

# VoIP Security

Piero Fontanini



**Master's Thesis**

**Master of Science in Information Security**

**30 ECTS**

**Department of Computer Science and Media Technology**

**Gjøvik University College, 2008**

Avdeling for  
informatikk og medieteknikk  
Høgskolen i Gjøvik  
Postboks 191  
2802 Gjøvik

Department of Computer Science  
and Media Technology  
Gjøvik University College  
Box 191  
N-2802 Gjøvik  
Norway

## Abstract

VOIP or Voice Over Internet Protocol is a common term for phone service over IP based networks. There are much information about VoIP and some of how VoIP can be secured. There is however no standard for VoIP and no general solution for VoIP Security. The security in VoIP systems today are often non existing or in best case weak and often based on proprietary solutions.

This master thesis investigates threats to VoIP system and describes existing alternatives for securing VoIP as it is today.

Security requirements for VoIP system are identified and a new security solution is proposed that will increase the security in VoIP system. The proposed solution secures the SIP signaling in VoIP and focus on end to end security.

The requirements on the new solution focus on confidentiality, integrity, availability and that the security solution is user friendly. We will here propose an zero effort security solution that easily can be used by non technical users.

A fully function prototype of a secured SIP phone is developed to test if suggested design works as assumed.



## Preface

I would like to thank my supervisor Dr. Chik How Tan for helping me in my work and for being a supportive contact. I would also like to thanks Jens-Are Amundsen for interesting discussions and answering many questions about Identity Based Encryption.

Piero Fontanini, 2008/10/31



## Contents

Abstract . . . . .	iii
Preface . . . . .	v
Contents . . . . .	vii
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Topic covered by the project . . . . .	1
1.2 Keywords . . . . .	1
1.3 Problem description . . . . .	1
1.4 Justification, motivation and benefits . . . . .	2
1.5 Research questions . . . . .	2
1.6 Contributions . . . . .	4
1.7 Choice of methods . . . . .	4
1.7.1 Case study . . . . .	4
1.8 Abbreviations . . . . .	6
<b>2 Literature survey . . . . .</b>	<b>7</b>
2.1 Security in VoIP . . . . .	7
2.2 Session Initiation Protocol, SIP . . . . .	8
2.2.1 Introduction to SIP . . . . .	8
2.2.2 HTTP Digest Authentication . . . . .	14
2.2.3 SIPS . . . . .	15
2.2.4 IPSec . . . . .	16
2.2.5 S/MIME . . . . .	16
2.3 Identity Based Cryptography . . . . .	18
2.3.1 An Short introduction to the Boneh-Franklin IBE system . . . . .	19
2.3.2 Identity-based signature schemes . . . . .	20
2.4 Security issues . . . . .	21
2.4.1 VoIP threats and Vulnerabilities . . . . .	21
<b>3 Securing the VoIP Signaling . . . . .</b>	<b>27</b>
3.1 Requirements . . . . .	27
3.2 Choice of technology . . . . .	27
3.3 Usage of S/MIME in SIP . . . . .	28
3.3.1 Providing integrity . . . . .	28
3.3.2 Providing integrity and confidentiality . . . . .	30
3.3.3 Removal of Call-ID . . . . .	32
3.3.4 Negotiating SMIME capabilities . . . . .	32
3.4 Identity Based Cryptography and S/MIME . . . . .	33
3.4.1 Usage of IBE in S/MIME . . . . .	33
3.4.2 Identity-based signature schemes in S/MIME . . . . .	34
3.5 System architecture . . . . .	34
3.5.1 Protocol design . . . . .	35
3.5.2 User friendly . . . . .	38

