:5060
From: <**sip: AE2F0ECC7B2DD80**>;tag=8030857
To: <**sip: AE2F0ECC7B2DD80**>
Call-ID: 2019873979
CSeq: 2 REGISTER
Max-Forwards: 70
Expires: 600000
p-header: on
p-prime: 434234123459
Authorization:**Digest username="AE2F0ECC7B2DD80"** ,
realm="testbed-ims.gr", nonce="", uri="sip:testbed-ims.gr", response=""
Content-Length: 0

The private and public users' identities (in bold) are meaningless values that may be different in every new session.

The UE sends a REGISTER message with the "p-header" enabled including the obscured ID. The proxy responds with a "202 accepted" message, providing the UE with a double encrypted ID into the "p-id" header. Afterwards, the UE sends a new registration request which includes the encrypted ID with proxy's randomly selected key. At this point, the Home Network (HN) is informed about the user's real identity and the S-CSCF proceeds with the AV retrieval querying the HSS database. S-CSCF then sends authentication token (AUTN), random number (RAND), confidentiality key (CK) and integrity key (IK) in the "401 unauthorized" response message. Proxy strips out CK and IK and forwards RAND and AUTN to the UE. The IPSec tunnel is now established. After the reception of the AVs, the UE is enabled to calculate the response (RES) in order to be authenticated to the HN. The identity obscuring handshake is depicted in Fig. 7.26b.

Table 7.12 depicts a SIP registration request message when the user has requested the employment of the proposed privacy enhancing scheme. The "p-header" is enabled ("on" flag) and also the "p-prime" is provided to the proxy. The private and public users' identities (in bold) are now meaningless values (AE2F0ECC7B2DD80) that may be different in every new session.

End-to-end Privacy Option

When the end-to-end privacy option is selected, the last hop is responsible for initiating the proposed protocol. Specifically, the proxy chooses the p and generates a key in mod p. Afterwards, encrypts the caller's PU and forwards the INVITE message to the UE2 including the prime. The UE2 double encrypts the received ciphertext (since it has already generated a key under mod p by the reception of the prime) and returns it to the proxy with the "200 OK" response. Finally, the proxy decrypts its part using the inverse of its key and sends an ACK message, including the remaining of the ciphertext ($E_{ue2}(ID)$), to the UE2. The latter decrypts

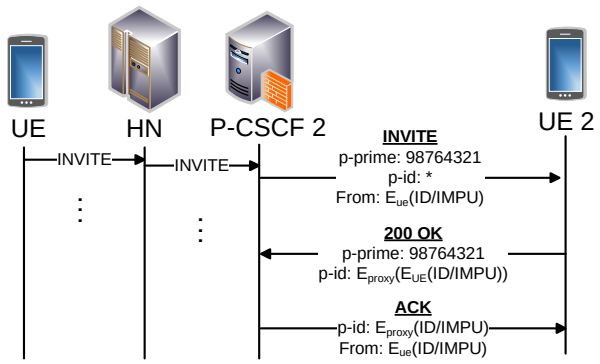


Figure 7.27: End-to-end one-time identity.

the ID as it is encrypted only with its key. The aforementioned procedure is illustrated in Fig. 7.27.

The proxy can also use an anonymous (e.g. *anon@anonymous.com*) user identity, at the last hop, instead of the encrypted one. This can be used when the caller wishes to hide his real identity from the callee.

7.4.4 Evaluation

To evaluate the performance overhead introduced by the proposed scheme, the OpenIMS platform [117] has been utilized. At server's side, the obscuring identity service has been implemented as a module for the P-CSCF server, while several modifications have been made for the whole OpenIMS core to support and utilize the new SIP headers introduced by the proposed solution. At client's side, the proposed scheme has been incorporated into the PJSIP stack [122]. The test-bed architecture is depicted in Fig. 7.28. The OpenIMS platform was installed on an Intel core 2 at 2.4 Ghz machine with 4 GB RAM, while the client was installed on an triple-core AMD at 2.4 GHz with 2 GB RAM. Both systems were running Ubuntu O/S.

The overhead considered was the execution time of the server and the client for generating the random keys and for inverting them under modulo of a randomly chosen prime number. Additionally, the server's response times have been monitored under normal traffic, when 30 registration requests per second reach the service. The client's side response times were also measured at a pace of 2 registrations per second, since this is the upper bound of requests that the client can generate and respond. Finally, the server and client response times have been measured without having enabled the proposed obscurity identities service, in order to compare them with the response times measured when the proposed scheme was utilized and thus identify the performance impact. The evaluation scenarios have considered five different key/ID lengths and are presented in Table 7.13.

7.4.4.1 Server's Evaluation

Scenarios S1 and S2 have been utilized in order to assess the overhead introduced by the privacy module to the P-CSCF server. Considering Fig. 7.29, it is deduced that that the

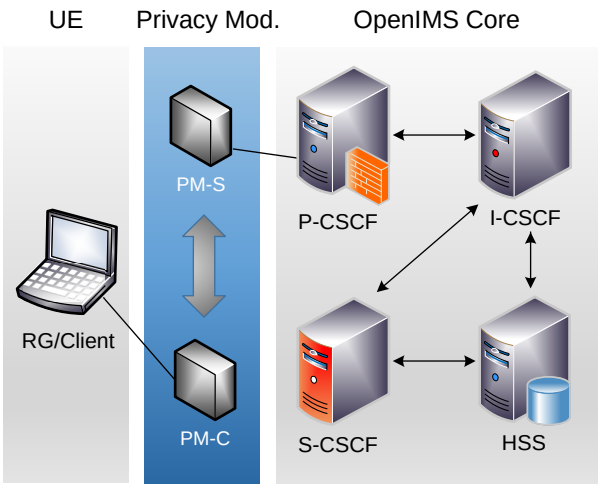


Figure 7.28: The employed test-bed architecture. The server side privacy module (PM-S) has been deployed in the proxy (P-CSCF) while the client side module (PM-C) was developed with the PJSIP stack.

