Security Enhancement for SIP in Ad Hoc Networks

by

Maram Alshingiti

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of

the requirements for the degree of

MASTER OF COMPUTER SCIENCE

School of Computer Science

at

CARLETON UNIVERSITY

Ottawa, Ontario

May 17, 2012

Your file Votre référence
ISBN: 978-0-494-93497-5

Our file Notre référence
ISBN: 978-0-494-93497-5

Abstract

Session Initiation Protocol (SIP) is an application-layer signaling protocol that initiates, modifies, and terminates IP-based multimedia sessions. Implementing SIP in ad hoc networks has been a topic of study for the past decade, and several proposals are available in the research domain. However, security aspects are not addressed in most of these proposals. In this thesis, SIP is extended to enhance its security in ad hoc networks. This is done by combining Cryptographically Generated Addresses (CGA) with the social network paradigm to provide authentication and message integrity. Simulations are conducted to evaluate the performance of the proposed security extension. These simulations demonstrate that the new extension reduces traffic overhead resulting from the registration process but increases the traffic caused by call establishment and termination. Furthermore, the time required to answer a call, i.e. the call setup delay, increases to an acceptable level of 300 milliseconds.

Acknowledgements

Foremost, I would like to express my heart-felt gratitude to my husband, Mahmoud, for his encouragement and persistent support, and for standing by my side throughout this journey. No words can express my love and appreciation to him.

My sincere thanks go to my supervisor Prof.Michel Barbeau for his guidance, words of encouragement, valuable comments, and support during these last few years.

I would also like to thank the Minstry of Higher Education represented by Saudi Arabian Cultural Bureau in Canada for their financial support.

Contents

Abstract iii

Acknowledgements v

Contents ix

List of Figures xii

List of Acronyms xiii

1 Introduction 1

1.1 Problem Statement 3

1.1.1 Attack Model 4

1.2 Assumptions 5

1.3 Contribution 6

1.4 Thesis Outline 7

2 Background 9

2.1 Overview of SIP 9

2.1.1 SIP Addressing 10

2.1.2 SIP Messages 10
2.1.3 SIP Architecture 12
2.1.4 SIP Operations 13
2.2 Overview of Ad hoc Networks 15
2.3 Overview of CGA 19
2.3.1 The CGA Data Structure 20
2.3.2 CGA Generation 22
2.3.3 The CGA Address Verification 25
2.3.4 Security Consideration of CGA 26

3 SIP in Ad Hoc Networks **29**
3.1 Using Service Location Protocol 30
3.2 Broadcasting the Binding Information 31
3.3 The Clustering Approach 32
3.4 SIP in Integrated MANET 33
3.5 SIP in Peer to Peer Networks 34

4 Related Work **35**
4.1 SIP Security Satndards 35
4.2 SIP Security in Ad hoc Networks 36
4.3 Authentication in SIP 37

5 Security Extension of SIP in Ad Hoc Networks **41**
5.1 Contribution Overview 41
5.1.1 Possible Scenario 45
5.1.2 Tools 46
5.1.3 Security of SIP Messages 48

5.2 SIP Protocol Architecture 50
5.2.1 Service Interface 52
5.2.2 Peer Interface 55

6 Evaluation **65**
6.1 Discussion 65
6.1.1 Memory Complexity 66
6.1.2 Friend Lookup Complexity 67
6.1.3 Registration Complexity 67
6.2 Simulation 68
6.2.1 Implementation 68
6.2.2 Simulation Setup 69
6.3 Performance Metrics 70
6.4 Simulation Results 71

7 Conclusion and Future Work **79**
7.1 Conclusion 79
7.2 Future Work 80

List of Figures

2.1 Basic parts of a SIP message 11
2.2 An example of SIP message 12
2.3 SIP components 13
2.4 Registration process 14
2.5 Call establishment in SIP 15
2.6 IP address 19
2.7 CGA generation (simplified) 20
2.8 CGADSP 21
2.9 The extension field of the CGADSP 22
2.10 The generation of Hash1 and Hash2 23
2.11 CGA generation algorithm 24
2.12 CGA verification algorithm 27

5.1 Friendship recommendation 44
5.2 Alice joining 44
5.3 Alice receives an INVITE message from non-friend Tom 47
5.4 The new CGA data structure parameter 48
5.5 Signing a SIP message 49
5.6 Verification of a SIP message 51

5.7 The process of adding a friend 54
5.8 Friend lookup request "simplified" 58
5.9 Response to Friend lookup request (simplified) 60
5.10 Flow chart for receiving a SIP INVITE message 61

6.1 Overhead per call 72
6.2 Total overhead 73
6.3 Registration overhead 74
6.4 Friend look-up overhead 74
6.5 Friend lookup overhead per call 76
6.6 Call setup delay 77
6.7 Percentage of unauthenticated calls 78

List of Acronyms

3GPP	3rd Generation Partnership Project
AODV	Ad hoc On-demand Distance Vector
AOR	Address of Record
CA	Certificate Authority
CGA	Cryptographically Generated Addresses
CGADSP	Cryptographically Generated Addresses Data Structure Parameter
DAD	Duplicate Address Detection
DHT	Dynamic Host Table
DNS	Domain Name Service
dSIP	decentralized SIP
DSR	Dynamic Source Routing
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
IBC	Identity Based Cryptosystem
IETF	Internet Engineering Task Force

IP	Internet Protocol
ITU	International Telecommunication Union
MANET	Mobile Ad hoc Network
MPR	Multi Point Relay
NS2	Network Simulator-2
OLSR	Optimized Link State Routing protocol
P2P	Peer to Peer
PKI	Public Key Infrastructure
PSTN	Public Switched Telephone Network
RREP	Route Reply
RREQ	Route Request
RSADSA	RSA Digital Signature Algorithm
RTP	Real-time Transport Protocol
SDR	Software Defined Radio
SAKA	Secure Authentication and Key Agreement
SHA-1	Secure Hash Algorithm-1
SIP	Session Initiation Protocol
SIPRREP	SIP Route Reply
SIPRREQ	SIP Route Request
SLP	Service Location Protocol
S/MIME	Secure/Multipurpose Internet Mail Extension

SRTP	Secure Real-time Transport Protocol
TA	Trusted Authority
TC	Topology Control
TLS	Transport Layer Security
TTP	Trusted Third Party
UAC	User Agent Client
UAS	User Agent Server
URI	Uniform Resource Identifier
VOIP	Voice Over Internet Protocol
ZRP	Zone Routing Protocol

Chapter 1

Introduction

Voice over IP (VOIP) services are widely used today due to their low cost, availability, and variety of services provided by applications. For example, besides voice sessions, VOIP also supports instant messaging, notifications, chat, and video conferencing. Several protocols have been proposed to carry multimedia service via the Internet such as Skype [44], H.323 [45], and Session Initiation Protocol (SIP) [41]. While Skype is a proprietary protocol, H.323 is recommended by the Telecommunication Standardization Sector of the International Telecommunication Union (ITU) and SIP is standardized by the Internet Engineering Task Force (IETF) and has been adopted by the mobile phone standardization body, the 3rd Generation Partnership Project (3GPP). Although H.323 was proposed earlier than SIP, SIP is gaining more popularity for several reasons. While H.323 is a binary encoded protocol, which is beneficial in terms of performance, SIP is a text based protocol, which makes it easier to debug and extend. In fact, SIP has been explicitly designed with the possibility to be extended and enhanced to support new

features. However, both SIP and H.323 rely on a client-server model, which is centralized and not applicable in ad hoc networks.

Ad hoc networks are constructed from autonomous devices that can communicate with each other via wireless links without relying on any infrastructure system. The lack of infrastructure makes the application of SIP in an ad hoc network impossible without modifying its structure. However, several research groups have attempted to enable SIP in ad hoc networks. The main aim of these attempts was to transform SIP from a naturally centralized protocol to a distributed one. More details about these efforts are found in Chapter 3. Furthermore, the distributed nature of ad hoc networks is an obstacle to the security countermeasures of SIP, since many of them rely on centralized entities. In particular, the wireless nature of ad hoc networks presents critical security risks. For instance, Barbeau notes that eavesdropping and intercepting of wireless traffic can be carried out without any technical difficulty [12]. He points out that this attack can be carried out in three different ways that involve hardware and/or software. First, a scanning software that has monitoring capabilities,such as Wireshark [7], Aircrackng [1], iStumbler [2], MacStumbler [5], KISMAC [3], and Kismet [4], can be used to intercept wireless data packets. Second, an attacker can write his/her own software using an application programming interface such as Linux Packet Socket that allows capturing of data frames. Third, a technology called Software Defined Radio (SDR)could be used to intercept data packets at the signal level.

Some work has already been done to address the security flaws of SIP, and the security of ad hoc networks has been studied extensively. However, to the best of our knowledge, examining the combination of SIP protocol

security and ad hoc network security has not yet been addressed except for one work by Leggio et. al. [30, 32]. Indeed, the efforts that have been made to enhance the security of SIP or the security of ad hoc networks don't suit SIP within ad hoc networks for two main reasons. First, the security of SIP requires centralized entities. For example, a centralized server is required to ensure the authentication of the user agents. Also, a centralized certificate authority is required to manage the key establishment and generation of digital certificates. All these solutions would not work in an ad hoc network since it is decentralized and requires distributed solutions. Second, the security enhancements of ad hoc networks don't take the SIP protocol into consideration, because they don't ensure secure exchange of SIP messages nor do they authenticate the SIP user agents. Hence, security mechanisms that are designed in a distributed way and fulfill the security requirements of SIP protocol are crucial to the possibility of enhancing the security of SIP in ad hoc networks.

1.1 Problem Statement

With the current technology, both hardware and software, intercepting wireless traffic can be carried out without any technical difficulty. SIP messages are text based and are not protected by any means in ad hoc networks. Therefore, it is possible to intercept and alter SIP messages. The main danger to SIP users comes from the combination of the attacker's ability to spoof the legitimate user's identity and the lack of a strong authentication mechanism. This means that attackers have the ability to launch an imper-

sonation attack, so a strong authentication mechanism is required to prevent these spoofing attacks, and message integrity is crucial to resist any message altering attempts.

1.1.1 Attack Model

Considering SIP used in ad hoc networks, thereare no security countermeasures to provide encryption or any sort of protection to the messages. Several types of attacks result from the interception and modification of these SIP messages. However, most attacks result from the attacker's ability to spoof a user's identity. The following are several types of attacks that could occur:

1. An attacker can spoof a user identity and send a SIP REGISTER message with an Expire header field set to zero. This message is normally sent when the device is shutdown and no more calls can be sent. Hence, the legitimate user would not be able to make or receive calls. This attack is called a de-registration attack.

2. An ongoing session can be terminated if an attacker spoofs the user identity of a caller or callee and constructs a SIP BYE or CANCEL message. This attack is known as a call tear down attack.

3. An attacker can resend a SIP INVITE message to an already maintained session to modify some parameters of the session. For example, the "Contact" header fields can be modified to cause a Denial of Service attack to the end party and redirect the call to the attacker.

4. A call can be hijacked and redirected to the attacker's device if the attacker sends a spoofed SIP REGISTER message that contains the attacker's binding information instead of that of the legitimate user.

5. An SQL injection attack, which harms the data base, can be carried out if an attacker injects malicious code into the SIP message.To illustrate, a malicious code can be injected as the content of one or more SIP header fields.

1.2 Assumptions

It is important to look at VOIP in ad hoc networks from a realistic point of view. Consequently, some considerations need to be taken into account as follows:

1. A VOIP session in an ad hoc network can be held at a conference, a university campus, a company campus, a small town, or within a neighbourhood.

2. In most VOIP scenarios it is most likely that the participants know each other, unless it is an emergency call where previous knowledge of the other participants is not important. In short, one participants will make a call to another partcipant that it knows or wants to know.

3. The distributed nature of the ad hoc network should be maintained when providing the proper security solutions. No centralized entities, such as certificate authorities, centralized servers, or centralized proxies, should be involved in the security countermeasures of SIP in ad hoc networks.

1.3 Contribution

In this work spoofing attacks and subsequent attacks in ad hoc network using the SIP protocol are prevented by providing both authentication and message integrity. A Cryptographically Generated Address (CGA) is used as a form of a self-signed certificate, without relying on any certificate authority since it binds the public key of the user to his/her IP address. One benefit from using CGA is its ability to prevent identity spoofing through all future sessions; however, validation of the identities in the initial phase is not possible. To overcome this limitation, the social network paradigm is used to validate users identities. This means that friends must exchange their CGA addresses in some offline way. Each user will maintain a list of the CGA addresses of his/her friends. The stored CGAs are used later to validate the participants' identities. Message integrity is achieved by signing the message using the private key of the originator.

The integration of social networking into SIP is a new extension to the protocol. Consequently, the following services must be introduced, such as: add, recommend, delete, and block/unblock friends. To apply the new services, some modifications have been introduced to the protocol. These modifications include modifying some header fields and the way the SIP messages are processed.

Simulations have been conducted to evaluate the performance of the new security extension. They show that without using the extension, SIP REGISTER messages cause increased network traffic because when they broadcast through the network, all nodes in the network respond to them. However, when using the security extension, the traffic caused by the registration pro-

cess is reduced because only the user's friends respond to the broadcasted REGISTER message. On the other hand, traffic overhead for establishing and terminating calls inccreases with the proposed extension. Simulation shows a call setup delay of less than 300 milliseconds which is acceptable as indicated by an Internet draft of IETF [42].

1.4 Thesis Outline

Some background about SIP, ad hoc networks, and CGA is presented in Chapter 2. Chapter 3 presents the state of art in enabling SIP in ad hoc networks. Chapter 4 shows related work to the security of SIP. A detailed discussion about the proposed security extension is presented in Chapter 5. Chapter 6 provides a discussion about the proposed extension, performance metrics, and simulation setup. Results from the performance evaluation is also presented. Chapter 7 concludes the thesis and highlights potential future work.

Chapter 2

Background

In this chapter, background information about SIP, ad hoc networks, and CGA is provided.

2.1 Overview of SIP

SIP is an application layer protocol. It is a signalling protocol that handles multimedia session establishment. The actual multimedia in these sessions is transferred using the Real-time Transport Protocol (RTP).

SIP uses a set of entities to construct an overlay network on top of an IP network . Based on the session type, there may be many SIP entities involved, such as in Public Switched Telephone Network (PSTN) gateways, voicemail servers, and conferencing servers. However, in this section, only the basic and general components are described.

2.1.1 SIP Addressing

SIP users are uniquely identified using SIP addresses. An SIP address is a Uniform Resource Identifier (URI) that is also known as the Address of Record (AOR). It has the same format as an email address and consists of three parts: the keyword "sip", the user name, and the domain name. For example, sip:alice@example.com is a valid AOR. The URI addresscan also represent a PSTN telephone number, by using the "tel" keyword. An example would be tel:+1-613-274-7470. In this case, the SIP proxy translates the telephone number into a DNS name, and gateways are then used for interconnection with the PSTN.

2.1.2 SIP Messages

SIP messages are text based and can be one of two types: a request or a response. A request message is directed from a client to a server, whereas a response is from a server to a client. SIP specifications define six requests and many responses. The requests are: REGISTER, INVITE, ACK, CANCEL. BYE, and OPTIONS. A SIP response, such as 200 OK, differs from a request by having a numerical status code in addition to its textual descriptive phrase. No matter the type, each SIP message consists of a start line, one or more header fields, an empty line that indicates the end of the header fields, and an optional message body all are illustrated in Figure 2.1.