4	Prototype . . . . .	39
4.1	Development plan . . . . .	39
4.2	Prototype design . . . . .	39
4.3	SIP, Softphone and Proxy software . . . . .	40
4.4	Usage of Boneh-Franklin IBE algorithm . . . . .	41
4.5	Development of IBE software . . . . .	42
4.6	S/MIME replacement . . . . .	43
4.7	Changes made to soft phone software . . . . .	44
4.8	Test scenarios . . . . .	45
4.8.1	Test scenario 1: Proof of Concept . . . . .	47
4.8.2	Test scenario 2: Measuring signaling overhead . . . . .	47
4.9	Test result . . . . .	48
4.9.1	Result from test scenario 1 . . . . .	48
4.9.2	Result from test scenario 2 . . . . .	53
5	Discussion . . . . .	55
5.1	Requirements . . . . .	55
5.2	Test result . . . . .	57
6	Conclusion . . . . .	59
7	Further work . . . . .	63
	Bibliography . . . . .	65
A	Configuring the prototype . . . . .	69
B	Building the prototype . . . . .	73
C	Running the prototype . . . . .	75



# 1 Introduction

## 1.1 Topic covered by the project

VoIP or Voice Over Internet Protocol refers to transmission of voice over a IP based network. This is one of the most important emerging trends in telecommunications [1] today. For many end users of traditional fixed line telephone, lower cost and the greater flexibility is very attractive. As with many new technologies, VoIP introduces both advantages and disadvantages. An obvious advantage is that you with VoIP have the potential to lower your cost for local and long distance calls, as VoIP is routed over the Internet and no separate telephone network is needed. There is also potential for better quality of the speech as VoIP can provide higher bandwidth.

Traditional telephony provides a fairly robust solution that works most of the times. VoIP requires other scenarios to be considered as VoIP can be routed over Internet. The quality of the service can be uneven, network in use may be highly congested, information can be dropped or delays introduced by long distance calls. Another potential drawback/disadvantage is, as VoIP is a fairly new technology, the security in VoIP. VoIP traffic is in many cases routed on the same network as other network traffic, which exposes the voice for the same security risks as ordinary data traffic. Many providers of VoIP do not support encryption, making it possible to tap a line between communicating parts and listen to conversations. It also makes is possible to pick up tones sent from subscribers to a service providers, that may be a distributors of sensitive data.

## 1.2 Keywords

Information Security, VoIP, Voice over IP, SIP, Session Initiation Protocol, SIPS, TLS, IPsec, S/MIME, Elliptic curves, IBE, Identity Based Encryption.

## 1.3 Problem description

There are some main challenges with VoIP security today. Mainly because call setup information and actual "voice data" is transported on an open network like Internet. This section will give a brief introduction to the technology in use and short description of security risks this introduces.

An user need some kind of equipment or software to be able to make a VoIP call. This equipment can be a VoIP telephone, an converter that let you use an ordinary PSTN telephone or software on a computer that emulates an ordinary telephone. User equipment or software are often referred to as an User Agent (UA). A call can be made directly from one subscriber to another or through one or more proxies between communicating parts. A proxy often works as a gateway to other networks like the Internet and is also often responsible for authentication of local users. The proxy component is often referred to as a proxy, gateway or a User Agent Server (UAS). Figure 1 shows a simplified version of call between two User Agents.

The Session Initiation Protocol or SIP[2] is a protocol for setting up sessions between two or more communicating parts. SIP can for example be used to set up VoIP calls or

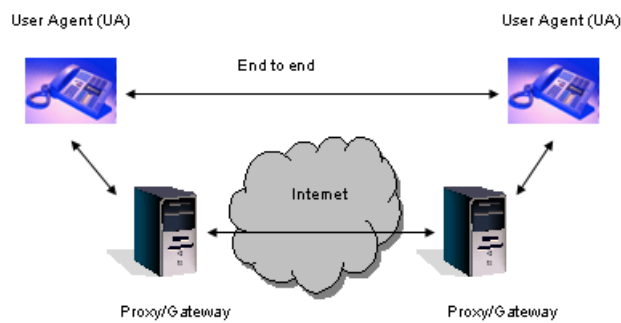


Figure 1: Voice Over IP

sessions for online games. SIP is an application layer protocol that can be transported with tcp/ip or udp/ip. SIP can be used on all ip based networks. SIP can however also be transported over the Internet and can be monitored, used, interfered with, or spoofed like any other IP communication. End users and VoIP providers can for example be exposed to Caller id spoofing. Caller id is used by many services like voicemail, automatic playback, emergency services (like 112), customer services and financial institutions to identify an end user. Can caller id for a VoIP user really be trusted by a service provider? and can an end user really trust the invoice from the VoIP provider if caller id can be abused?