Table 7.13: The scenarios Employed for Evaluating the Proposed Privacy Scheme

Scenario Name	Scenario Description
Scenario 1 (S1)	In this scenario registration requests are initiated from a client with the privacy protection service enabled. The random number's (key) generation procedure is calculated when the ID has the size of 8, 16, 24, 32 and 64 bits.
Scenario 2 (S2)	In this scenario registration requests are initiated from a client with the privacy protection service enabled The key's inversion is calculated when the ID has the size of 8, 16, 24, 32 and 64 bits.
Scenario 3 (S3)	In this scenario traffic of 30 SIP REGISTER requests per second is generated through a Registration Generator (RG). The server's response time is calculated when the ID has the size of 8, 16, 24, 32 and 64 bits.
Scenario4 (S4)	In this scenario, the client initiates 2 registration requests per second (rps). The client's response time is calculated when the ID has the size of 8, 16, 24, 32 and 64 bits.

overhead imposed on the server during key generation is consistent and relatively low since almost all calculations have been executed within a period of 3 to 6 microseconds. As also depicted in Fig. 7.31, the average time required for the server to generate a random key is constantly low for all ID lengths, with a mean value of 4 microseconds at 64bit and 3.76 at 8bit. This slight increase of 6%, proves that the ID length does not considerably affect the key generation procedure and, consequently, the overhead introduced to the communication.

A similar situation was found for the key inversion procedure of scenario S2. The delay imposed on the server during the key inversion was in all cases from 2 to 5 microseconds Fig. 7.30. The mean delay values were in the worst case 6.33 microseconds (64 bit ID length) and 6.11 microseconds in the best (8 bit ID length), as depicted in Fig. 7.32.

7.4.4.2 Client's Evaluation

The random key generation procedure does not seem to delay the registration procedure, as illustrated in Fig. 7.33; the 95% of this calculation required 6 to 8 microsecond to complete. The mean values were also equivalent for all tested ID lengths (Fig. 7.35). An increase of 5.5% from the 8 to 64 bit length is not considered significant.

Similarly, the key inversion procedure has a minor impact on the client's registration. The delay is bounded form 9 to 12 microseconds in the 95% of the executions of S1 (Fig. 7.34). The average delay is also constant form 10 to 10.7 microseconds with an ID length of 8 to 64 bit correspondingly (Fig. 7.36). A difference in mean values of 6.3% also proves that the key length during inversion does not have a significant impact to the client's registration procedure.

7.4.4.3 Response Times

The first three scenarios have been carried out by measuring the functions' execution times inside the client and the server. A more realistic and objective view of the overall overhead is derived by the employment of scenarios S3 and S4, since they were carried out in a normal traffic environment and include network's delay. The delay imposed on the server's and the client's response times under a normal traffic environment (scenarios S3 and S4) is measured, in order to identify the impact of the proposed privacy service when employed in a real environment. This is more important for the server side due to the fact that the P-CSCF deals with many concurrent requests per second. Thus, even a small delay per request may lead to a large overhead per second.

The uniformity of the results when the server handles 30 registrations per second is depicted in Fig. 7.37. For all ID lengths the delay was from 7 to 11 milliseconds. The average delay may experience an increase of up to 32% (Fig. 7.39) compared to the average server's response time when the privacy service is disabled. This is the worst case scenario since the registration procedure involves the prime generation that is the most time consuming calculation. Without loss of generality, a user considers such an increase in the response time acceptable especially since it will happen only during the registration phase.

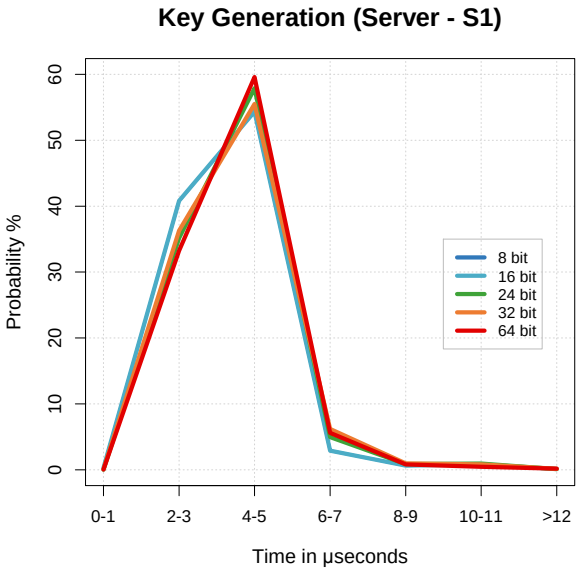


Figure 7.29: Probability densities function of scenario 1. The key generation procedure required from 3 to 6 microseconds for every ID length.

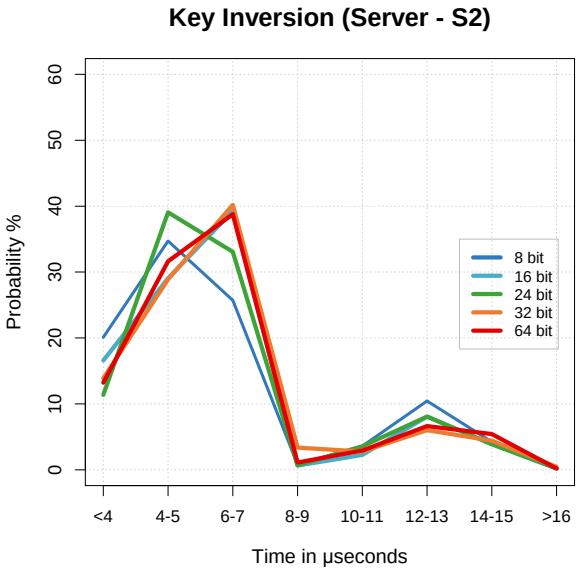


Figure 7.30: Probability densities function of scenario 2. The key inversion procedure required from 3 to 5 microseconds for every ID length.