SIP specifications specify more than 40 header fields, which can grow in number as SIP extensions are introduded. However, a subset of six header fields are mandatory for all SIP messages. All are: To, From, CSeq, Call-

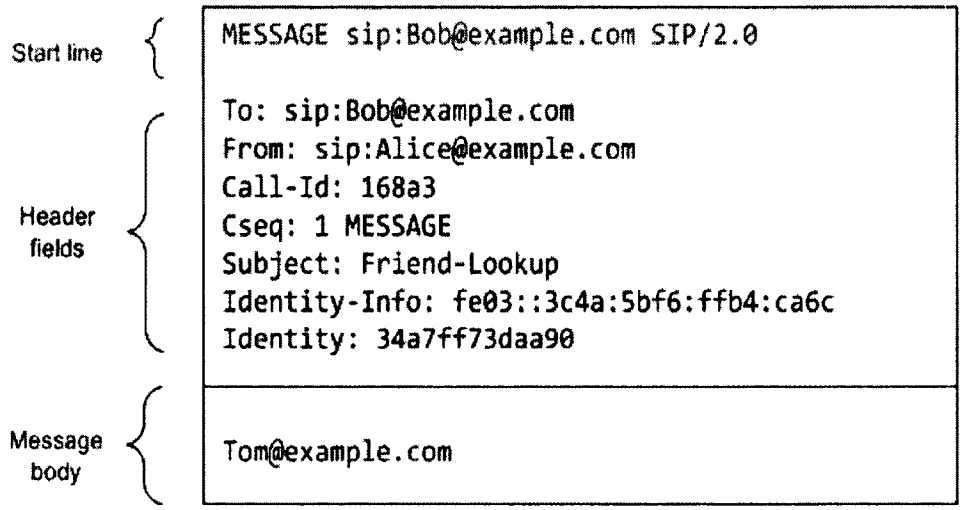

Figure 2.1: Basic parts of a SIP message

ID, Max-Forwards, and Via. The "To" header field indicates the target of the request, which is represented as a SIP URI. The "From" header field is similar, but indicates the request's originator. The "CSeq" header holds a sequence number that uniquely identifies transactions in a session and allows them to be ordered. The "Call-ID" uniquely identifies all registrations or a particular invitation of a particular client; in other words, the "Call-Id" is a globally unique identifier of the call. The "Max-Forwards" and the "Via" header fields have routing-specific purposes [41]. Figure 2.2 presents an example of a SIP message showing the mandatory header fields.

```
INVITE sip:Bob@example.com SIP/2.0

To: sip:Bob@example.com
From: sip:Alice@example.com
Call-Id: 64f1e9
Cseq: 43 INVITE
Max-Forwards: 32
Via: SIP/2.0/UDP pc33.example.com
Contact: sip:Bob@pc33.example.com
```

Figure 2.2: An example of SIP message

2.1.3 SIP Architecture

SIP infrastructure consists of: SIP user agents, a SIP registrar, a SIP proxy server, and a SIP redirect server, as shown in Figure 2.3. A SIP user agent is an endpoint device that is identified by the user's URI. The user agent is either a user agent client (UAC), the caller, or a user agent server (UAS), the callee. The former initiate sessions and generates requests, while the latter generates responses to the requests. The SIP registrar is in charge of storing the binding information in a user location database. The binding information is the association of the user's SIP address and their IP address. The user location database is updated using the REGISTER message. The SIP proxy server is responsible for routing requests and responses between the UAC and the UAS. The SIP proxy queries the registrar to find the bindings of a

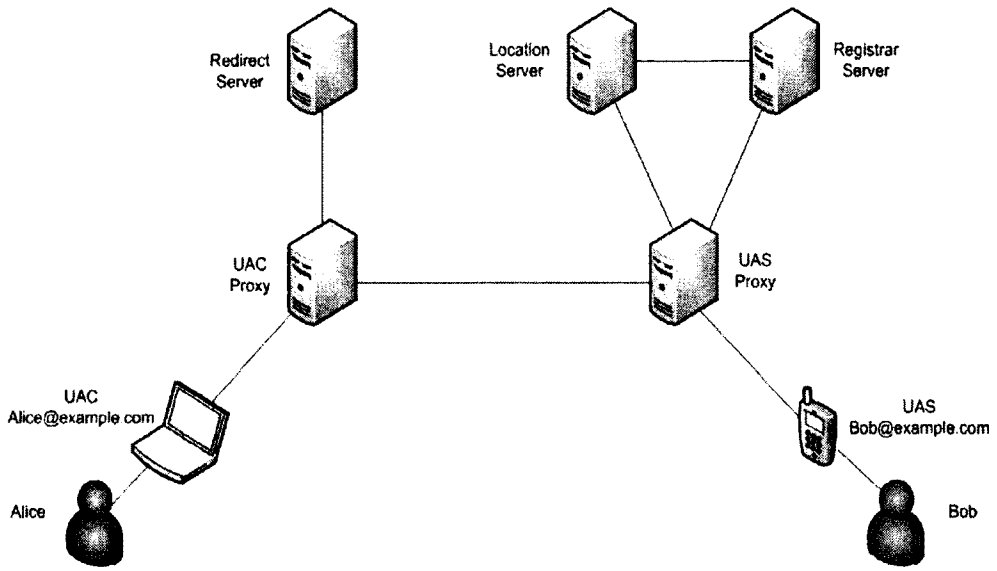

Figure 2.3: SIP components

user agent. The SIP redirect server reroutes incoming requests by responding with the IP address of the UAC to go to.

2.1.4 SIP Operations

Generally speaking, there are three main operations in SIP: registration, session set up, and session teardown. To carry out a session successfully the user agents must first register their current location (binding information) to the SIP registrar by sending a REGISTER message. The SIP registrar stores the binding information in the user location database, which is also called the SIP location service. Whenever the user bindings change, a REGISTER message is sent to the registrar to update the user agent binding. The registration process is shown in Figure 2.4. Second, the UAC sends an INVITE

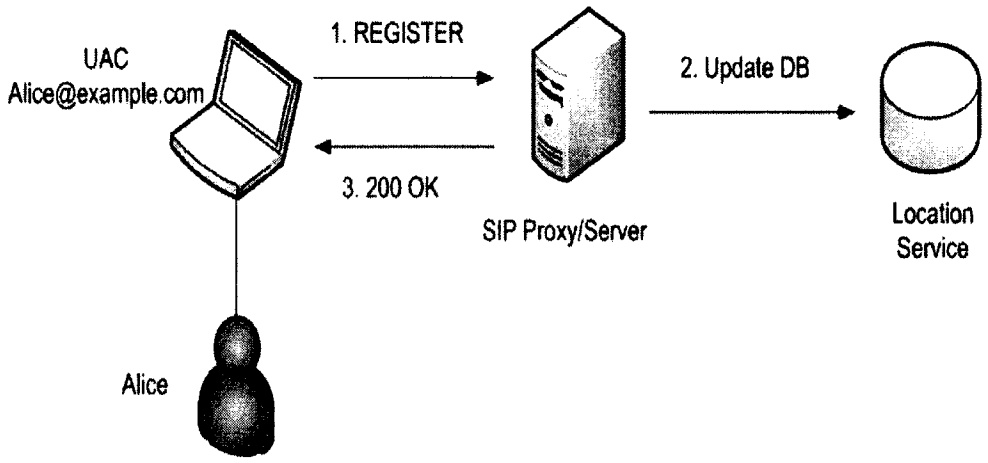

Figure 2.4: Registration process

message to the proxy, inviting the UAS. The proxy responds by sendinga "100 Trying" message back to the caller, which means that the proxy has received the message and it is routing the request to the destination. The SIP proxy queries a DNS to determine which SIP proxy serves the domain that the UAS belongs to. The INVITE message is forwarded to the proxy responsible for the callee's domain, which in turn queries the service location to find the IP address (the bindings) of the UAS. this binding information is used to send the INVITE message to the destination. A "180 Ringing" message is then sent back to the UAC. When the callee answers the call, a 200 OK message is also sent back to the caller. The UAC completes the session establishment by sending an ACK message to the UAS. Finally, the actual media session is carried out using the Real Time Protocol (RTP). Figure 2.5 shows the process of establishing a call. If any participant wishes to end the

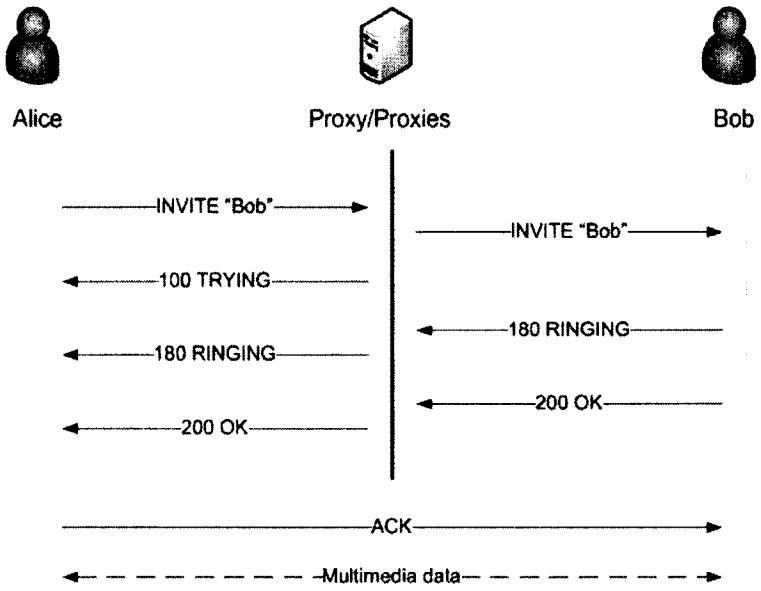

Figure 2.5: Call establishment in SIP

session he/she sends a BYE message to all other participants. The receiver confirms receipt of the BYE message by sending a 200 OK message, which terminates the session.

2.2 Overview of Ad hoc Networks

A wireless network, as the name suggests, is not connected by any cables; instead it uses radio waves to connect the nodes. There are two types of wireless network: infrastructure and ad hoc. Infrastructure networks rely on centralized entities such as routers, base stations, and access points. Ad hoc networks are the complete opposite because there are no centralized com-

ponents. Instead, each node acts as a router and may also have forwarding capabilities. The nodes are limited in resources like battery life and computation capacity. Any and all nodes in the ad hoc network can be mobile, which makes the network topology dynamic and unpredictable. The links between nodes are constrained in bandwidth and can be either symmetric (both directions have some level of bandwidth) or asymmetric (each direction has different transmit-receive characteristics).

Forwarding and routing both use routing tables, but a distinction can be made between them. Forwarding is the retransmission of received packets based on the information in the routing tables. Routing is the process of populating the routing table. Routing can be conducted in two ways: source routing or hop-by-hop. With source routing, a list of intermediate node addresses specifying the path that must be followed to reach the destination is included in each packet. For each distination, the source routing protocol stores a list of intermediate nodes in the routing table that provides a path to the target. With hop-by-hop routing, the addressing information in each packet includes only the source address and the destination address. For each destination, hop-by-hop routing protocols store only the address of next node to be used to reach the destination in the routing tables.

There are two main approaches for routing protocols in ad hoc networks: reactive and proactive. In the reactive approach, routes are discovered and maintained only if a packet needs to be delivered to a destination. Some well known reactive routing protocols include Dynamic Source Routing (DSR) [25] and Ad hoc On-demand Distance Vector (AODV) [35]. In the proactive approach, routes are discovered and maintained whether there are packets to

be delivered or not. The Optimized Link State Routing protocol (OLSR) is a well known proactive routing protocol [19].

AODV is an Internet Engineering Task Force (IETF) standard authored by Perkins *et al* [35]. It is a reactive routing protocol that uses hop-by-hop routing. If a connection is required, the source node broadcasts a Route Request (RREQ) message. The RREQ message floods the network until it reaches a node that has a valid route to the destination. Then, that node sends back a Route Reply (RREP) that includes the valid route. Since many routes might be discovered from a single RREQ message, the source node chooses the route that has the least number of hops.

The OLSR protocol also uses hop-by-hop routing but is a proactive routing protocol. It first uses HELLO messages to discover its one and two-hop neighbours. Each node then selects its multi point relays (MPRs), such that there exists a path to each of its 2-hop neighbours via a node selected as MPR. The function of MPR is to disseminate the topology control messages (TC) that contains routing information. Basically, OLSR is a routing-only protocol; forwarding is then done following the information in the routing tables.

The reactive approach generates less control traffic, but some latency is expected at the beginning of the establishment of a session because a route to the destination needs to be discovered or repaired. On the other hand, the proactive approach generates more control traffic, but packets expect less latency because the routes are already discovered. Because dynamic networks can have high mobility, the proactive approach is not a suitable candidate.

Besides these two general routing approaches, there exists a hybrid ap-

proach that combines ideas from both the reactive and proactive protocols. The Zone Routing Protocol (ZRP) is one example of this hybrid approach. Because most of the traffic in ad hoc networks is between nodes that are geographically close, ZRP selects either reactive and proactive routing based on distances. The proactive approach is used to deliver packets to any short-distance destination, and the reactive approach is used to deliver packets to any long-distance destinations.

In ad hoc networks, one way to make the network more scalable is to divide the network into partitions. Each partition is a cluster that represent a subset of the network's nodes. Clustering by itself is not a routing protocol, and any routing protocol can be modified to be a cluster-based routing protocol. For example, it is possible to have a cluster based AODV or a cluster based OLSR. The nodes that form a cluster elect a cluster head that knows its members and how to reach them. The cluster heads can communicate with each other, and they form a virtual topology that consists only of cluster heads. Each cluster head acts as a router for its members, where the cluster members send all their traffic to their cluster head, and the cluster head forwards these packets on behalf of its members.

Ad hoc network users can also benefit from the decentralization inherent in ad hoc networks and the Internet when both are used at the same time via an integrated Mobile Ad hoc NETwork (MANET). In integrated MANET, the ad hoc network is connected to the infrastructure network (the Internet) through a gateway. Integrated MANET is also known as hybrid MANET, converged MANET, and Internet connected MANET.

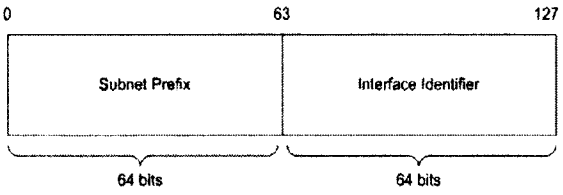

Figure 2.6: IP address

2.3 Overview of CGA

An IP address consists of two parts: the network identifier and an interface identifier. The network identifier, also called the subnet prefix, is used to determine a node's location in the Internet topology. The interface identifier denotes an interface of a particular node. IP address version 6 (IPv6) defines 128-bit IP addresses where the left-most 64 bits are the subnet prefix and the right-most 64 bits are the interface identifier, as shown in Figure 2.6.

A Cryptographically Generated Address (CGA) [8] is a self certified IPv6 address that cryptographically binds a public key with the IPv6 address of the public key owner. The basic idea of CGA is to compute a cryptographic one-way hash function of the public key and auxiliary parameters. The hashed value is used as the interface identifier of the IPv6 address. Figure 2.7 simplifies the generation of CGA. The binding between the IP address and the public key is verified by recomputing the hash value and comparing it with the interface identifier. The corresponding private key is used to sign messages sent by the CGA owner and achieve message integrity. The receiver can authenticate the address owner by verifying the CGA and signature based on the knowledge of the public key and auxiliary parameters. CGA guarantees

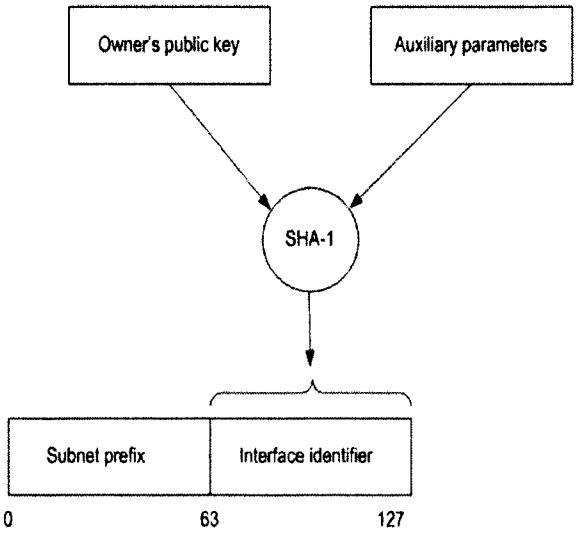

Figure 2.7: CGA generation (simplified)

un-spoofable addresses because the claimed CGA cannot be generated using any other public key and no one can sign the messages unless the private key is known. What makes CGA suitable for ad hoc networks is that there is no need for security infrastructure, such as a certificate authority (CA), public key infrastructure (PKI), or other trusted servers.