The actual voice data is sent to the receiver by using the Real-time Transport Protocol (RTP) [3]. RTP packages are transferred in an IP package as other data on the Internet. This will of course make the speech (voice data) available to whoever taps the connection. As a result, speech data has to be encrypted in some way. This can for example be done with new profiles of RTP or by implementing security at a lower levels in the protocol stack.

A general introduction to VoIP can be found in [1][4].

#### 1.4 Justification, motivation and benefits

VoIP has quickly become a popular alternative to regular telephones as it today is both cheaper and give more flexibility. In some cases, services are provided for free by vendors like Skype and Ekiga.

As with many new technologies, the security aspect has not been fully investigated. Most people are not aware that their VoIP calls are transferred over the Internet in "clear text" or just using weak security solutions. The technology is not yet mature, but VoIP providers wanting to be able to compete need a solution today. These solutions are in general, not secure enough.

#### 1.5 Research questions

- How can the Signaling protocol used in VoIP be secured?
- Is confidentiality needed and if so, how can this be provided?
- How can integrity be provided?
- How can availability be assured in a VoIP system?

- How can a client be assured that message is secured from one phone to another?
- How can a recipient be assured that a message really is from the stated sender?
- Is it possible to introduce a "zero effort" security solution that can be used by non technical users?

## 1.6 Contributions

The MSc will focus on security for the Session Initiation Protocol (SIP). Security issues regarding H.323 that is an alternative to SIP will not be addressed.

The result should be an description of security issues with VoIP, and how these issues can be solved. There are probably many solutions to the problems that will be identified and the different approaches for securing data and the protocols involved has to be reviewed and compared. This MSc should focus on practical solutions for securing VoIP as it is today and in the the next couple of year, rather of proposing an "perfect" solution that is impossible to implement for VoIP providers. The questions addressed earlier in "Research questions" should, if possible, be answered.

## 1.7 Choice of methods

The research questions described in section 1.5 all needs to be studied in depth. There are probably several answers to each question where each answer results in a unique solution. A suggested solution to a problem has to be investigated and described in detail to be able to understand the result.

The research questions defined are quite general. There is also no easy way of getting discrete, measurable variables out of the data collection. This indicate that the methods to be used to find an answer, are of a qualitative nature.

The main research methods in qualitative research are action research, case study research, ethnography and grounded theory. In a case study, a particular individual, program, or event is studied in depth for a defined period of time [5]. The case study research method is assumed to be the best method for research on information systems and is the chosen one in this MSc. More information on qualitative research and case study in special can be found in [5] [6].

The research project is going to be divided into a theoretical part and a practical part. The theoretical part consist of definition of cases to be studied, literature review, data analysis and suggestion of a solution to solve the problems defined. The practical part consists of development of a prototype of the suggested solution where open source code is used as a base for the implementation.

### 1.7.1 Case study

Case study research is the most common qualitative method used when doing a research on a information systems. We will in this project use the case study approach to try to answer all the research questions identified. The different cases has to be identified and described in the initial phase of the project. The research will then be focused on these cases narrowing the problem area. This will be followed by a literature review, data collection and analysis.

#### Literature review

A literature review is a survey of important articles, books and other sources pertaining to your research topic. The literature review is done to get the necessary theoretical background to the subject. Different research reports of the chosen subject should be identified, read and evaluated. The review will hopefully also give new ideas for solutions to the problems specified.

Much work has already been done on VoIP security in general. This MSc will however focus on the SIP protocol and different security methods available. It is important to

identify relevant literature early in the process to be able to investigate the area in depth. Result from this work should be a suggestion of security solution that could be used in conjunction with SIP. This security solution is then going to be implemented in a prototype to investigate if the concept works or not.

The SIP specification [2] describes an authentication method (HTTP Digest authentication) that has to be investigated in detail to identify weaknesses. My hypothesis is that the suggested method don't provide sufficient security to the SIP protocol and that the solution contains several weaknesses.

The prototype will be based on an open source tool kit for SIP. The tool kit has to be identified and reviewed. This should be done by identifying different implementations and get the one best suited for the job.