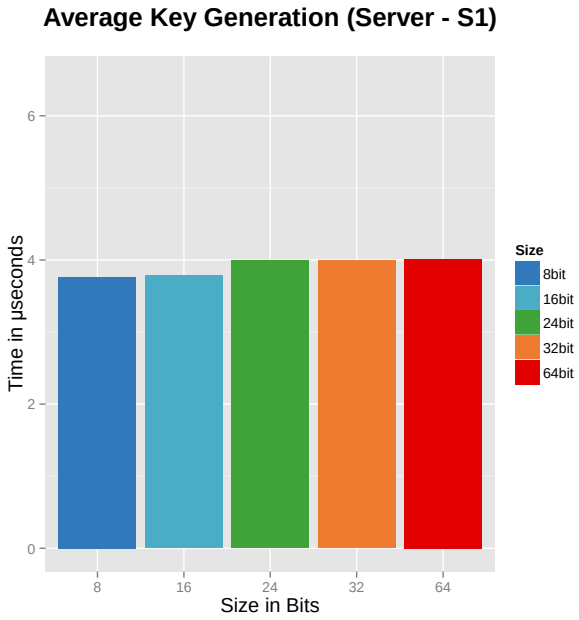


Figure 7.31: The average time required for the server to generate a random key. There is an increase of 6% from 8 to 64 bits.

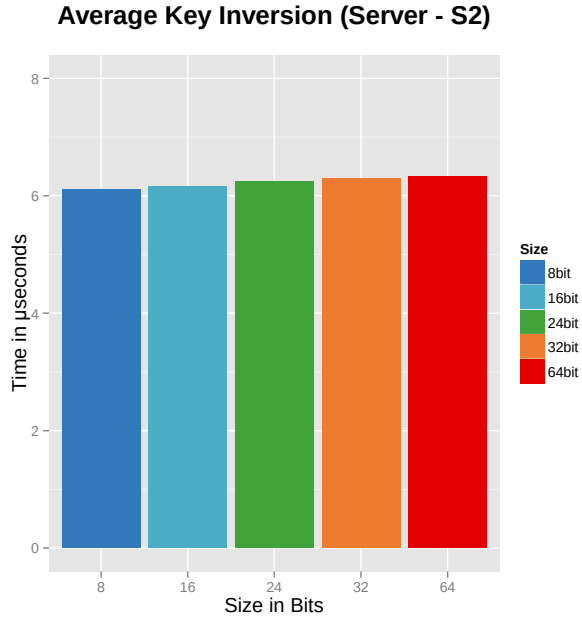


Figure 7.32: The average time required for the server to invert a key. There is an increase of 3.5% from 8 to 64 bits.

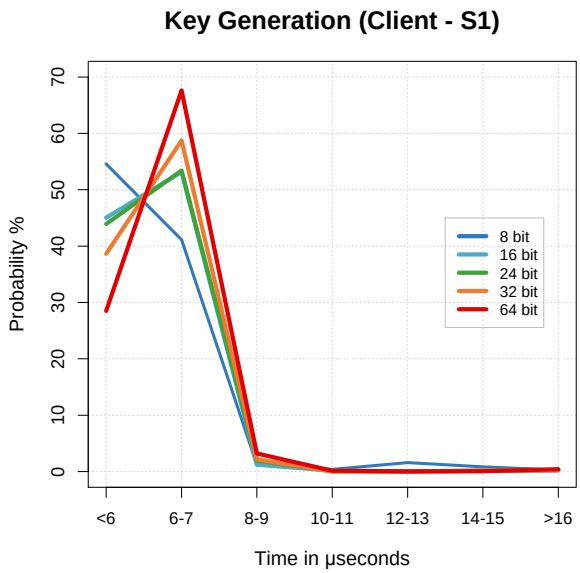


Figure 7.33: Probability densities function of scenario 1. The key generation procedure required 6 to 8 microseconds for every ID length.

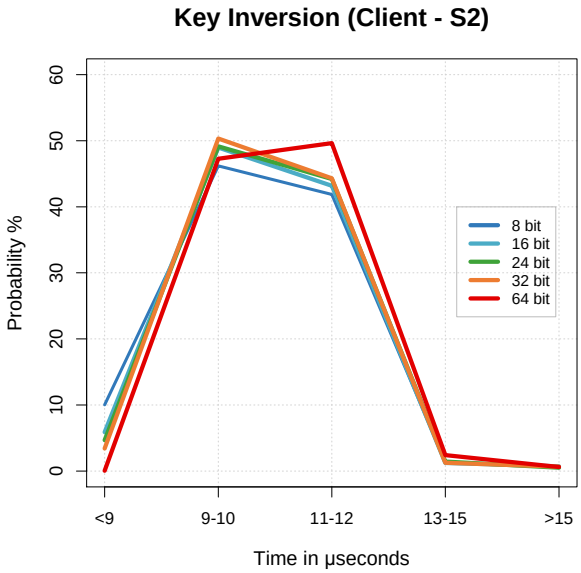


Figure 7.34: Probability densities function of scenario 2. The key inversion procedure required from 9 to 12 microseconds for every ID length tested.

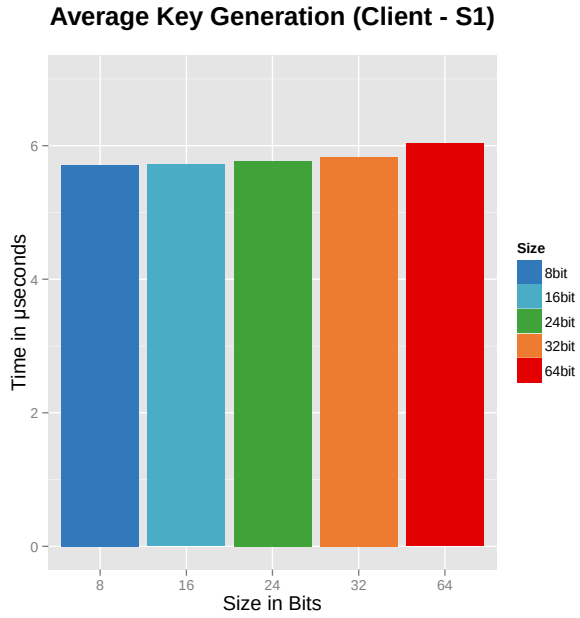


Figure 7.35: The average time required for the client to generate a random key. There is only an increase of 5.5% from 8 to 64 bits.