2.3.1 The CGA Data Structure

The CGA is associated with a public key and auxiliary parameters referred to as the CGA data structure parameters (CGADSPs). CGADSP consists of the following set of parameters: modifier, subnet prefix, collision count, public key, and extension fields, as shown in Figure 2.8. The modifier is a random 128-bit unsigned integer. It is used to strengthen the robustness of

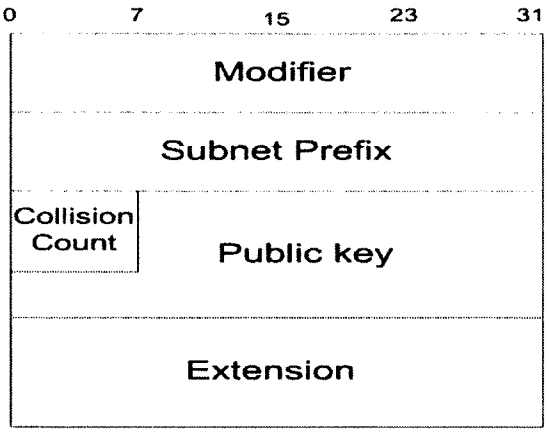

Figure 2.8: CGADSP

the hashed value and to enhance privacy by adding some randomness so that two generation processes with the same public key will result in two different addresses and linkability is not possible. The subnet prefix is the leftmost 64 bits of the CGA address. The collision count is an 8-bit unsigned integer that takes the values of 0,1, or 2. Duplicate Address Detection (DAD) is used to prevent address collisions by preventing two nodes in the network from having the same CGA. Whenever DAD detects a collision, the collision count is incremented by one.

The extension field is an optional variable length field that may be used for additional data items that need to be included in the CGA data structure. For example, the SIP URI of the address owner could be used in the extension field, so that the resulted CGA binds both the public key and SIP URI of the CGA owner. The extension field itself consists of three fields: extension type, extension data length, and extension data, which are described in the RFC 4581 document [9]. As a brief summary, the extension type field is a

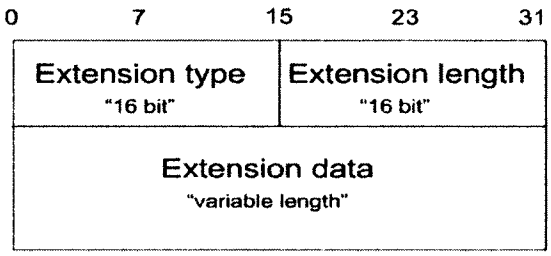

Figure 2.9: The extension field of the CGADSP

16-bit identifier of the extension type, the extension data length field is a 16-bit unsigned integer that determines the length of the extension data field, and the extension data field is a variable length field that contains actual extension data. Figure 2.9 provides a view of the extension field.

2.3.2 CGA Generation

The CGA has a security parameter (Sec) that determines its security level. The security parameter is a 3-bit unsigned integer that takes a value between 0 and 7. It is encoded in the three leftmost bits of the interface identifier.

The generation of the CGA takes three inputs: an initial modifier, a public key, and a security parameter. It outputs a new address and CGADSP. The generation process consists of two phases: the computation of "Hash2" and the computation of "Hash1". Hash1 is composed of the 64 leftmost bits resulting from hashing the CGADSP. Furthermore, it forms the interface identifier of CGA before setting its three left-most bits to Sec and the 6th and 7th bits, which are known as "u" and "g" bits respectively, to zero. Hash2 is also the hash of the CGADSP, but it is computed to find the value of the

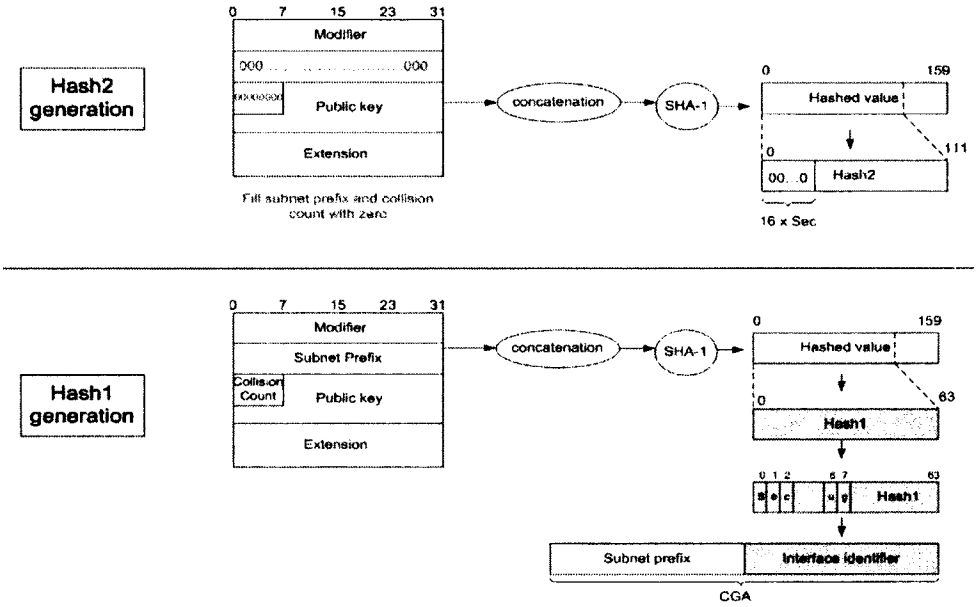

Figure 2.10: The generation of Hash1 and Hash2

modifier that causes the (16 x Sec) leftmost bits of the hash digests to be zeros. The modifier is incremented by one each time the computed hash value does not satisfy the condition. Finding a modifier that satisfies this condition is time consuming. Whenever the security parameter increase, the cost of both the CGA address generation and brute-force attack increase. Therefore, Sec increases the computational power/time on both the generator and the attacker, but increases the security level of CGA. The generation CGA that composed of the generation of Hash1 and Hash2 is illustrated in Figure 2.10

The detailed steps of generating CGA are shown in Figure 2.11 and as follows [8]:

1. Set the modifier value to a random 128-bit value.
2. Concatenate the modifier, 72 zero bits, the encoded public key, and

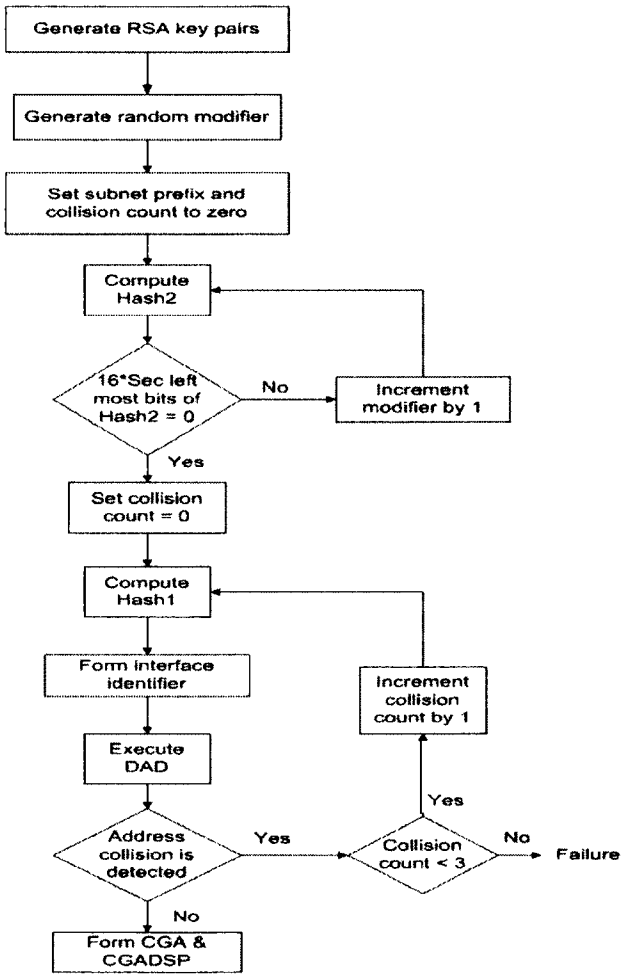

Figure 2.11: CGA generation algorithm

any optional extension fields. Execute the Secure Hash Algorithm-1 (SHA-1) algorithm on the concatenation. The leftmost 112-bit of the resulted 160-bit SHA-1 hash value are the Hash2.

3. Compare the (16 x Sec) leftmost bits of Hash2 with zero. If they are not zero, increment the modifier with one and go back to step 2.

4. Set the collision count to zero

5. Concatenate the final modifier value, the subnet prefix, the collision count, the public key, and any optional extension fields. Execute the SHA-1 on the concatenation. The leftmost 64 bits of the 160-bit SHA-1 hash value are Hash1.

6. Form the interface identifier by setting the leftmost 3 bits of Hash1 to the Sec value and the reserved "u" and "g" bits to zero.

7. Concatenate the subnet prefix and interface identifier to form a 128-bit IPv6 address.

8. If an address collision is detected, increment the collision count by one and go to step 5. After the detection of three address collisions stop and report the error.

9. Form the CGA data structure parameters by concatenating the final modifier value, the subnet prefix, the final collision count, the public key, and any extension fields.

2.3.3 The CGA Address Verification

The process of verifying the bindings between the IP address and public key takes as input the CGA address and CGADSP. The verification either succeeds or fails, as shown in Figure 2.12, according to the following steps

[8]:

1. Check that the collision count value is 0, 1, or 2, and that the subnet prefix value of the data structure is equal to the subnet prefix of the CGA address. The CGA verification fails if either check fails.

2. Concatenate the field of the data structure and execute SHA-1 on the concatenation. The leftmost 64 bits of the 160-bit SHA-1 value is Hash1.

3. Compare the Hash1 value with the interface identifier of the CGA address, but ignore the bits 0, 1, 2, 6 and 7. If the two values differ, the CGA verification fails.

4. Read the security parameters Sec, which is the three leftmost bits of the interface identifier of the CGA address.

5. Concatenate the modifier, 72 zero bits, the encoded public key, and any optional extension fields. Execute the SHA-1 algorithm on the concatenation. The leftmost 112-bits of the resulted 160-bit SHA-1 hash value are the Hash2.

6. Compare the (16 x Sec) leftmost bits of Hash2 with zero. The verification fails if any one of these is not zero. Otherwise, the verification succeeds.

2.3.4 Security Consideration of CGA

There is a security risk because CGA is not certified itself. To illustrate, because the public key used to generate the CGA is not authentic, an attacker can use his/her own unauthentic public key to generate a vlid CGA. Then, the attacker can use the corresponding private key to sign any message. However, the attacker can not take someone's else CGA and sign messages as he/she is the address owner unless the attackers knows the corresponding private key. As indicated by Cao et al [16], this weakness is known as *Unauthentic Key*

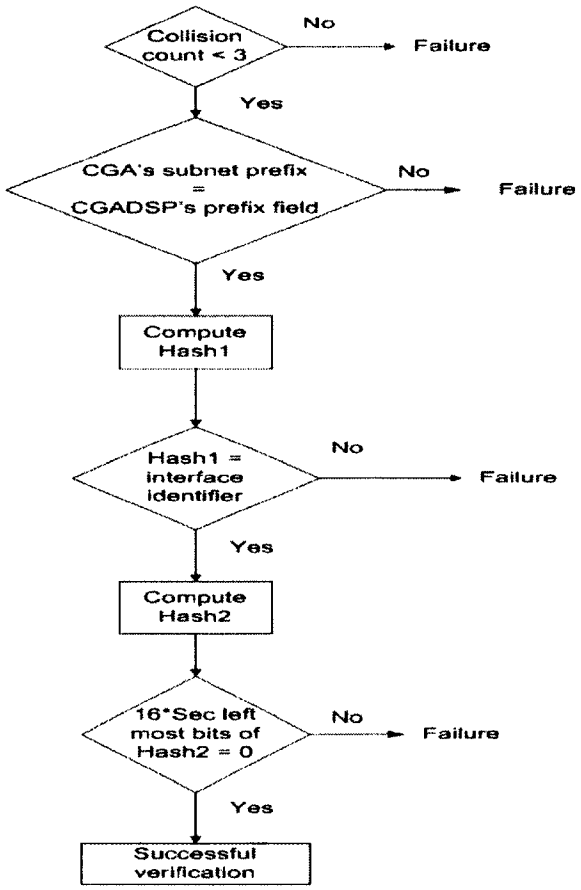

Figure 2.12: CGA verification algorithm

Attacks. Coe et al propose a solution to this attack by using an authenticated public key. They suggest using Identity Based Cryptosystem (IBC) where a trusted key generation/management authority is involved. Therefore, this solution is not applicable to ad hoc networks.

Chapter 3

SIP in Ad Hoc Networks

Several attempts have been proposed in the literature to enable SIP in ad hoc networks. The centralized nature of SIP architecture, where central proxies and location servers are used, does not suit ad hoc networks. Endpoint discovery is a challenge in ad hoc networks due to this absence of a central location service. Efforts have been made to distribute proxy/registrar functionalities among nodes and to use a local cache. The existing solutions can be classified into three categories:

1) Using Service Location Protocol (SLP) [46].

2) Broadcasting the binding information.

3) Using a clustering approach.

Also, integrated MANET (mobile ad hoc network) is a form of ad hoc network in which the mobile ad hoc network is connected to an external network. Several proposed solutions claim to solve the issue of discovery in integrated MANET and are discussed in this chapter.

Since ad hoc and peer to peer (P2P) networks have decentralized archi-

tectures, it is worth mentioning the effort that has been made to enable SIP in P2P networks.

3.1 Using Service Location Protocol

SLP is a protocol used to find services by name and properties. A user agent generates a request specifying the characteristics of the service that they require. A service agent receives the request and replies with the location of the service.

Stuedi *et al.* [43] use a local cache to store users' binding information. They propose a middleware infrastructure for session setup and management in MANET called SIPHoc. They use SLP for SIP endpoint discovery. One of the SIPHoc components is MANET SLP, which provides fully distributed registration and lookup services. An SLP service query is sent via routing messages by piggybacking the binding information onto these routing messages. This is done through a routing handler plug-in. The routing handler is a software module that receives raw routing packets and generates altered packets that include the binding information. The routing handler makes SIPHoc independent of any routing protocol. Also, each node has a SIPHoc proxy that serves as an outbound SIP proxy for the local SIP application. Other components of SIPHoc include gateway and connection providers. While the gateway provider turns a node into a gateway, the connection provider manages the node's connection to the Internet.

Similarly, Leggio *et al.* [29] point out that it is possible to know the binding information of other nodes by sending an SLP query request that

includes the URI of the user. However, only nodes that send queries can learn about tother nodes. This means that nodes that only receive the query cannot learn the binding information of the requested node. Hence, more overhead is imposed.

3.2 Broadcasting the Binding Information

Advertising the binding information of users using a broadcast is an idea proposed by several groups. In this case, each node stores the received binding information in its local cache. However, this approach introduces significant overhead, so it works better in small ad hoc networks.

Leggio *et al.* [29] present a framework to enable a decentralized SIP (dSIP) in ad hoc networks. Using broadcasting, each node sends its binding information in a REGISTER message. The receiver stores the binding information in its local cache and unicasts a 200 OK message that contains its own binding information. In the case of an INVITE message, the local proxy module does a lookup in the local cache and sends an INVITE message to the callee. Similarly, the main idea of Khlifi *et al.* [28] is to periodically broadcast a REGISTER message that contains the binding information of the user. Upon receiving the REGISTER message, the receiver saves the binding information in its cache for a limited time. In addition, they introduce a new field called "Conf-Id" which contains the identities of all ongoing conferences. Each REGISTER message includes the Conf-Id of all conferences for which the sender is the leader. To join a conference, a node sends an INVITE message, which includes the Conf-Id field, to the leader. Moreover,

they propose an enhancement to reduce the overhead caused by broadcasting. They discuss unifing the network layer and the application layer. For example, in the case of AODV, instead of using RREQ and RREP messages, the routing information is included in the SIP REGISTER message.

One of two approaches proposed by Banerjee *et al.* [11], introduces two new messages to perform endpoint discovery. They are SIP Route Request (SIPRREQ) and SIP Route Reply (SIPRREP). These two messages are similar and borrowed from AODV RREQ and RREP messages. First, SIPRREQ, which contains the binding information of the user, is flooded. Second, the receiver returns a SIPRREP that contains its own binding information to the sender. It is worth mentioning that this approach is independent of the underlying routing protocol.

O'Doherty, in an expired internet draft [34], proposes a method to periodically advertise the presence of SIP clients and their binding information by sending "hello" messages. Using broadcasting, a periodic hello message that contains the binding information of the client is sent . Any node receives the hello message extract the binding information and stores it in its local cache. The author also claims that the REGISTER message can be used for the same purpose.

3.3 The Clustering Approach

The benefits of a cluster-based routing protocol could be leveraged to enable SIP in ad hoc networks. In the second approach by Banerjee *et al.* [11], SIP endpoint discovery is integrated with a distributed cluster-based routing

protocol. The cluster-based routing protocol creates a virtual topology of cluster heads. The cluster heads form a backbone network that is used in the routing of SIP messages. Meanwhile, each cluster head hosts an SIP registrar and proxy. The SIP user agents, which are the cluster members, register themselves by sending a REGISTER message to their cluster head. Similarly, in the case of invitation, cluster members send INVITE messages to its cluster head. The cluster head functions as a proxy to locate the invited user agent server (UAS) by flooding other cluster heads with the INVITE message. The cluster head of the cluster where the UAS is located, forwards the INVITE message to the UAS.

3.4 SIP in Integrated MANET

As mentioned earlier, SIP endpoint discovery is a problem in integrated MANETs. There is also the additional problem of having two different types of nodes. Some nodes are located in the MANET and others are in a different type of network, in the Internet for example.

Manner *et al.* [32] propose a solution to solve this problem. They introduce a SIP gateway that allows communication between nodes in an ad hoc network and nodes in the Internet. The gateway has registrar and proxy functionalities. The gateway ensures that nodes in the Internet are not aware that the called parties are in an ad hoc network. The gateway modifies SIP message headers in order to add the gateway in the messaging path. To illustrate, if an Internet node invites an ad hoc node, the gateway adds a "Path" header field. The Path header field tells the registrar that all mes-

sages addressed to the ad hoc node must be routed to the address in the Path header field. Also, a "Record-Route" header field is added to the SIP message if an ad hoc node is calling an Internet node. The authors claim to use a modified SLP, which is adopted for use in ad hoc networks, to perform the tasks of gateway advertisement and discovery. Similarly, Balvo *et al.* [10] propose an approach where the gateway advertises its presence using routing messages. Then, each node creates a unique address formed by using the gateway address as a prefix. The nodes unicast a gateway confirmation message back to the gateway, which in turn keeps a list of the global addresses of MANET nodes. Another similar gateway-based approach has been proposed by Castro and Kassler [17].

3.5 SIP in Peer to Peer Networks

Peer to peer (P2P) networks are similar to ad hoc in that they are decentralized. There have been several attempts to enable SIP in P2P networks [33], [26], [14]. A general feature, of most of this research, is to have two layers: P2P and SIP. Indeed, SIP is layered on top of P2P. The P2P layer is only used by SIP for endpoint discovery. For example, Fessi *et al.* [22] use a Dynamic Host Table (DHT) to resolve endpoint discovery needed by the SIP layer. While P2P and ad hoc networks have some similarities, P2P nodes are static and ad hoc ones are mobile. Accordingly, Barbeau [13] and Banerjee *et al.* [11] have already pointed out that P2P solutions are not applicable in ad hoc networks due to the randomness in node mobility.

Chapter 4

Related Work

In this chapter, SIP authentication is described. Several solutions have previously been introduced to provide authentication for SIP users; however, most of these solution are not applicable to SIP within ad hoc networks because they depend on a centralized certificate authority.

4.1 SIP Security Satndards

The SIP specification offers authentication security based on HTTP digest authentication. HTTP digest authentication is a challenge-response mechanism in which the client sends a request to a server and the server challenges the client in order to authenticate them. Before the process starts, the client and the server must pre-establish a password; this password is used to authenticate the client's identity. Then, the server sends a challenge composed of a random string to the client. The client's response to the challenge is a hash value computed using the random string, as well as

the client's username and password. Based on the client's username, the server retrieves the client's password and verifies his/her authenticity. SIP also provides secure URIs, referred to as SIPS URIs(sips:alice@example.com, for example). It uses Transport Layer Security (TLS) to provide end-to-end security. TLS offers mutual authentication by exchanging certificates during a handshake process. Furthermore, the Secure/Multipurpose Internet Mail Extension (S/MIME) is specified as another way to provide authentication. SIP messages can carry MIME bodies that contain the sender's certificates. However, as indicated in several works [27], [50], [21], [48], SIP authentication mechanisms are vulnerable to several attacks.

The IETF also introduced an authentication enhancement extension for SIP [36]. In this extension, two new header fields are introduced: Identity and Identity-info. SIP user agents connect and authenticate with a SIP server that runs an authentication service. When receiving a message from an authenticated user agent, the server signs the message using its domain certificate. The signature is then added to the Identity header field, and an address where the certificate can be fetched is included in the Identity-info header filed.

4.2 SIP Security in Ad hoc Networks

SIP Security in ad hoc networks has been addressed by Leggio *et al.* [30, 32]. In their research, they try to achieve both mutual authentication of the participants in a session as well as message integrity. They adapt the SIP authentication extension [36] and assume the presence of pre-distributed self-

signed certificates. Each node signs its SIP messages by hashing all SIP header fields; the signature is included in the *Identity* header field, which is discussed in Section 4.1. When a node recieves the signed message, the signature is verified. There is a server module in each node that is responsible for the certificate verification. Each node has a database of certificates.

4.3 Authentication in SIP

Several papers [50], [20], [49], [51] investigate enhancing SIP authentication using HTTP Digest authentication. Yang *et al.* [50] use the Diffie-Hellman key exchange algorithm, which is based on the difficult-to-solve discrete logarithm problem. However, their scheme comes with a high computation cost; hence, Durlanic *et al.* [20] propose an efficient SIP authentication scheme using Elliptic Curve Diffie-Hellman (ECDH) algorithm, which is also based on the discrete logarithm problem. It reduces the execution time because it has a small key size and offers the same security level of classical cryptosystems. Wu *et al.* [49] point out an efficient SIP authentication scheme and key agreement protocol at the same time using Elliptic Curve Cryptography (ECC). Along with Yang *et al.*, they claim that their scheme is secure against several attacks, including replay, off line password guessing, and server spoofing. However, Yoon *et al.* [51] claim that Durlanik's and Wu's schemes are vulnerable to offline password guessing attacks, a Denning-Sacco attack, and stolen-verifier attacks. They propose an authentication scheme that overcomes these attacks and is based on the HTTP digest authentication and ECC. While the Denning-Sacco attack occurs when an attacker compro-

mises an old session key and tries to find a long term private key or other session keys, the stolen-verifier attack involves an attacker stealing the password verifier (hashed password) from the server and using it to masquerade a legitimate user in the authentication process.

Wang and Zhang [47] also propose a secure mechanism for authentication and session key management in SIP. They use certificate-less public key cryptography based on the bilinear Diffie-Hellman problem. This mechanism relies on a trusted third party (TTP) that cooperates with the participants to generate their private key, which is unknown to the TTP. A participant's public key is bound to their SIP identity. Their scheme called Secure Authentication and Key Agreement mechanism (SAKA) and its handshake process is based on the HTTP digest authentication handshake. After the participants authenticate themselves, they can compute the shared secret key, which is required to establish a secure session using the Secure Real-time Transport Protocol (SRTP). Similarly, Liao and Wang [31] propose an authentication and key agreement scheme where the TTP cooperates with the participants to generate their private keys. They use a self-certified public key based on Elliptic Curve Cryptography (ECC) while keeping the main structure of the HTTP digest authentication. However, in their scheme the TTP issues a smart card that contains the secret parameters that are sent to each party via a secure channel. The smart card is used to store the parties' password and helps to generate the long term private keys, which are also stored in this card.

Ring *et al.* [40] bounce a SIP authentication and key agreement mechanism using identity-based cryptography, where the public key is function of

the user's identity (the user's SIP identity in this case). The private key is calculated by a trusted authority (TA). Similar to many other approaches, they use the main SIP HTTP digest authentication structure, but they use identity-based ECC, which allows the generation of secure session keys between the two participants. However, the use of elliptic pairing makes this mechanism very costly in terms of computation and it is not suitable for constrained devices. A similar solution has been offered by Han *et al.* [24].

Geneiatakis and Lambrinoudakis [23] introduce a new SIP header field called "Integrity-Auth" to protect against SIP signalling attacks and to ensure integrity and authenticity. Users' passwords are stored in a table that is maintained by a server. The value of the new header is the hash value of the user's password combined with some known parameters. The verification of the new header value provides message integrity and authenticity of the users.

Chapter 5

Security Extension of SIP in Ad Hoc Networks

5.1 Contribution Overview

The traditional mechanisms for providing authentication in ad hoc networks are not suitable for two reasons. First, traditional authentication solutions rely on centralized entities, such as servers, to verify the authenticity of the node. Second, it is a well known problem that traditional Certificate Authorities, which are centralized solutions, are not always accessible in ad hoc networks. Consequently, there is a need to provide a unique authentication mechanism specially designed for ad hoc networks.

This work proposes a distributed authentication mechanism that does not rely on any sort of centralized certificate authority. The paradigm of social networks is combined with CGAs to provide authentication, prevent several types of attack, and achieve message integrity.

CGAs are used to prevent impersonation and other attacks that result from the attacker's ability to impersonate a legitimate user. In fact, CGA guarantees the prohibition of impersonation attacks by ensuring that a claimed identity does not change during a session. In this context, since a CGA binds the public key of a node to its IP address without relying on any certificate authority, it is reasonable to look at CGA as a self-certified public key. Thus, it fits the characteristics of an ad hoc network. In short, CGA will function as a replacement of the public key infrastructure; moreover, it provides security against spoofing attacks and any subsequent attack. However, CGA cannot validate a user's identity in the initial phase, i.e. if a new node is joining the network, CGA cannot guarantee that the node is who it claims to be. Once the CGA is accepted or used in the network, authentication is guaranteed from that point forward. Consequently, a mechanism needs to be used to authenticate CGA users in the initial phase.

The proposed security mechanism inherits CGA's security properties but also its limitations. To overcome the CGA limitation in providing authentication, a new authentication mechanism is introduced. The design of the new mechanism benefits from concepts in social networking. Indeed, it is an integration of the social networks paradigm into VOIP in ad hoc networks. Basically, the participants of the network play the roles of the certificate authority, not by signing the public key certificates, since self-signed public keys using CGA are used, but by ensuring the identity of the CGA owner. Validating the identity of the CGA user is the responsibility of the nodes in the ad hoc network.

Consider that when a node calls another node it is a friend, a friend

of a friend, or someone who wants to be a friend. In the new authentication mechanism, participants pass their CGA addresses along with the CGA data structure parameters (CGADSPs) in some offline way. In fact, the participants are usually friends or acquaintances. They may exchange their CGA during a physical meeting, by passing their business cards for example, or they could exchange it via email, text message, etc. Then, each node maintains the CGA addresses and CGADSPs of the nodes that it can authenticate in a friends list. These nodes have already had their identity validated. Whenever a node receives a SIP message, it checks to see if it is from a friend or not. Friend identities are authenticated by validating the CGA address in the message using the CGADSPs stored in the friend list. Also, if the message is from a non-friend, the user can do a friend lookup by asking his/her friends if any of them can authenticate the identity of the non-friend node.

To use the new authentication mechanism within the SIP protocol, an extension is added to the SIP protocol. Also, the integration of the social network paradigm into the VOIP application adds more services for the user beyond the actual VOIP services. The new services provided to the user include: "add friend", "accept friend", "recommend friend", "block or unblock friend", and "delete friend". The new extension to SIP requires some modification to the peer interface of the SIP protocol. Also, some modification has been introduced to the processing of SIP messages. This includes modifying the process of generating and processing SIP requests.

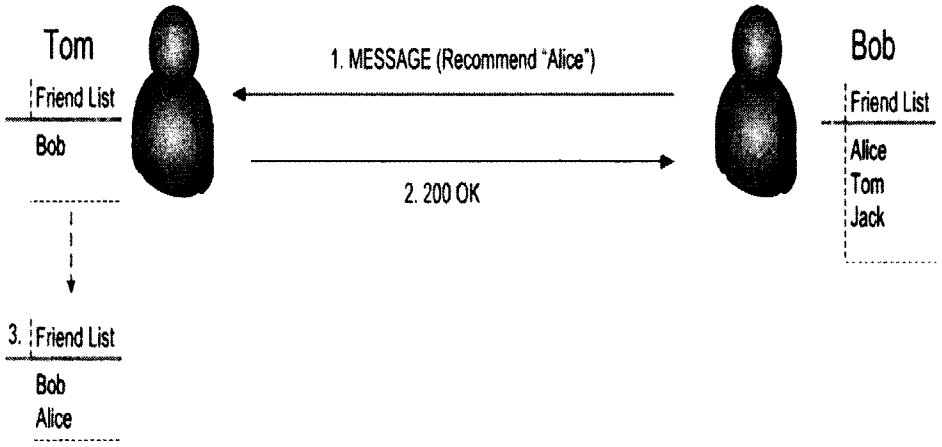

Figure 5.1: Friendship recommendation

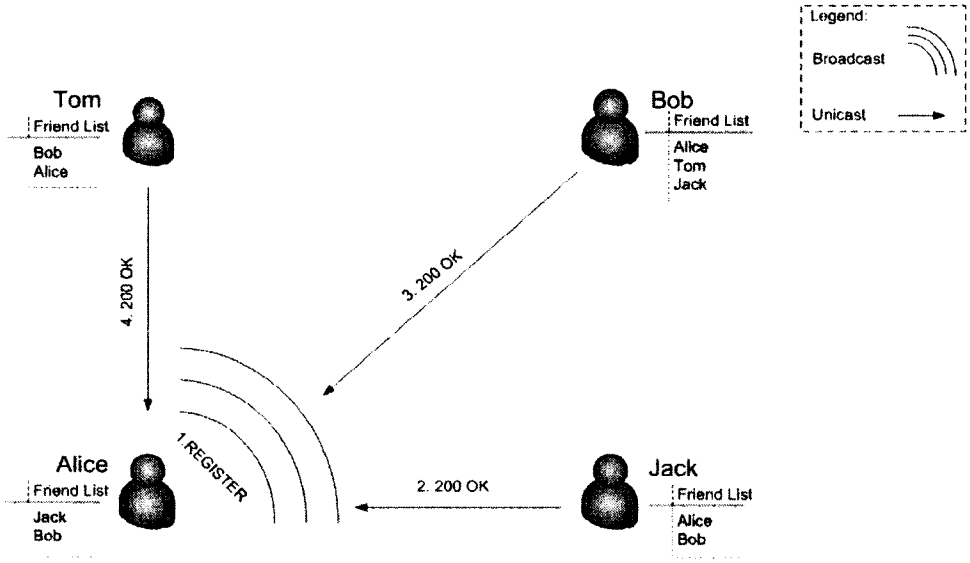

Figure 5.2: Alice joining

5.1.1 Possible Scenario

A possible scenario in which to apply the new security extension is as follows:

Consider a network of four nodes: Alice, Bob, Jack and Tom. Alice, Bob, and Jack are friends, but Tom is only a firend to Bob. The friends have met and exchanged their CGA addresses offline. In their friends list they have stored each other's CGA along with the CGA data structure parameters. Bob wants to recommend Alice as a friend to Tom, so he sends a friendship recommendation to Tom. Tom accepts the recommendation and adds Alice to his friend list. The messages transmitted to carry out the friendship recommendation are shown in Figure 5.1.