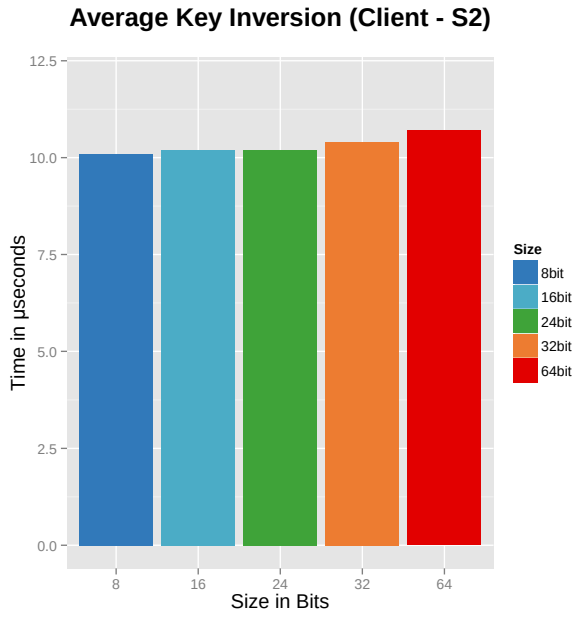


Figure 7.36: The average time required for the client to generate a random key. The delay is limited from 10 to 10.7 microseconds.

Regarding the client's response times, they are not significantly affected by the employment of the privacy service. Fig. 7.38 depicts that most messages had response times between 7 and 9 milliseconds. This is also the case when the proposed privacy service was disabled. In all configurations the client's average response time are not affected by the privacy service. There is only a deviation of 3% among the mean values in all measurements (Fig. 7.40).

7.4.5 A Comparative Study with Proposals from Other Researchers

The contribution of other research work in this field has been presented in Chapter 6.4. The schemes proposed by other researchers either require modification in the underlying infrastructure or the deployment of additional network entities [35]. The employment of a PKI is proposed in [32, 34] which involves among others digital certificate management, signing, revoking and updating procedures. On the other hand, symmetric encryption techniques introduce key related concerns (e.g. key exchange/agreement protocols, key storing).

One might assume that the employment of a mechanism such as S/MIME (see Section 6.4) could facilitate the protection of signaling's sensitive information but, on the other hand, the digital certificates involved may disclose user's identity. Alternatively, the AKA with IPSec and the SIP Digest with TLS can protect users' privacy since they provide confidentiality and integrity services to the communication. However, during the registration procedure, the user must provide his identity in clear text.

Table 7.14 summarizes the requirements and services offered by various mechanism proposed by other researchers including a comparison with the proposed identity protection mechanism. It should be stressed that the proposed mechanism protects the identities of both communicating member (caller and callee) when requested while it does not require:

(a) modification of the underlying infrastructure (inter-operable)

(b) key management scheme

(c) PKI infrastructure and finally

(d) any prior-knowledge between the communicating entities

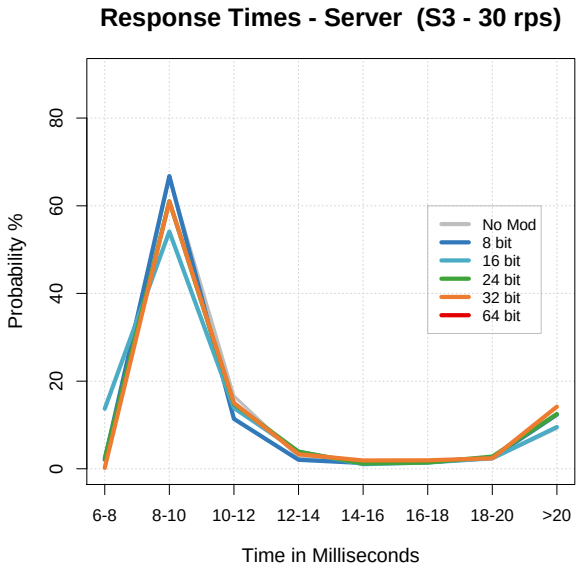


Figure 7.37: Probability densities function of scenario 3. The server's response times vary from 7 to 11 milliseconds for every ID length. No overhead is imposed on the server.

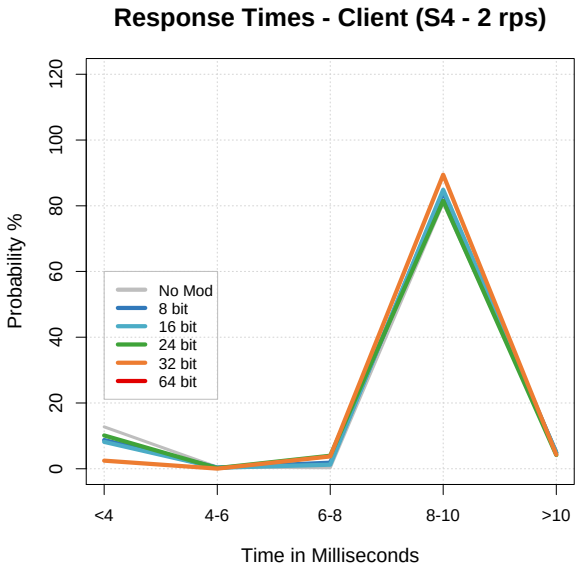


Figure 7.38: Probability densities function of scenario 4. The client's response times vary from 7 to 9 milliseconds for every ID length. No overhead is imposed on the client.

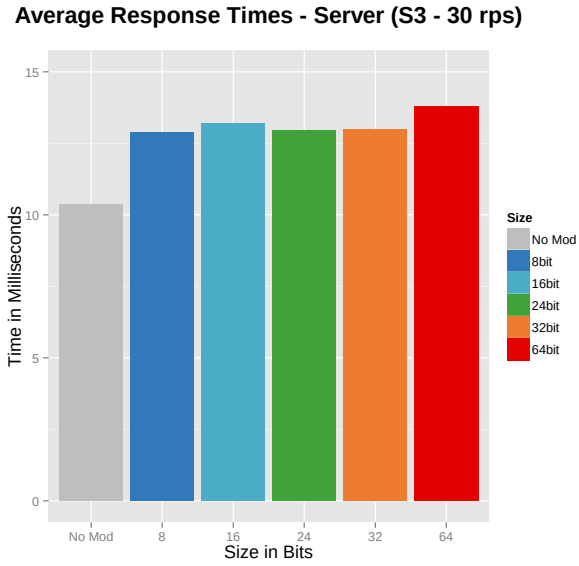


Figure 7.39: The server's average response times. A 32% increase to the response time has been measured due to the employment of the privacy module.