When Alice is online, she broacasts a REGISTER message as shown in Figure 5.2. The other users are already online, and they respond by unicasting a "200 OK" message because they all have Alice in their friends list. However, Alice does not have Tom in her friends list, so she discards his response.

If Tom wants to call Alice, he sends an INVITE message to her. Because Tom is not in Alice's friends list, she does not answer his call right away. Instead, Alice can ask Bob or any of her other friends about Tom; the semantic of this question is *who knows Tom,* and from a security prospective, *who can authenticate Tom.* If she gets a positive response, which means one or more of Alice's friends know Tom, then she can accept the invitation and answers Tom's call. Figure 5.3 shows that Alice first asks Jack if he knows Tom. She sends a MESSAGE message to Jack, and the latter responds by sending a 200 OK message that carries nothing in its body, which means Jack does not know Tom. Then, Alice sends a friend lookup request to Bob, and he

responds with a 200 OK message that contains Tom's CGA and CGADSP in its body, which means that Bob knows Tom. To authenticate Tom, Alice gets his CGA from the INVITE message and verifies it based on the CGADSP received in Bob's response. Similarly, she verifies the digital signature that signs the INVITE message based on the CGADSP in Bob's 200 OK message. Finally, Alice answers Tom's call after a successful verification of his CGA and signature.

Another possibility is if none of Alice's friends know Tom. In this case it is up to Alice to accept the invitation on her own.

Yet another scenario might be if Alice decided to end her friendship with Bob for some reason. She can then block him and add him to her blocked list so that any future call or multimedia communication from him will be automatically rejected. Also, if at any time Alice decides to block one or more of her friends, they will be put into the blocked list and their multimedia communications will also be rejected. Also, Alice can delete any of her friends at any time. The deleted friends can re-request to be Alice's friend again, but the blocked friends cannot unless she unblocks them.

5.1.2 Tools

The cryptographically generated address (CGA) is used to secure the communication between the endpoints. However, a few modifications have been introduced to the CGA data structure parameters to fulfill the requirements of the SIP protocol and ad hoc network characteristics. The modifications include adding a SIP URI to the CGA data structure parameters. The SIP URI is included in the extension field of the CGA data structure parameters

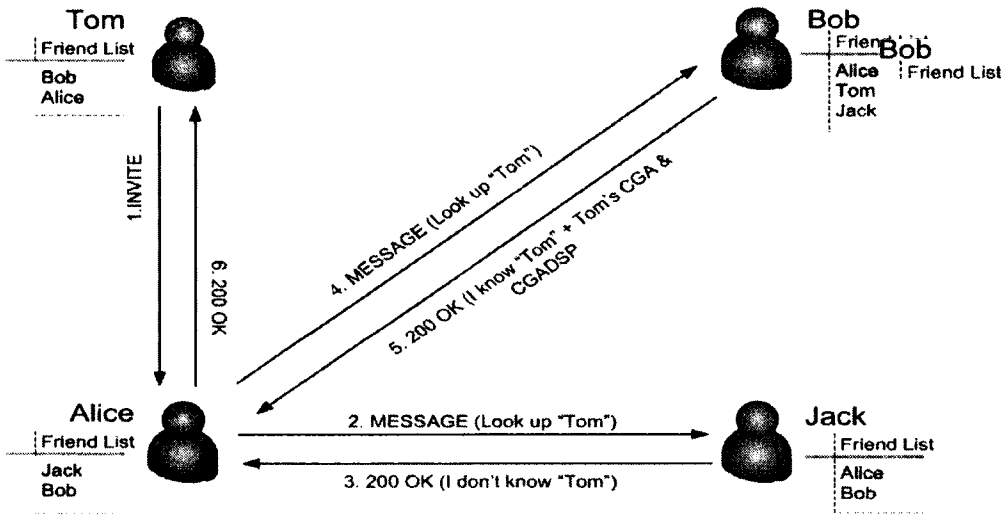

Figure 5.3: Alice receives an INVITE message from non-friend Tom

shown in Figure 5.4. Thus, the CGA address is not only bound to the public key of the user, but also to the user's SIP URI as well.

The two new headers introduced in the Authenticated Identity Management extension of SIP [36] are used. First, the "Identity" header field is used to hold the value of the digital signature of the SIP message. Second, "Identity-Info" header filed, which is defined in the extension specifications to convey a reference to the certificate of the signer, is used. Since a self-certified public key is used and there is no certificate authority to refer to in this header field, the CGA is used instead. In other words, the value of the Identity-Info header field is the CGA of message originator, which allows the receiver to verify the digital signature of the message and the CGA itself.

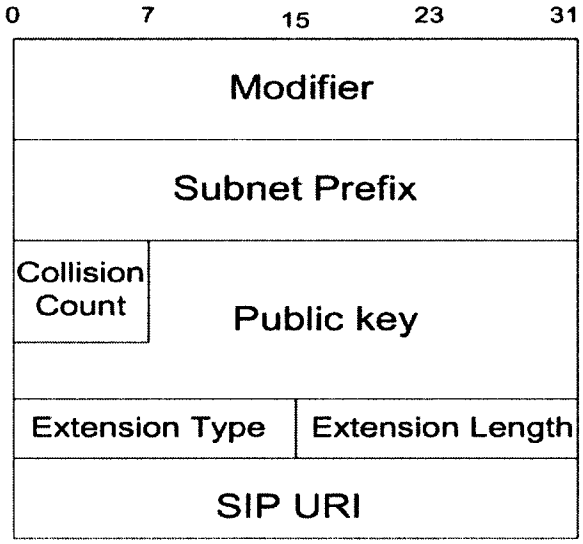

Figure 5.4: The new CGA data structure parameter

5.1.3 Security of SIP Messages

To secure SIP messages, any SIP message, whether it is a request or a response, is signed using the private key of the message originator. Generally speaking, a digital signature algorithm involves a hash function and an encryption function. First, the message that is going to be signed is hashed. Second, the resulting hashed value is encrypted using the private key of the originator to produce the signature. Then, the Identity header field is added to the message, which contains the digital signature. The Identity-Info header field is added as well and contains the CGA address of the originator as shown in Figure 5.5.

The verification of the SIP messages involves two verification processes :

1. Verification of the CGA address of the message originator ensures that

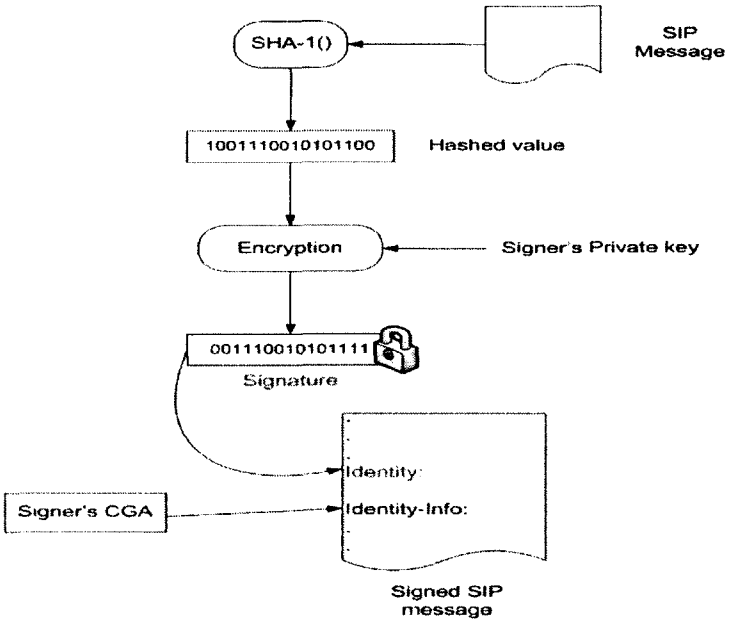

Figure 5.5: Signing a SIP message

it is a valid CGA address. In fact, the CGA address is validated using the previously stored CGADSP in the friends list. This step is to authenticate the user. If the CGA verification succeeds, it means the user owns the claimed identity (the CGA and the bounded public key).

2. Verification of the digital signature in the message to ensure message integrity. The verification process of the digital signature requires the decryption of the signature using the public key of the originator, which is stored in the friends list as a part of the CGADSP. The hashed value of the message, after excluding the identity and identity header fields, is compared with the resulting decrypted value. The verification succeeds if the two values match.

The verification of the SIP message requires that these two processes succeed. Figure 5.6 illustrates these processes.

The proposed authentication mechanism prevents several attacks that include: de-registration, tear down, denial of service, and redirecting attacks. Moreover, the SQL injection attack is prevented by using the message integrity because the signature of the message detects any modification to the message by an attacker.

5.2 SIP Protocol Architecture

Peterson and Davie [37] state that any protocol defines two different interfaces: a service interface and a peer interface. The service interface defines the operations that can be performed on the protocol. For example, in the SIP protocol, the service interface defines the operation of inviting another party to a VOIP session. The peer interface defines the structure and the

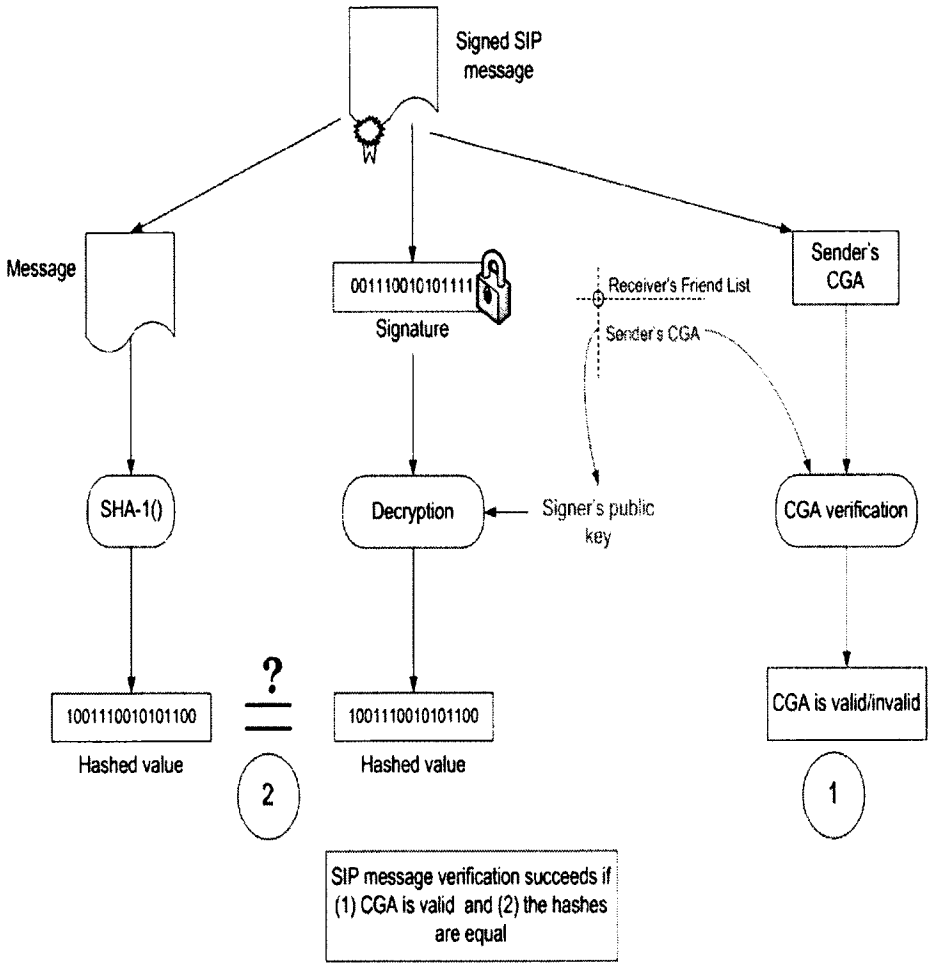

Figure 5.6: Verification of a SIP message

semantics of the messages exchanged between the protocol peers in order to implement the service provided by the protocol. To illustrate, the SIP specifications define in detail how to construct and handle the SIP messages in order to invite the second party to a session; this is considered aspart of the peer interface. Consequently, in order to add an extension to the SIP protocol both the service and peer interfaces need to be defined.

5.2.1 Service Interface

The new extension to the SIP protocol adds more services for the user of the VOIP application. It also allows the user to integrate his/her social network into the VOIP application. Thus, users are able to call their friends and see when they are ON or OFF (online or offline). They can also expand their friendship to include more friends who at first only a friend of a friend. Besides the regular services provided by any VOIP application, the new extension gives users the ability to maintain a list of their friends and to see their statuses (ON/OFF). Moreover, users of VOIP applications are able to add new friends to their friends list or accept a friendship request from someone. They also can delete a friend at any time, but the deleted friends are able to re-request to be a friend again. However, if the user wants to prevent all communication with a friend, it is better to block that friend. The user cannot receive anything from a blocked friend, including the current status of the blocked friend. Nevertheless, it is always possible to unblock a blocked friend, and all the prohibited communications will be possible again. Finally, the user can recommend one or more of his/her friends to one or more of his/her other friends. So, the new services include add, accept, delete,

block/unblock, recommend, and lookup.

When running the VOIP application for first time, users manually enter their friends' CGA addresses and CGA data structure parameters, which they got from their friends in some alternate way. As mentioned in section 2.3.3, before storing this information, the CGA address should be verified to avoid any mistakes in the information exchange. The CGA address and CGADSPs are saved in the friends list and are used by the underlying peer interface.

To add a friend manually, the following steps, also shown in Figure 5.7, are taken:

1. Validate the CGA address that was exchanged offline. If it is valid, proceed, otherwise, notify the user and exit.

2. Send a friendship request.

3. If the request is accepted create an entry in the friends list with the entered information; otherwise, notify the user and exit.

If a user receives a friendship request from someone he/she does not know, then a friend lookup process takes place. To carry out the friend lookup, the user chooses all or some of his/her friends to send the friend lookup request to. If one or more friends replies that they know that friend, then the user can add that friend to their own friends list. However, if all the existing friends reply that they do not know that user, it is up to the user to add him/her to the friends list or reject the friendship request.

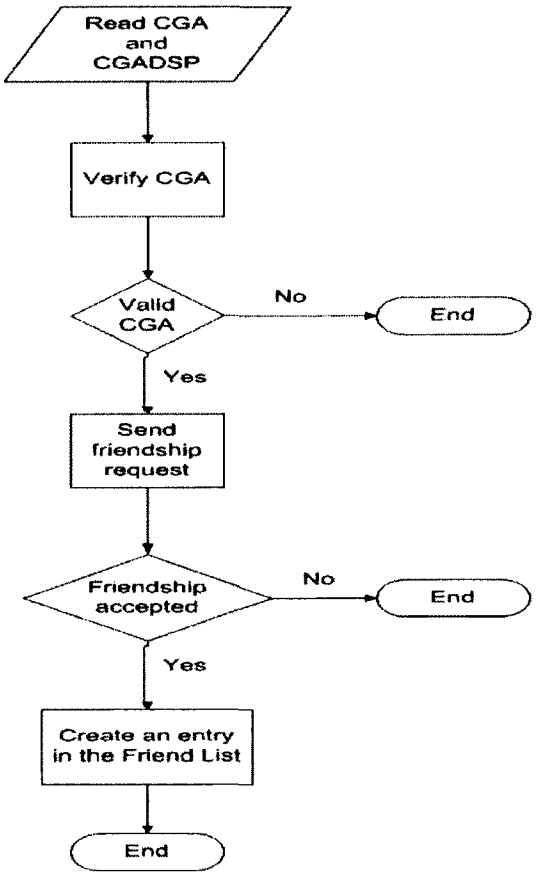

Figure 5.7: The process of adding a friend

5.2.2 Peer Interface

Each user agent's friends list takes the form of a table. The "FriendList" table consist of five columns: SIP URI, CGA, CGADSP, current status, and valid interval. Note that it is possible to have more than one CGA address and CGA data structure parameters for the same SIP URI because some users have more than one device that run SIP applications (PC, smart phone, IPAD etc.) and each of them requires a different CGA. The current status determines whether the user is ON or OFF, and is updated with SIP REGISTER messages. The current status is valid for the specific time interval found in the valid interval column. If the interval expires without receiving a new SIP REGISTER message then the user is considered to be OFF. The table is populated either manually in the set up phase or as a result of accepting a friendship.