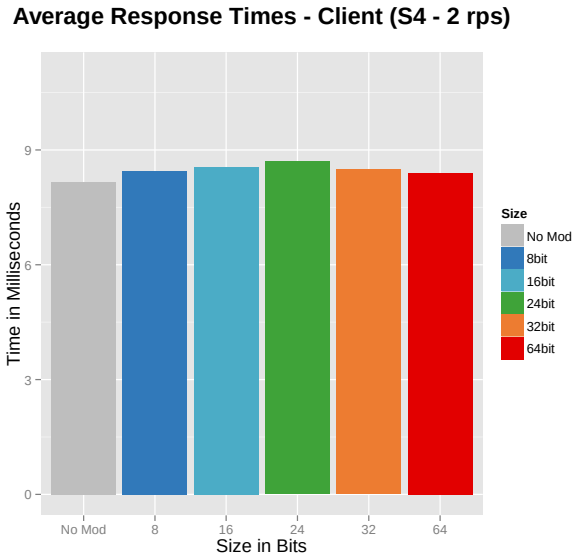


Figure 7.40: The client's average response times are similar even when the privacy service is turned off. There is only a 3% increase when the service is enabled.

Table 7.14: A Qualitative Comparison of Identity Protection Schemes

		Prop.	[3]	[31]	[100]	[35]	[32]	[18]	[33]	[34]	[112]
Requirements of the Mechanism	PKI	x	x	x	x	x	●	x	●	x	●
	Infrastructure Modification	x	x	x	x	●	x	x	x	○	●
	Key Exchange	x	x	x	x	x	x	x	x	○	●
	Key Management	x	x	x	x	x	●	x	●	●	●
	Prior Knowledge	x	x	x	x	x	x	●	x	○	x
	Hi-resource Enabled Device	x	x	x	x	x	x	●	x	x	x
Provisions of the Mechanism	Caller's ID Anonymity	●	○	○	○	●	○	●	●	●	x
	Callee's ID Anonymity	●	x	x	x	x	x	●	●	●	○
	First-hop ID Anonymity	●	x	x	x	x	●	○	●	●	x
	Last-hop ID Anonymity	●	●	●	●	●	●	●	●	●	x
	End-to-end ID Anonymity	●	x	x	x	x	●	●	●	●	x
	Intra-domain ID Anonymity	●	x	x	x	x	●	●	●	●	x
	Bid-down Attack Protection	●	x	x	●	●	●	●	●	●	x

●: Required/provided ○: Partially Required/provided x: Not Required/Provided

Chapter 8

Conclusions and Future Research Directions

Numerous malicious acts may threaten users' private information, system availability and the overall security level in IMS environments. A thorough review of related literature, 3GPP and SIP specifications, has demonstrated that attackers can exploit a variety of vulnerabilities originating from signaling protocol weaknesses or from the other protocols involved in the communication. Besides, the experimental testing of new and slightly modified old attacks, that has been carried out during this research, has proven that IMS deployments cannot effectively deal with the aforementioned threats.

Existing security frameworks do not offer satisfactory protection and moreover they do not take into account the important fact that the attacker may, in many cases, be an internal user that has a legitimate subscription to the service.

8.1 Contributions

The research work presented in this thesis has focused on enhancing the security and privacy protection levels in VoIP/IMS environments without introducing significant overheard to user's equipment and home network elements. The proposed protection mechanisms are lightweight, keeping to a minimum the delay introduced to network's response times and, consequently, supporting the weaker in terms of resources, devices. Additionally, the proposed framework does not require the deployment of supplementary infrastructures and can thus be utilized instantly by different operators.

The availability of services, in VoIP/IMS environments, comprised the first objective of this research. The development of a mechanism against various flooding attacks has been described. A session distance metric has been introduced for incomplete session detection. This metric enables the detection of single and distributed source denial of service attacks which are the main attacks against system availability.

A cross-layer intrusion detection and prevention mechanism has been also proposed by gathering information from three different layers of the protocol stack, the mechanism can detect the manipulated signaling messages and prevent the propagation of a possible attack. While the tampered messages can be detected and various attack cases that threaten messages authenticity can be prevented, the gathered data can be utilized for preventing flooding attacks. The deployment of Bloom filters in conjunction with a statistical model provides the means for spoofed single and distributed source denial of service attack prevention.

Another important contribution is the development of a mechanism that addresses the users' identity disclosure problems. Specifically, all the identities are transformed by utilizing commutative functions. This one-time identities replaces the user identity concealing that way the true identities from passive eavesdroppers. The key feature of this mechanism is that both user identities (caller and callee) can be concealed without the need of key related techniques (key establishment, agreement, storing etc.) or the deployment of additional infrastructure.

All the developed mechanisms have been implemented and evaluated through an IMS test-bed architecture. Numerous scenarios have been adopted for assessing their efficiency, resource consumption, effectiveness and accuracy. The experimental results demonstrate relatively low resource consumption and fast detection times, while the accuracy remains high. Concerning the privacy protection mechanism, the results highlight a negligible amount of overhead since it is executed only once per session and not for every single message.

8.2 Research Directions

The VoIP/IMS vulnerabilities, presented in this thesis, have different levels of exploitability thus influencing the probability of an attack to be successful. Specifically, this probability depends on various factors such as: (a) the number of attacks required in order a vulnerability to be exploited, (b) the layer of the protocol stack where the attack develops, (c) the number of existing attacking tools available. This probability degrades when protection frameworks are utilized by the network operator. It would be interesting to quantify that probability on a specific system, considering all the above parameters in conjunction with the employed security countermeasures. Such a quantification can be used as a security metric representing the system's protection level as well as its improvement when specific security solutions are adopted.

Since we cannot overlook the explosive entry of cloud systems and their numerous benefits against conventional and expensive deployments, the applicability of the proposed IMS security framework in cloud environments would be worth studying. Particularly, it would be interesting to consider the provisions of this security framework as a cloud hosted service to VoIP and IMS operators.

Bibliography

[1] J. Postel, "RFC 791: IP: Internet Protocol," 1981.

[2] 3GPP, "TS 23.228: IP Multimedia Subsystems (IMS)," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2011.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP: Session Initiation Protocol," 2002.