There are three options to carry out the new services as follows:

1. Introducing new SIP messages with the standard header fields for each service. For example, it is possible to create a new SIP message called FRIENDSHIP-REQUEST to request a friendship with a user. In this option, three new SIP messages are required: FRIENDSHIP-REQUEST, FRIENDSHIP-RECOMMENDATION, FRIEND-LOOKUP.

2. Using available SIP messages and introducing new header fields. The best candidate message to alter to employ the new services is the SIP OPTIONS message since its purpose is to query about capabilities. In this option, a new header field is required to specify the service. The new header field could be called "Query", and it could have one of three values: a friendship request, a friendship recommendation, or a friend lookup.

3. Using the instant messaging SIP extension [15]. A SIP MESSAGE message is used to carry out all new services. The "Subject" header field takes one of three values:friendship request, friendship recommendation, or friend lookup.

The third option is used because, unlike the other two options, it is compatible with SIP and requires a small amount of modification to the standards.

Many SIP user agent behaviours have been modified in the new SIP extension. The modifications took place to implement the new services provided to the SIP users and to fulfill the new security requirements. The following sections describe the modified user agent behaviours.

General User Agent Behaviour

With the new SIP extension, the REGISTER message has additional semantics. It determines the current status of the user (whether the user is ON or OFF). Whenever the user agent sends a REGISTER message, they are stating that the user is ON for a specific time interval specified by the "Expire" header field. The user agent's current status is set to OFF if the time interval passes without receiving a new REGISTER message or if a new REGISTER message is sent but with the Expire header field set to zero, which means the user device is shut down. This strategy uses the Broadcasting framework, which means that REGISTER messages are broadcasted to many nodes at once. Any node that receives one of these REGISTER messages, first checks if the sender belongs to their friends list. If the sender is listed in the receiver's friends list, the receiver proceeds with checking the validity of the

CGA and the signature; otherwise, the message is ignored.

Nothing has been changed to the BYE and CANCEL messages except the addition of the security mechanisms. Indeed, the SIP specifications suggest that these two requests require special handling of authentication since they can be used by attackers to carry out tear-down attacks. The introduced authentication mechanisms prevent such an attack. Whenever a SIP BYE or CANCEL message is received, the user agent first checks the validity of the CGA address included in the Identity-Info header field. The validation is done using the CGA data structure parameters stored in the friends list. Second, the digital signature in the Identity header field is verified using the public key stored in the CGADSPs. Failing in the verification process of the signature means that the message has been altered by an illegitimate user.

User Agent Client Behaviour

To call a friend, the SIP user agent sends a INVITE message to the callee. It is possible for the caller to call one of the friends already listed in their friends list, and in this case the INVITE message is securely sent to that friend, or it is also possible to call an unknown user. For example, if the user is in an airport or riding a bus and wants to establish a VOIP session with anyone at random, then there is no need for previous knowledge between the participants. Although, security is not as important in this case, the introduced security mechanisms are still used. The difference between an INVITE message sent to a friend and one sent to a non-friend is that the INVITE message sent to an unknown user contains the caller's CGADSP to allow the verification of the CGA and establish a secure session. In this

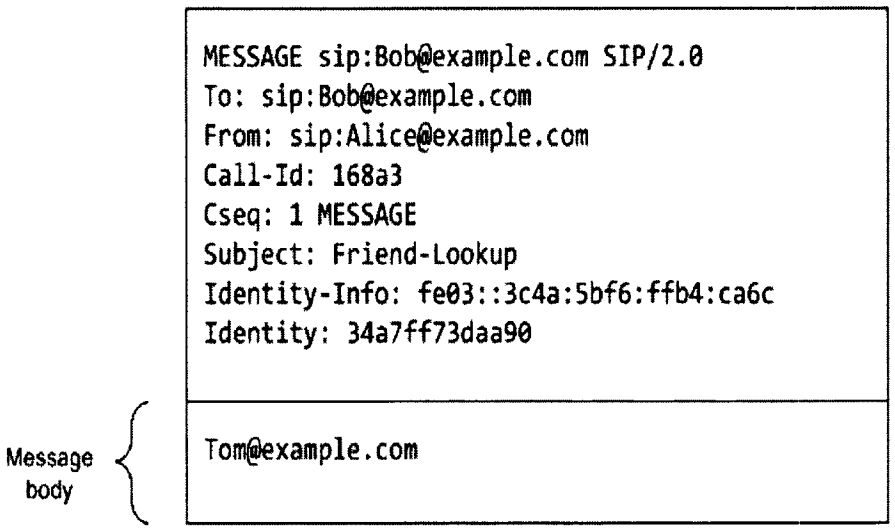

Figure 5.8: Friend lookup request "simplified"

way, the participants' identities are not going to be impersonated during the session.

As mentioned before, the SIP extension for instant messaging [15] is used to carry out the new services. it is done as follows:

1. A MESSAGE message is used to request another friendship with a user. In this case, the Subject header field has the value of "Friendship-Request". The CGADSP is included in the message body to allow the receiver to verify the validity of the CGA and the digital signature. Upon receiving the 200 OK message, which is the response to a MESSAGE message, an entry in the friends list is created. However, if the friendship request is rejected, a "SIP 406 Not Acceptable" response is sent to the requester and no further action is required.

2. To look for a friend, a MESSAGE message is used with a Subject

header field of "Friend-Lookup". The SIP URI of the person that the requester is looking for is included in the body of the message. An example of a MESSAGE message that is a friend lookup request is shown in Figure 5.8. A 200 OK is the response to this message whether the person is a friend of the responder or not. Generally speaking, a friend lookup will take a place if an INVITE message or a friendship request is received from a nonfriend. In both situations a minimum of one positive response is required to accept the call or the friendship. However, it is up to the user to accept the call or the friendship even if no positive response is received.

3. A MESSAGE message can also be used to recommend a friend to another friend, and the Subject header field will have the value of "Friendship-Recommendation". The CGA address of the recommended friend and the CGADSP are included in the message body to allow the receiver to verify the CGA of the recommended friend.

User Agent Server Behaviour

If the SIP user agent server (UAS) receives a SIP MESSAGE message, then there are three possibilities as to what could take place:

1. The MESSAGE message is from a blocked friend and will be ignored.

2. The sender of the message is a friend. First, the verification of the CGA and signature should be successful. Second, the subject of the message should be verified to know the purpose of the message. If the message is to recommend a friend, the user should be notified and he/she might decide to send a friendship request to the recommended friend. A 200 OK message is sent if the request is a friend lookup. If the friend is found in the friends

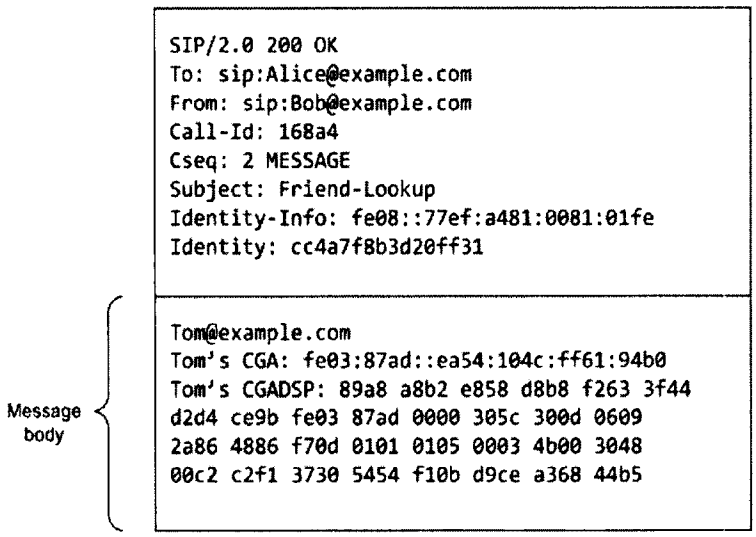

Figure 5.9: Response to Friend lookup request (simplified)

list of the UAS then the CGADSP is included in the body of the 200 OK message, as shown in Figure 5.9. Also, if the friend is not found in the friends list then a 200 OK message is sent withan empty body.

3. The SIP MESSAGE message could be from a non-friend. In this case, if the request is not a friendship request, it should be ignored. However, if the message is a friendship request, a 200 OK message should be generated to accept that friendship request. Before generating the 200 OK message the UAS may do a friend lookup to see if the requester is known to the UAS. If the friendship request is rejected because none of the UAS's friends know the requester, or for any other reason, then a 406 NOT ACCEPTED message is sent instead. Before sending this message, the user must be notified since he/she might accept the friendship even if known of his/her friends knows the requester.

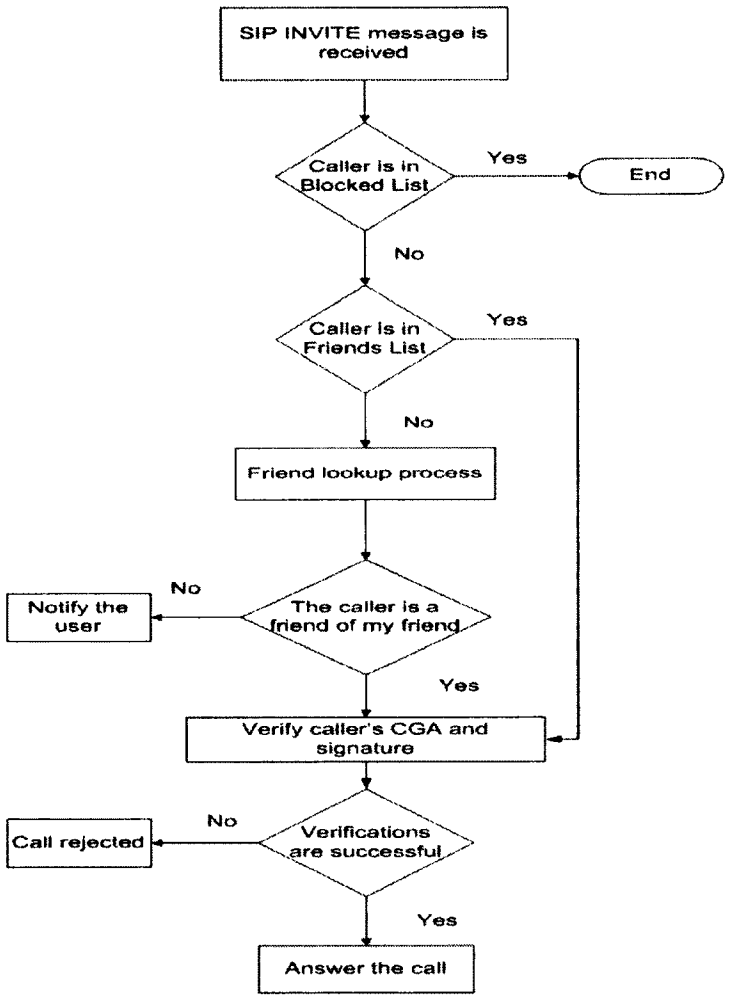

Figure 5.10: Flow chart for receiving a SIP INVITE message

Figure 5.10 presents the steps followed when the SIP user agent server receives a SIP INVITE message, and they are as follows:

1. Check if the caller is in the Blocked list. Reject the massage if the caller is blocked; continue otherwise.

2. Check if the caller is in the friends list. If so, go to step 6. If the caller is not in the friends list nor the blocked list, then a friend lookup procedure will take place, where a SIP MESSAGE message is sent to all or some of the friends in the friends list. To avoid generating extreme overhead from asking all the friends at one time,

3.friends are asked one at atime, and only if that friend responds negatively will another friend be asked. In this case the number of messages is reduced, which also significantly reduces the overhead.

4. If one of the user's friends sends positive feedback, the CGA of the caller and signature are verified. The verification is based on the CGADSP that is included in the response message of the friend lookup. If the verification succeeds, the call is accepted by sending a 200 OK message and exits; otherwise, a 406 NOT ACCEPTED is sent before exiting.

5. If none of the callee's friends knows the caller then the user is notified and if he/she decided not to accept the call, a 406 NOT ACCEPTED message is sent before exiting.

6. Validate the CGA using the CGA data structure parameters stored in the friends list. If the CGA is valid, continue; otherwise, the request is from an illegitimate user or attacker and the call should be rejected.

7. Verify the digital signature as described in section 5.1.3.

8. If the signature is valid, accept the call by sending a 200 OK message;

otherwise, an attacker has subverted the message's integrity and it must be rejected.

Chapter 6

Evaluation

This chapter provides a discussion about the overhead, delay, and complexity of the proposed security extension. Also, the implementation and simulation setup of the extension are discussed. The performance of the security extension is evaluated and compared with the non-secure SIP in ad hoc networks, which is the case where the security extension is not activated.

6.1 Discussion

The new security extension introduces additional overhead as follows:

1. Each SIP message has two additional header fields: "Identity" and "Identity-Info".

- The Identity header field contains the digital signature of the message. As per CGA specification, RSA keys are used. Accordingly, the RSA digital signature algorithm is used to sign SIP messages. The size of

an RSA digital signature depends on the size of the key. For instance, using a 1024-bit public key generates a 1024-bit digital signature.

- The Identity-Info header field contains the CGA, which is a 128-bit address. Therefore, each SIP message introduces an additional 1152 bits when a 1024-bit public key is used.

2. The friend lookup procedure results in additional overhead. It is carried out by sending a MESSAGE message, and getting a 200 OK message in response. The overhead depends on the number of transmitted messages required to authenticate a user. More details about this procedure are ifound later in this chapter.

3. All SIP MESSAGE messages carry the URI of the caller in the body; however, not all responses to MESSAGE messages contain this URI. If the user can authenticate the caller, he body of 200 OK message contains the URI of the caller along with the CGA and the data structure parameter of the caller.

The proposed security extension introduces an additional delay because for all SIP messages, the verification of the CGA and digital signature of the message both introduce some delay. Also, the friend lookup procedure delays the call response.

6.1.1 Memory Complexity

In a network of n nodes, each node has a friend list of average size x. Since each node needs, on average, x entries to store the friends list of size x, the memory complexity required to store the friend list in each node is $O(x)$.

6.1.2 Friend Lookup Complexity

The number of messages transmitted during the procedure of friend lookup depends on the number of entries in the friend list and the average number of friends per node in the network. The best case requires only one MESSAGE message and one "200 OK" response for authentication. The message complexity for this best case friend lookup process is $O(1)$. However, this usually requires the network to be a very connected network. A very connected network is a network where the average number of friends per node is more than 50% of the total population. In contrast, the worst case requires sending a SIP MESSAGE to every entry in the friend list. The worst case occurs if the network is not connected and one or more nodes have a high number of entries. For example, in a network of 50 nodes such that on average each node has 10 friends, a worst case could occur if the node that is looking for a friend has 20 friends and none of them can authenticate the caller. Message complexity in the worst case friend lookup process, where n represents the number of entries in friend list, is $O(n)$.

6.1.3 Registration Complexity

In the broadcast framework of SIP, the REGISTER message is sent using broadcast, and any node that receives it responds with a 200 OK message. In a network of n nodes,each node will send a REGISTER message usong broadcast, resulting in n REGISTER messages. For each REGISTER message there will be n 200 OK response messages. Hence, message complexity during registration, when the new security extension is not used, is $O(n^2)$.

However, the new extension reduces the registration overhead by introducing the concept of a friend list. There is a positive relationship between registration overhead and the average number of friends per node; more friends results in a greater registration overhead. The number of REGISTER messages is static in all cases n , but the number of response messages depends on the average number of friends per node. Message complexity of registration in this case, where x is the average number of friends, is $O(nx)$.