[4] N. Ehrlich, "The advanced mobile phone service," *Communications Magazine, IEEE*, vol. 17, no. 2, pp. 9–16, 1979.

[5] J. Cai and D. Goodman, "General packet radio service in GSM," *Communications Magazine, IEEE*, vol. 35, no. 10, pp. 122–131, 1997.

[6] A. Furuskar, S. Mazur, F. Muller, and H. Olofsson, "EDGE: Enhanced data rates for GSM and TDMA/136 evolution," *Personal Communications, IEEE*, vol. 6, no. 3, pp. 56–66, 1999.

[7] 3GPP2, "SS0086-B: IMS security framework," Tech. Rep., 2008.

[8] P. Samarati and S. de Vimercati, *Access Control: Policies, Models, and Mechanisms*. Springer Berlin / Heidelberg, 2001, vol. 2171, pp. 137–196.

[9] T. J. Walsh and D. R. Kuhn, "Challenges in securing voice over IP," *IEEE Security and Privacy*, vol. 3, no. 3, pp. 44–49, 2005.

[10] S. Bellovin, "Security problems in the TCP/IP protocol suite," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 2, p. 48, 1989.

[11] M. Tanase, "IP spoofing: an introduction," *Security Focus*, vol. 11, 2003.

[12] R. Wagner, "Address resolution protocol spoofing and man-in-the-middle attacks," 2001, the SANS Institute. [Online]. Available: http://rr.sans.org/threats/address.php

[13] D. Geneiatakis, A. Dagiouklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, and D. Sisalemm, "Survey of security vulnerabilities in session initiation protocol," *IEEE Communications Surveys & Tutorials*, vol. 8, pp. 68–81, 2006.

[14] Y. Park and T. Park, "A survey of security threats on 4G networks," in *IEEE Globecom Workshops*, Washington, DC, 2007, pp. 1–6.

[15] A. Keromytis, "A comprehensive survey of voice over IP security research," *Communications Surveys & Tutorials, IEEE*, no. 99, pp. 1–24, 2011.

[16] S. Kent and R. Atkinson, "RFC 2401: Security Architecture for the Internet Protocol," 1998.

[17] T. Dierks and C. Allen, "RFC 2246: The TLS Protocol Version 1.0," 1999.

[18] 3GPP, "TS 33.203: 3G security; Access security for IP-based services (Release 10)," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.

[19] D. Sisalem, J. Kuthan, U. Abend, J. Floroiu, and H. Schulzrinne, *SIP Security*. Wiley, 2009.

[20] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "RFC 2617: HTTP authentication: basic and digest access authentication," 1999.

[21] N. Vrakas, D. Geneiatakis, and C. Lambrinoudakis, "Evaluating the security and privacy protection level of IP multimedia subsystem environments," *Communications Surveys & Tutorials, IEEE*, vol. PP, no. 99, pp. 1–17, 2012.

[22] H. Abdelnur, T. Avanesov, and M. Rusinowitch, "Abusing SIP authentication," *Journal of Information Assurance and Security*, vol. 4, no. 4, 2009.

[23] Y. Wu, V. Apte, S. Bagchi, S. Garg, and N. Singh, "Intrusion detection in voice over IP environments," *International Journal of Information Security*, vol. 8, no. 3, pp. 153–172, 2009.

[24] Y. Wu, S. Bagchi, S. Garg, N. Singh, and T. Tsai, "Scidive: A stateful and cross protocol intrusion detection architecture for voice over IP environments," in *Proceedgins of the 2004 International Conference on Dependable Systems and Networks (DSN 2004)*, Firenze, Italy, 2004, pp. 433–442.

[25] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, "Detecting VoIP floods using the Hellinger distance," *IEEE Transactions on Parallel and Distributed Systems*, pp. 794–805, 2008.

[26] X. Wan, Z. Li, and Z. Fan, "A SIP DoS flooding attack defense mechanism based on priority class queue," in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, Beijing, China, 2010, pp. 428–431.

[27] D. Geneiatakis, N. Vrakas, and C. Lambrinoudakis, "Utilizing Bloom filters for detecting flooding attacks against SIP based services," *Computers & Security*, vol. 28, no. 7, pp. 578–591, 2009.

[28] R. Srinivasan, V. Vaidehi, K. Harish, K. Lakshmi Narasimhan, S. Lokeshwer Babu, and V. Srikanth, "Authentication of signaling in VoIP applications," in *Asia-Pacific Conference on Communications*, Perth, WA, 2005, pp. 530–533.

[29] P. G. Argyroudis, R. Verma, H. Tewari, and D. O'Mahony, "Performance analysis of cryptographic protocols on handheld devices," in *Third IEEE International Symposium on Network Computing and Applications (NCA 2004)*, 2004, pp. 169–174.

[30] C. Shen, E. Nahum, H. Schulzrinne, and C. Wright, "The impact of TLS on SIP server performance," in *IPTComm 2010: 4th Conference on Principles, Systems and Applications of IP Telecommunications Principles, Systems and Applications of IP Telecommunications*, Munich, Germany, 2010, pp. 59–70.

[31] J. Peterson, "RFC 3323: A Privacy Mechanism for the Session Initiation Protocol (SIP)," 2002.

[32] B. Ramsdell, "RFC 2633: S/MIME version 3 message specification," 1999.

[33] G. Karopoulos, G. Kambourakis, and S. Gritzalis, "Caller identity privacy in SIP heterogeneous realms: A practical solution," in *IEEE Symposium on Computers and Communications, ISCC*, Marrakech, Morocco, 2008, pp. 37–43.

[34] G. Karopoulos, G. Kambourakis, S. Gritzalis, and E. Konstantinou, "A framework for identity privacy in SIP," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 16–28, 2010.

[35] M. Castleman, "SIPANON: A SIP Anonymizer," *Course Report (CS W3998)*, 2001.

[36] N. Vrakas, D. Geneiatakis, and C. Lambrinoudakis, "Is IP multimedia subsystem affected by "malformed message" attacks? an evaluation of openims," in *SECRYPT 2011, the International Joint Conference on e-Business and Telecommunications*, Seville, Spain, 18-21 July, 2011.