6.2 Simulation

6.2.1 Implementation

The performance of the new extension is evaluated using Network Simulator 2 (NS2) [6] under Linux. Prior's previously proposed implementation of SIP in NS2 [38] is not applicable to ad hoc networks since it requires centralized proxies and servers. To implement the new extension, Prior's module is used and modified to fulfil the distributed nature of ad hoc networks. The broadcast framework described in Section 3.2 is chosen to enable SIP in ad hoc networks. Basically, SIP servers and proxies are eliminated and replaced by a local cache that stores the binding information of users. Also, Prior's module doesn't support instant messaging, so the SIP extension of instant messaging is implemented in the new framework. Finally, the proposed new extension implements a security aspect and integrates social networking to SIP. However, not all new services are implemented. Add, block, unblock, delete, and recommend friend are not yet implemented. CGA generation and verification are implemented using static IPv6 addresses.

6.2.2 Simulation Setup

The nodes in all scenarios are static, i.e. node mobility is not supported. Every scenario is composed of 50 nodes distributed randomly in a grid of 250 x 250 meters. All scenarios run for 1000 seconds, where the first 300 seconds of the simulation are reserved for node registration and no calls are allowed. All calls in all scenarios are chosen randomly.

There are three categories of scenario based on the call rate:

1. One call per second, where on average there are 600 call attempts.

2. One call every two-seconds, where on average there are 300 call attempts.

3. One call every five-seconds, where on average there are 120 call attempts.

In all scenarios, the call is terminated after 0.5 seconds of initiation because from observation 0.4 seconds is the maximum call setup delay. Hence, whether the call is accepted or not, a BYE message is sent after receiving a response. Also, all calls are between non friends.

Each category has nine subcategories based on the average number of friends per node. To illustrate, one subcategory disables the proposed security extension and contains no friends, while the other eight subcategories use friends and the extension. The examined networks, where security is involved, have on average 5, 10, 15, 20, 25, 30, 40, and 50 friends per node. Friends are normally and randomly distributed among the nodes. In the case that the network has no friends, there is no security countermeasure and the social network paradigm is not taken in consideration. Networks with 5 and 50 friends per node are extreme cases, where the network is slightly connected

or extremely connected respectively. Networks of 25, 30, and 40 friends per node are very connected because the percentage of friends each node has is more than 50% of the population. In each subcategory 100 scenarios are run and averages are calculated for each scenario and recorded. The averages of all 100 runs in each subcategory is calculated and a 95% confidence interval is achieved. There is a total of 900 subcategory scenarios, so 2700 scenarios are examined in total.

6.3 Performance Metrics

Three performance metrics are considered in our simulations. The costs of the new extension in terms of overhead, delay, and number of unauthenticated calls. Overhead per call is evaluated based on initiation, acceptance and termination of a call. In addition, total overhead is calculated by accumulating the size of all messages in a scenario. Moreover, a distinction is made between total, registration, and friend lookup overheads. While total overhead is the overhead resulting from all SIP messages, the registration overhead is the overhead resulting from the registration phase, which occurs in the first 300 seconds of every scenario. The REGISTER message and its 200 OK response message are considered for evaluating the registration overhead. For friend lookup overhead, only the size of the MESSAGE message and its 200 OK response messages is accumulated to measure the cost of the new authentication mechanism. Both the total friend lookup overhead and the per-call friend lookup overhead are evaluated. In terms of delay, call setup delay is evaluated at both sides, that is, from the moment the

INVITE message is sent at the caller side to the moment the ACK message is received at the callee side. It is also important to consider the number of unauthenticated calls. Unauthenticated calls are rejected by the callee because a friend lookup process failed to find a user who can authenticate the caller. In addition to the number of unauthenticated calls, the percentage of unauthenticated calls is also evaluated. For all performance metrics, the average of 100 scenarios is calculated per subcategory to achieve a 95% confidence interval.

6.4 Simulation Results

Figure 6.1 represents the overhead resulting from initiating, accepting and terminating a call. This overhead is not affected by the call rate. Without the proposed extension, this overhead includes INVITE, 200 OK, ACK and BYE messages. As expected, the cost of security creates a greater overhead; the least overhead occurs when there is no security and no social networking involved. This is because there is no friends lookup process required to answer the calls. Also, message size is smaller because the CGA, CGA-DSP, and digital signature are not included in any of the messages. Without the security extension the overhead is 1 kB per call. In contrast, when the new security extension is used, the overhead is generally between 3 to 4 kB of traffic per call. The additional overhead comes from two factors. First, more messages are required to accept the call. Since all calls are coming from non friends, a friend lookup procedure is carried out via a MESSAGE message and its 200 OK response. Second, the message size is greater because additional

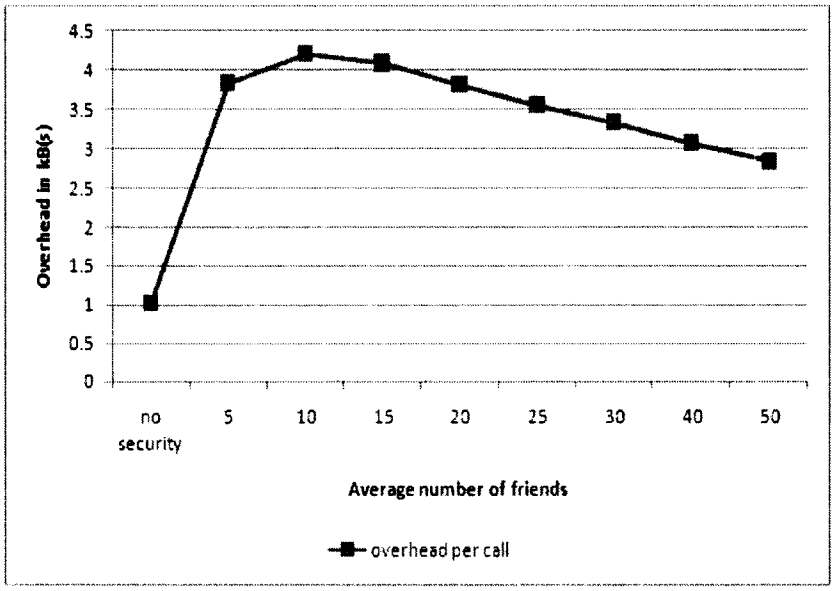

Figure 6.1: Overhead per call

header fields are included in all messages, and some messages carry CGADSP in their payload. There is an inverse relationship between the number of friends and the overhead. The highest overhead is in the cases where there are on average 5, 15, 10, and 20 friends per node respectivley. This is a result of the friend lookup process, where it is likely to go through most or all entries in the friend list to find an authenticator to the caller. Also, the less friends the node has the less probable it is that it will find an authenticator. This explains why the overhead is greater in the cases where the average number of friends is lower. The very connected networks, with an average of 25, 30, 40 and 50 friends per node, generate less over head than less connected networks.

Figure 6.2 represents the total overhead that results from all SIP messages. Surprisingly, comparing the overhead with and without the security

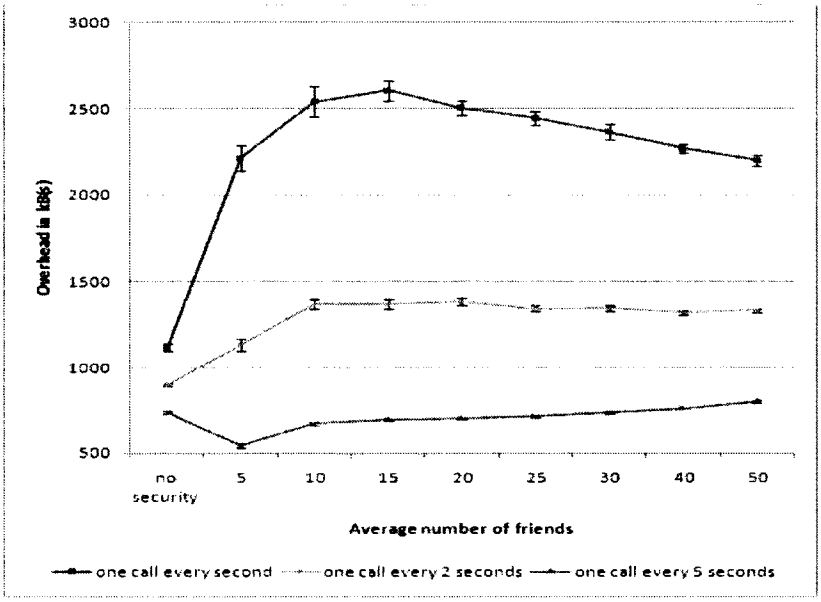

Figure 6.2: Total overhead

extension with a call rate of one call every five-seconds, the overhead with the security extension is lower in most subcategories. The lowest overhead occurs when the average friends per node is five. Also, if the network is not connected, the overhead when using the extension is less than when the extension is not involved. In the cases where the network is very connected, there is almost no difference between overheads with and without security. However, as the call rate increases, the overhead when using the security extension increases dramatically. Without the security extension the overhead is close in all call rates. This is because more friend lookup processes are required. Each call requires one friend lookup procedure; this explains why the overhead increases dramatically when the call rate is high.

The majority of the overhead is coming from two processes: registration and friend lookup. Figure 6.3 evaluates the process of registration. Regis-

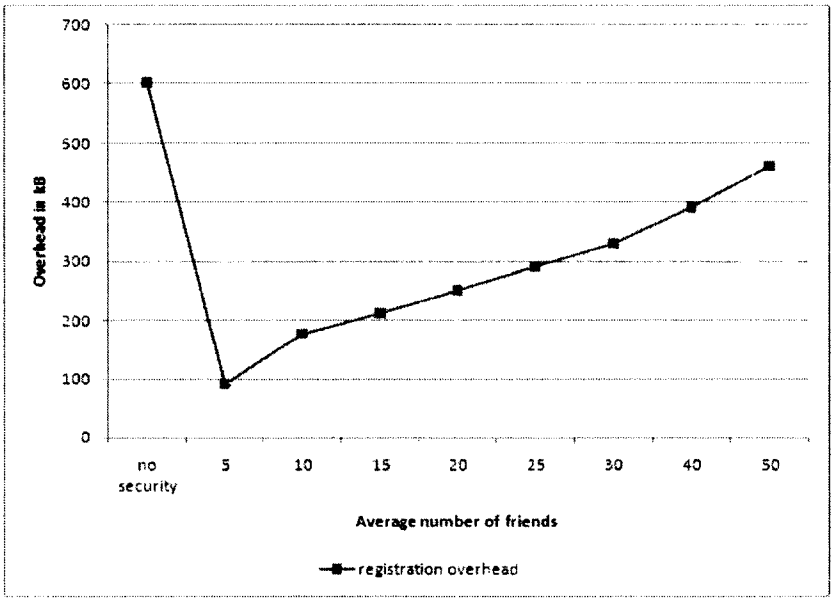

Figure 6.3: Registration overhead

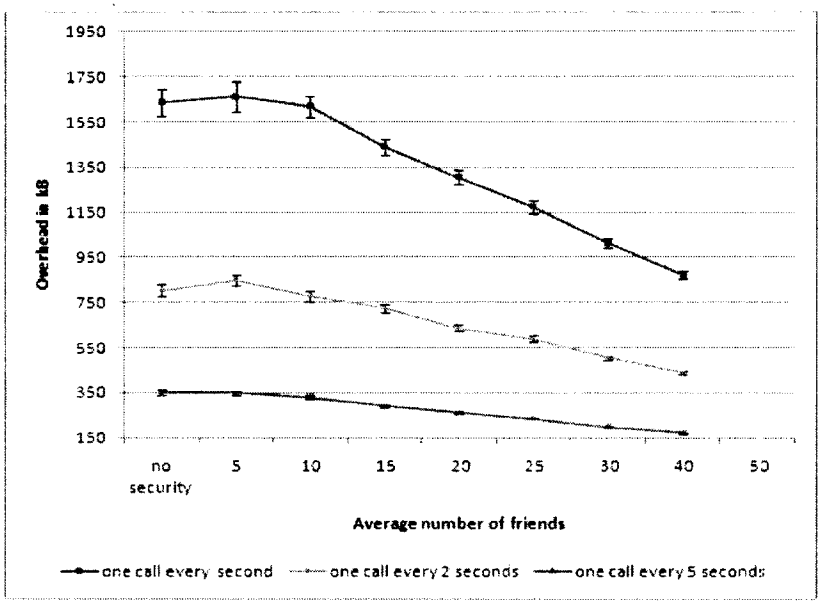

Figure 6.4: Friend look-up overhead

tration overhead is not affected by the call rate, but it does depend on the average number of friends. Having more entries in a friend list resulted in a greater overhead. As shown in the figure, registration overhead increases as the average number of friends increases. When the average number of friends per node is 5, the registration overhead is at the minimum; in this case, on average 5 friends will respond to each REGISTER message. In contrast, an average of 50 friends per node will respond to each REGISTER message if the average number of friends in the network is 50. However, the highest registration overhead occurs in the case where the new extension is not used. This is because all nodes in the network respond to all broadcasted REGISTER messages, maximizing the potential traffic.

Friend lookup overhead depends on the average number of friends per node in the network and on the call rate. Obviously, more calls result in more friend lookup, and thus more overhead. There is an inverse relationship between the average number of friends per node and the friend lookup overhead. To illustrate, if the network is very connected, i.e. the average number of friends is high, less MESSAGE messages are required to find a node that can authenticate the caller. As shown in figure 6.4 the overhead decreases as the average number of friends is increased. In a very connected network, the friend lookup overhead is less than in less connected ones. In addition, in figure 6.3, an interesting observation is made. For the call rate of one call every five-seconds, where approximately 120 calls are made, the friend lookup overhead is greatest with 5, 10 and 15 friends per node, and this is still less than the registration overhead without the security extension. This means the cost of authenticating a caller using the extension, is

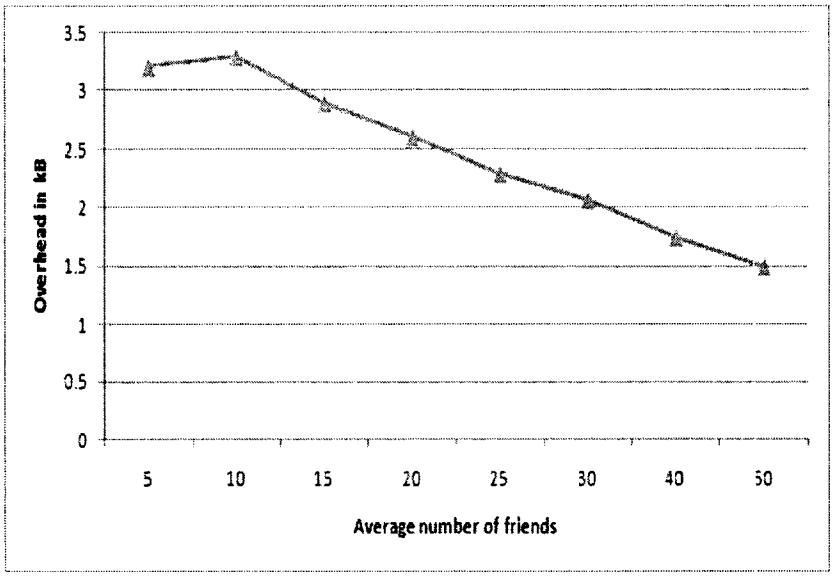

Figure 6.5: Friend lookup overhead per call

less than the registration process without the extension.

Figure 6.5 shows the overhead that results from the friend lookup procedure per call. It is not affected by the call rate. The highest overhead occurs when there are on average 5 to 10 friends per node. It is more than 3 kB per call. These networks are not connected networks, and 3 kB is the cost of going through almost all the entries in the friend list until an authenticator of the caller is found. However, the overhead decreases as the average number of friends per node increases. The lowest overhead is approximately 1.5 kB and it occurs if the network is an extremely connected network where almost every node is a friend of every other node. In this case, one or two MESSAGE messages are enough to find an authenticator of the caller.