[37] N. Vrakas, D. Geneiatakis, and C. Lambrinoudakis, "A call conference room interception attack and its detection," in *7th International Conference on Trust, Privacy & Security in Digital Business*, Bilbao, Spain, 2010.

[38] N. Vrakas and C. Lambrinoudakis, "A cross layer spoofing detection mechanism for multimedia communication services," *International Journal of Information Technologies and Systems Approach*, vol. 4, no. 2, pp. 32–47, 2011.

[39] N. Vrakas and C. Lambrinoudakis, "An intrusion detection and prevention system for IMS and VoIP services," *International Journal of Information Security*, 2013.

[40] ITU-T, "Packet-based multimedia communication systems: Recommendation H. 323," 1999.

[41] F. Andreasen and B. Foster, "RFC 3435: Media gateway control protocol (MGCP) version 1.0," 2003.

[42] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 3550: A transport protocol for real-time applications (RTP)," 1996.

[43] M. Handley, C. Perkins, and V. Jacobson, "RFC 4566: SDP: session description protocol," 2006.

[44] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "RFC 3711: The secure real-time transport protocol (SRTP)," 2004.

[45] P. Mockapetris, "RFC 1035: Domain names-implementation and specification," 1987.

[46] R. Droms, "RFC 2131: Dynamic host configuration protocol," 1997.

[47] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "RFC 3588: Diameter base protocol," 2003.

[48] H. Zimmermann, "OSI reference model: The ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

[49] V. Cerf and R. Icahn, "A protocol for packet network intercommunication," *IEEE trans. on Commun.*, vol. COM-22, pp. 637–648, 1974.

[50] A. Tanenbaum, *Computer networks*. Prentice-Hall, 1996.

[51] S. Deering and R. Hinden, "RFC 2460: Internet protocol, version 6," 1998.

[52] J. Postel, "RFC 793: Transmission control protocol (TCP)," 1980.

[53] J. Postel, "RFC 768: User datagram protocol (UDP)," 1980.

[54] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "RFC 2960: Stream control transmission protocol (SCTP)," 2000.

[55] J. Postel, "RFC 821: Simple mail transfer protocol (SMTP)," 1982.

[56] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "RFC 2616: Hypertext transfer protocol-http/1.1," 1999.

[57] R. Sparks, "RFC 3515: The session initiation protocol (SIP) refer method," 2003.

[58] A. B. Roach, "RFC 3265: SIP-specific event notification," 2002.

[59] A. Roach, "RFC 6665: SIP-specific event notification," 2012.

[60] M. Garcia-Martin, E. Henrikson, and D. Mills, "RFC 3455: Private header extensions to the session initiation protocol for the 3rd-generation partnership project," 2003.

[61] P. Faltstrom and M. Mealling, "RFC 3761:The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)," Tech. Rep., 2004.

[62] 3GPP, "TS 23.002: Network Architecture," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.

[63] H. Schulzrinne, "RFC 3966: The tel URI for telephone numbers," 2004.

[64] T. Taylor, "Megaco/H. 248: a new standard for media gateway control," *Communications Magazine, IEEE*, vol. 38, no. 10, pp. 124–132, 2000.

[65] 3GPP, "TS 23.218: Session handling - IM call model," Third Generation Partnership Project-Technical Specification Group Core Network and Terminals, 2012.

[66] 3GPP, "TS 29.163: Interworking between the IP Multimedia (IM) Core Network (CN) Subsystem and Circuit Switched (CS) Networks," Third Generation Partnership Project, 2009.

[67] 3GPP, "TS 23.078: Customized applications for mobile network enhanced logic (CAMEL)," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.

[68] B. Aboba and M. Beadles, "RFC 2486: The network access identifier," 2009.

[69] 3GPP, "TS 23.003: Numbering, addressing and identification," Third Generation Partnership Project, Technical Specification Group Core Network and Terminals, 2012.

[70] J. Rosenberg, "RFC 5627: Obtaining and using globally routable user agent uris (GRUUs) in the session initiation protocol (SIP)," 2009.

[71] G. Camarillo and M. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. John Wiley & Sons, 2008.

[72] 3GPP, "TS 29.228: Ip multimedia (IM) subsystem Cx and Dx interfaces;signalling flows and message contents," Third Generation Partnership Project; Technical Specification Group Core Network and Terminals, 2012.

[73] P. Faltstrom, "RFC 2916: E. 164 number and DNS," 2000.

[74] M. Garcia-Martin, M. Belinchon, M. Pallares-Lopez, C. Canales-Valenzuela, and K. Tammi, "RFC 4740: Diameter session initiation protocol (SIP) application," 2006.

[75] 3GPP, "TS 24.228: Signalling flows for the IP multimedia call control based on session initiation protocol (SIP) and session description protocol (SDP)," Third Generation Partnership Project, Technical Specification Group Core Network and Terminals, 2006.

[76] W. Marshall, "RFC 3313: Private session initiation protocol (SIP) extensions for media authorization," 2003.

[77] M. Poikselka and G. Mayer, *The IMS: IP multimedia concepts and services*. Wiley-Blackwell, 2006.

[78] 3GPP, "TS 24.147: Conferencing using the IP multimedia (IM) core network (CN) subsystem," Third Generation Partnership Project, Technical Specification Group Core Network and Terminals, 2009.

[79] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka, "RFC 3329: Security mechanism agreement for the session initiation protocol (SIP)," 2003.

[80] 3GPP, "TR 33.978 Security aspects of early IP Multimedia Subsystem (IMS)," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2008.

[81] ETSI, "TS 187.003: Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): Security Architecture," 2008.

[82] 3GPP, "TS 33.102 universal mobile telecommunication system (UMTS); 3G security; security architecture," Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.

[83] D. Geneiatakis, G. Kambourakis, T. Dagiuklas, C. Lambrinoudakis, and S. Gritzalis, "SIP security mechanisms: A state-of-the-art review," in *Proceedings of Fifth International Network Conference*, Samos, Greece, 2005, pp. 147–155.

[84] F. Lau, S. Rubin, M. Smith, and L. Trajkovic, "Distributed denial of service attacks," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, Nashville, USA, 2000, pp. 2275–2280.

[85] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, pp. 38–47, 2001.