As with many performance metrics, call setup delay depends on the average number of friends per node. Figure 6.6 shows that it takes approximately

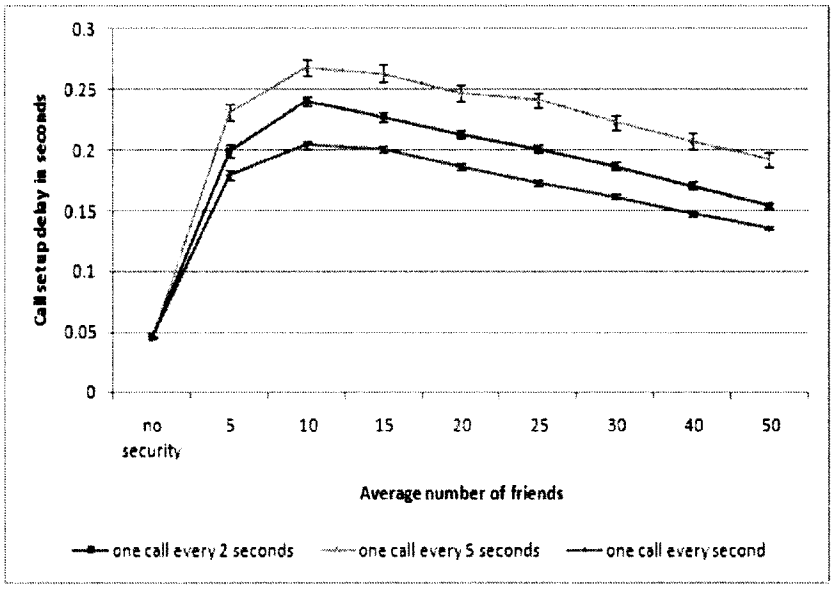

Figure 6.6: Call setup delay

50 milliseconds (ms) to answer a call in the case where the new extension is not used. With the new extension, the best call setup delay occurs when the network has an average of 50 friends per node, reaching 150-200 ms. Two operations are causing this delay in the new extension: the verification of CGA and digital signature and the friend lookup operation. The worst call setup delay was approximately 250 ms. Although the delay in the new extension is much higher than without it, it is still acceptable at less than half a second. The delay is affected by the call rate most probably because an on-demand routing protocol is used. The network is active with a rate of one call per second, but it is less active with one call every two seconds and one call every five seconds respectively. This could be a reason why the delay increases as number of calls decreases.

Figure 6.7 shows the percentage of unauthenticated calls. It is not af-

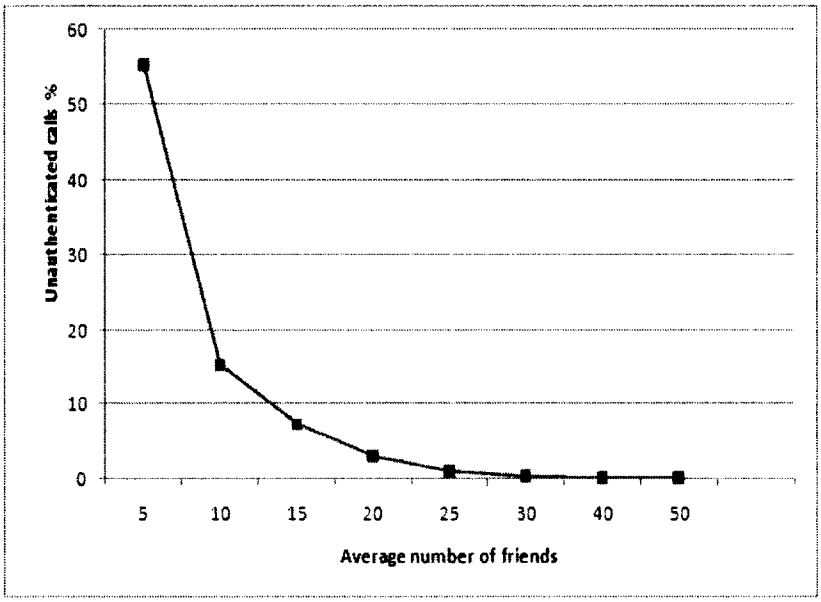

Figure 6.7: Percentage of unauthenticated calls

fected by the call rate; it depends only on the average number of friends in the network. In less connected networks, where the average number of friends is relatively low, the number of unauthenticated calls is high because the callee fails to find a node that can authenticate the caller. However, unauthenticated calls decrease dramatically as the network gets more connected. There are almost no rejected calls due to a lack of authentication in the cases where the average number of friends is equal or greater than 50% of the population.

Chapter 7

Conclusion and Future Work

This chapter summarizes the contributions outlined in this thesis and the simulation results. Also, the direction of future work is examined.

7.1 Conclusion

In this thesis, SIP is extended to enhance the security in ad hoc networks. Authentication and message integrity are provided with the aid of CGA and the social network paradigm. A CGA binds the public key of the address owner to his/her IP address. This technique guarantees un-spoofable addresses. However, because the public key used is not authenticated by a third party, an attacker can use a malicious public key to generate a valid CGA. This causes a security risk in the initial phase when a new node joins a network. To solve this problem, the idea of social networking is applied. Friendships between the participants exist before they join the network, so CGAs can be exchanged between friends in some offline way first. Each user

keeps a list of his/her friends along with their CGAs and CGADSP. All SIP messages are signed using the private key of the sender to achieve message integrity. The CGA of the originator and digital signature of the message are verified based on the CGADSP stored in the friend list to assure sender identity and message integrity. Due to the integration of social networking, new services are added to SIP. The new services include: add, delete, recommend, lookup, and block/unblock friends.

Simulations have been conducted to evaluate the performance of the proposed security extension. Simulations show that the traffic overhead resulting from the registration process is reduced when using this extension. However, traffic overhead caused by making calls is increased. As the average number of friends per node increases, call overhead decreases. Similarly, call setup delay decreases as the node becomes more social and adds more friends; however, the call setup delay when using the security extension is greater than without it, but it is not larger than 300 milliseconds, which is considered acceptable.

7.2 Future Work

As future work, we would be interested in introducing a mobility variable into the simulation scenarios. In this thesis, all simulations have been run in static scenarios, but evaluating the performance of the proposed security extension with varying node mobility would affect the performance metrics and provide more insight into its potential. A mobility management protocol, such as mobile IPv6, would be required. Furthermore, not all services introduced

in this thesis were implemented in the simulations. At this point only the "friend lookup" service has been implemented. Implementation of the add, delete, block/unblock, and recommend friend will be considered for future work.

In therms of the security extension itself, while it provides authentication and message integrity to a certian degree, there exist other types of attack that need to be addressed. For instance, it is important to protect message content from passive attacks, where eavesdropping takes place without altering the message content. In this case, message confidentiality must be maintained. Encrypting SIP messages would possibly enhance the security against this type of attack; however, encrypting the whole SIP message is not practical, because some header fields, such as "To" and "Via", are required to route the messages. In this case, encrypting only selected SIP header fields would be better strategy.

Elliptic Curve Cryptography (ECC) is a promising encryption strategy for ad hoc networks. It requires a shorter key size than RSA keys to achieve the same level of security. Cheneau has already compared the performance of CGA generation and verification using RSA and ECC [18]. He points out that it takes less time to generate and verify CGAs using ECC keys. However, the Elliptic Curve Digital Signature Algorithm (ECDSA) verification is slower than the RSA Digital Signature Algorithm (RSADSA). In future work, a comparison between the extension using ECC and ECCDSA and the extension using RSA and RSADSA would determine if the extension could benefit from the new wncryption. It is anticipated that using ECC instead of RSA would optimize the performance of the new extension as it would

reduce the traffic overhead, but it might increase the call setup delay.

Bibliography

[1] Aircrackng. [Online]. Available: http://www.aircrack-ng.org/doku.php.

[2] istumbler. [Online]. Available: http://istumbler.net/.

[3] Kismac. [Online]. Available: http://mac.free.comprolive.com/2007/10/kismac-stumblerscanner.html.

[4] Kismet. [Online]. Available: http://www.kismetwireless.net/.

[5] Macstumbler. [Online]. Available: http://www.macstumbler.com/.

[6] The network simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/.

[7] Wireshark. [Online]. Available: http://www.wireshark.org/.

[8] T. Aura. Cryptographically generated addresses (CGA). IETF RFC 3972, March 2005.

[9] M. Bagnulo and J. Arkko. Cryptographically generated addresses (CGA) extension field format. IETF RFC 4581, October 2006.

[10] K. Balov, K. Kawagoe, and T. Nishimura. SIP deployment in integrated mobile ad hoc networks: Centralized and quasi-decentralized approaches. In *11th International Conference on Advanced Communication Technology, ICACT 2009*, volume 01, pages 203 –207, 2009.

[11] Nilanjan Banerjee, Arup Acharya, and Sajal Das. Enabling SIP-based sessions in ad hoc networks. *Wireless Networks*, 13:461–479, 2007. 10.1007/s11276-006-9200-8.

[12] Michel Barbeau. Wireless security in the home and office environment. Technical Report TR-10-12, School of Computer Science, Carleton University, May 2010.

[13] Michel Barbeau. Point-to-point voice over ad hoc networks: A survey. *Pervasive and Mobile Computing*, (0):–, 2011.

[14] David A. Bryan and Bruce B. Lowekamp. Sosimple: a SIP/SIMPLE based P2P VOIP and IM system. White paper, Computer Science Department, College of William and Mary, Williamsburg, VA, November 2004.

[15] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. Session initiation protocol (SIP) extension for instant messaging. IETF RFC 3428, December 2002.

[16] Zhen Cao, Hui Deng, Yuanchen Ma, and Po Hu. Integrating identity based cryptography with cryptographically generated addresses in mobile ipv6. In *Proceedings of the 2007 international conference on Compu-*

tational science and Its applications - Volume Part II, ICCSA'07, pages 514–525, Berlin, Heidelberg, 2007. Springer-Verlag.

[17] Marcel C. Castro and Andreas J. Kassler. Optimizing SIP service provisioning in internet connected MANETs. In *International Conference on Software in Telecommunications and Computer Networks, SoftCOM 2006.*, pages 86 –90, oct 2006.

[18] Tony Cheneau, Aymen Boudguiga, and Maryline Laurent. Significantly improved performances of the cryptographically generated addresses thanks to ecc and gpgpu. *Computers and Security*, 29(4):419 – 431, 2010.

[19] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). IETF RFC 3626, October 2003.

[20] Aytunc Durlanik and Ibrahim Sogukinar. SIP authentication scheme using ECDH. *World Academy of Science, Engineering and Technology*, 8:350–353, 2005.

[21] S. El Sawda and P. Urien. Sip security attacks and solutions: A state-of-the-art review. In *Information and Communication Technologies, 2006. ICTTA '06. 2nd*, volume 2, pages 3187 –3191, 0-0 2006.

[22] Ali Fessi, Heiko Niedermayer, Holger Kinkelin, and Georg Carle. A cooperative SIP infrastructure for highly reliable telecommunication services. In *Proceedings of the 1st international conference on Principles, systems and applications of IP telecommunications*, IPTComm '07, pages 29–38, New York, NY, USA, 2007. ACM.

[23] Dimitris Geneiatakis and Costas Lambrinoudakis. A lightweight protection mechanism against signaling attacks in a SIP-based VoIP environment. *Telecommunication Systems*, 36:153–159, 2007. 10.1007/s11235-008-9065-5.

[24] Kyusuk Han, Chan Yeob Yeun, and Kwangjo Kim. Design of secure VoIP using ID-based cryptosystem. In *SCIS 2008, The 2008 Symposium on Cryptography and Information Security*, 2008.

[25] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. IETF RFC 4728, February 2007.

[26] A. Johnston. SIP, P2P and internet communications. IETF Internet Draft draft-johnston-sipping-p2p-ipcom-00.txt, January 2005.

[27] Angelos D. Keromytis. Voice over ip: Risks, threats and vulnerabilities. In *In: Proceedings of the Cyber Infrastructure Protection (CIP) Conference*, 2009.

[28] H. Khlifi, A. Agarwal, and J.-C. Gregoire. A framework to use SIP in ad-hoc networks. In *Canadian Conference on Electrical and Computer Engineering. IEEE CCECE*, volume 2, pages 985 – 988, May 2003.

[29] S. Leggio, J. Manner, A. Hulkkonen, and K. Raatikainen. Session initiation protocol deployment in ad hoc networks: a decentralized approach. In *2nd International Workshop on Wireless Ad-hoc Networks (IWWAN)*, May 2005.

[30] S. Leggio, J. Manner, and K. Raatikainen. A secure SIP-based instant messaging and presence framework for ad-hoc networks. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1 –6, dec 2006.

[31] Yi-Pin Liao and Shuenn-Shyang Wang. A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves. *Computer Communications*, 33(3):372–380, 2010.

[32] J. Manner, S. Leggio, and K. Raatikainen. An internet SIP gateway for ad-hoc networks. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON '06*, volume 3, pages 740 –745, 2006.

[33] P. Matthews and B. Poustchi. Industrial-strength P2P SIP. IETF Internet Draft draft-matthews-sipping-p2p-industrial-strength- 00.txt, February 2005.

[34] Mick O'Doherty. Pico SIP. IETF Internet Draft draft-odoherty- pico-sip-00.txt, January 2001.

[35] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. IETF RFC 3561, July 2003.

[36] J. Peterson and C. Jennings. Enhancements for authenticated identity management in the session initiation protocol (SIP). RFC 4474, IETF, August 2006.

[37] Larry L. Peterson and Bruce S. Davie. *computer networks: a system approach.* Morgan Kaufmann, 4th edition, 2007.

[38] Rui Prior. ns-2 network simulator extensions. [Online]. Available: http://www.dcc.fc.up.pt/ rprior/ns/.

[39] Rui Prior. *Scalable Network Architectures Supporting Quality of Service.* PhD thesis, Faculty of Sciences of the University of Porto, 2007.

[40] Jared Ring, Kim kwang Raymond, Choo Ernest Foo, and Mark Looi. A new authentication mechanism and key agreement protocol for SIP using identity-based cryptography. In *AusCERT R and D Stream*, pages 57–72, May 2006.

[41] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. IETF RFC 3261, June 2002.

[42] Taruni Seth, Christian Huitema, Kun-Min Yang, and Huai-An P. Lin. VoIP signaling performance requirements and expectations. IETF expired internet draft, October 1999.

[43] Patrick Stuedi, Marcel Bihr, Alain Remund, and Gustavo Alonso. SIPHoc: efficient SIP middleware for ad hoc networks. In *Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware*, Middleware '07, pages 60–79, New York, NY, USA, 2007. Springer-Verlag New York, Inc.

[44] Skype Technologies. Skype. [Online]. Available: http://www.skype.com.

[45] International Telecomunications Union. ITU-T recommendation H.323: Pecket-based multimedia communication systems. International Telecomunications Union, December 2009.

[46] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service location protocol. The Internet Society Network Working Group, 1997. Request for Comments: 2165.

[47] Fengjiao Wang and Yuqing Zhang. A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography. *Computer Communications*, 31(10):2142–2149, 2008.

[48] W. Werapun, A.A. El Kalam, B. Paillassa, and J. Fasson. Solution analysis for SIP security threats. In *International Conference on Multimedia Computing and Systems, ICMCS '09*, pages 174 –180, 2009.

[49] Liufei Wu, Yuqing Zhang, and Fengjiao Wang. A new provably secure authentication and key agreement protocol for SIP using ECC. *Computer Standards and Interfaces*, 31(2):286–291, 2009.

[50] Chou-chen Yang, Ren-chiun Wang, and Wei-ting Liu. Secure authentication scheme for session initiation protocol. *Computers Security*, 24(5):381–386, 2005.

[51] Eun-Jun Yoon, Kee-Young Yoo, Cheonshik Kim, You-Sik Hong, Minho Jo. and Hsiao-Hwa Chen. A secure and efficient SIP authentication scheme for converged VoIP networks. *Computer Communications*, 33:1674–1681, September 2010.