[86] B. Harris and R. Hunt, "TCP/IP security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885–897, 1999.

[87] S. Limited, "Skype." [Online]. Available: http://www.skype.com

[88] P. D'haeseleer, P. Forrest, and S. Helman, "An immunological approach to change detection: Algorithms, alalysis and implication," in *Proceedings of IEEE Symposioum on Security and Privacy*, Oakland, CA, 1996, pp. 110–119.

[89] A. Bremler-Barr, R. Halachmi-Bekel, and J. Kangasharju, "Unregister attacks in SIP," in *2nd Workshop on Secure Network Protocols, NPSec*, Santa Barbara, CA, 2006, pp. 32–37.

[90] C. Anley, "Advanced SQL Injection In SQL Server Applications," 2002. [Online]. Available: http://www.nextgenss.com/

[91] N. Asokan, V. Niemi, and K. Nyberg, "Man-in-the-middle in tunnelled authentication protocols," *Lecture Notes in Computer Science*, vol. 3364, p. 28, 2005.

[92] H. Xia and J. Brustoloni, "Hardening web browsers against man-in-the-middle and eavesdropping attacks," in *Proceedings of the 14th international conference on World Wide Web*, Chiba, Japan, 2005, pp. 498–498.

[93] A. Klein, "BIND 9 DNS cache poisoning," 2007. [Online]. Available: http://www.trusteer.com/docs/bind9dns.html

[94] 3GPP, "TS 24.229: IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)," 2009.

[95] R. Zhang, X. Wang, R. Farley, X. Yang, and X. Jiang, "On the feasibility of launching the man-in-the-middle attacks on VoIP from remote attackers." Sydney, Australia: ACM, March 2009, pp. 61–69.

[96] C. Wieser, M. Laakso, and H. Schulzrinne, "SIP robustness testing for large-scale use," in *First International Workshop on Software Quality (SOQUA)*, Erfurt, Germany, 2004, pp. 165–178.

[97] D. Wang and C. Liu, "Model-based vulnerability analysis of IMS network," *Journal of Networks, Academy Publisher*, vol. 4, 2009.

[98] M. Sher, S. Wu, and T. Magedanz, "Security Threats and Solutions for Application Server of IP Multimedia Subsystem (IMS-AS)," in *IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation*, Tubingen, Germany, 2006.

[99] G. Combs, "Wireshark." [Online]. Available: http://www.wireshark.org/

[100] C. Jennings and J. Peterson, "RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks," 2002.

[101] F. S. Park, D. Patnaik, C. Amrutkar, and M. T. Hunter, "A security evaluation of IMS deployments," in *the 2nd International Conference on Internet Multimedia Services Architecture and Applications, IMSAA*, Bangalore, India, 2008, pp. 1–6.

[102] D. Geneiatakis, G. Kambourakis, T. Dagiuklas, C. Lambrinoudakis, and S. Gritzalis, "SIP message tampering: The SQL code injection attack," in *Proceedings of 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005)*, Split, Croatia, 2005.

[103] H. Takahara and M. Nakamura, "Enhancement of SIP Signaling for Integrity Verification," in *10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, Seoul, Korea, 2010, pp. 289–292.

[104] G. Ormazabal, S. Nagpal, E. Yardeni, and H. Schulzrinne, "Secure SIP: A Scalable Prevention Mechanism for DoS Attacks on SIP Based VoIP Systems," in *Proceedings of the 2nd International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, Heidelberg, Germany, 2008, pp. 107–132.

[105] E. Chen and M. Itoh, "A whitelist approach to protect SIP servers from flooding attacks," in *IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, Vancouver, BC, 2010, pp. 1–6.

[106] V. Balasubramaniyan, A. Poonawalla, M. Ahamad, M. Hunter, and P. Traynor, "PinDr0p: using single-ended audio features to determine call provenance," in *Proceedings of the*

ACM Conference on Computer and Communications Security (CCS), Chicago, IL, USA, 2010, pp. 109–120.

[107] M. Nassar and S. Niccolini, "Holistic voip intrusion detection and prevention system," in *Principles, Systems andApplications of IP Telecommunications (IPTComm 2007).*, New York, USA, 2007, pp. 1–9.

[108] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, T. Dagiuklas, and S. Gritzalis, "A framework for protecting a SIP-based infrastructure against malformed message attacks," *Computer Networks*, vol. 51, no. 10, pp. 2580–2593, 2007.

[109] K. Rieck, S. Wahl, P. Laskov, P. Domschitz, and K. Muller, "A self-learning system for detection of anomalous SIP messages," in *Proceedings of the 2nd International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, Heidelberg, Germany, 2008, pp. 90–106.

[110] J. Daemen and V. Rijmen, *The design of Rijndael: AES - The advanced encryption standard*. Springer-Verlag New York Inc, 2002.

[111] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. Mickunas, and S. Yi, "Routing through the mist: Privacy preserving communication in ubiquitous computing environments," in *22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 74–83.

[112] L. Kazatzopoulos, C. Delakouridis, and G. Marias, "Providing anonymity services in SIP," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, Cannes, France, 2008, pp. 1–6.

[113] P. Bose, H. Guo, E. Kranakis, A. Maheshwari, P. Morin, J. Morrison, M. Smid, and Y. Tang, "On the false-positive rate of bloom filters," *Inf. Process. Lett.*, vol. 108, no. 4, pp. 210–213, 2008.

[114] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2004.

[115] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[116] J. Udhayan and T. Hamsapriya, "Statistical segregation method to minimize the false detections during ddos attacks," *International Journal of Network Security*, vol. 13, no. 3, pp. 152–160, 2011.

[117] F. Fokus, "OpenIMS." [Online]. Available: http://www.openimscore.org

[118] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[119] "People search," 2011. [Online]. Available: www.pipl.com

[120] A. Shamir, "On the power of commutativity in cryptography," *Automata, Languages and Programming*, pp. 582–595, 1980.

[121] K. McCurley, "The discrete logarithm problem," in *Proc. of Symp. in Applied Math*, vol. 42, 1990, pp. 49–74.

[122] PJSIP, "Stack and Media Stack for Presence, Instant Messaging and Multimedia Communication," 2008. [Online]. Available: http://www.pjsip.org