#### Data analysis

The data analysis are in cases study often started in the data collection phase, i.e during the literature review. The data analysis consists of the following steps (taken from [5]):

- Organization of details about the case
- Categorization of data
- Interpretation of single instances
- Identification of patterns
- Synthesis and generalization

The outcome of the different steps is documented in the MSc thesis.

## 1.8 Abbreviations

VoIP	Voice over Internet Protocol
SIP	Session Initiation Protocol, an application layer signaling protocol
IETF	Internet Engineering Task Force, develops and promotes Internet standards
DoS	Denial of Service, makes a service unavailable to a user of a system
DDoS	Distributed Denial of Service
Jitter	Packets from the source will reach the destination with different delays
UA	User Agent that consists of a UAC and a UAS
UAC	User Agent Client
UAS	User Agent Server
IBE	Identity Based Encryption
PKG	Private Key Generator
CA	Certificate Authority

## 2 Literature survey

There are several areas that has to be investigated to cover the scope of this MSc. We will in this chapter show what has been done to secure VoIP and the components that VoIP consists of today. We will first identify relevant literature for the subject. We will then identify the research done in VoIP Security and identify weaknesses that still exists. This chapter should also identify potential answers to the research questions, relate information to these and identify research that should be done.

### 2.1 Security in VoIP

There are much information about VoIP and some of how VoIP can be secured. There is however no standard for VoIP and no general solution for VoIP Security. We will here give an status of the research done so far on VoIP security.

IP telephony is a technology in which IP networks are being used to transport voice data. Ofir Arkin states in his article [7] that security threats associated with IP Telephony is far greater than with regular telephone networks. He base this statement on the fact that VoIP uses the IP protocol to transfer signaling and voice and the fact that the same network can be used for both data and voice. There are technologies for separating data and voice networks virtually with virtual LANs (VLANs). The same network components are however used making it vulnerable to DoS or similar attacks. VoIP vulnerabilities are also more likely to be published than similar weaknesses in PSTN as it is a more accessible technology for most people. Arfin also states that VoIP clients contains more intelligence than ordinary PSTN telephones and that this will make these clients more likely to be compromised by an attacker. One must also assume that an VoIP call is transfered through different networks controlled by different entities making it hard to determine the security level from one end to another.

Availability are an important part of VoIP. The ordinary PSTN network has a availability of 99.999 % and people converting to VoIP expects no less. This can however be very difficult to achive as the availability depends on so many components in the network and many of them don't have any redundancy. Redundancy will, of course make the solution more expensive.

The research work should identify requirements for availability in VoIP network. A trade off between availability, user friendliness and security should then be done to identify a reasonable level of security. This should be considered as an input the the rest of the research.

The National Institute of Standards and Technology (NIST) has developed a recommendation that contains security considerations for Voice Over IP systems [1]. NIST states that establishing a secure VoIP network is a complex process, mostly because of the integration of voice and data in one network. NIST also presents a list of recommendations that a security architect should consider when setting up a VoIP network. These recommendations seems to be very sensible but tends to be on a very high level and very hard to implement.

Other important parts are the influence that security has on Quality of Service (QoS). As voice has to be transferred in real-time, it is important that the packages are transferred without too much delay and that the network is reliable and not loses packages on the way. Availability and quality of speech are here essential.

VoIP networks probably have a lot of unique vulnerabilities that can be exploited. [8] identifies and describes the following attacks on VoIP networks: Reconnaissance Attacks, Floods and Distributed Floods, Protocol Fuzzing, Misuse or Spoofing, Session Anomalies, Stealth Attacks and VoIP Spam. We will not go in to the details here, but a conclusion that can be drawn from this is that VoIP networks are at least as vulnerable as networks used for data and that there is no simple solution for securing VoIP. Additional complexity is also added as VoIP is a real-time service, that requires stable networks and immediate security responses to be able to provide high availability and the QoS that is expected.

## 2.2 Session Initiation Protocol, SIP

There are today no standard for signaling in VoIP. SIP, H.323 and different proprietary protocols are in use. It seems though that the proposed standard for SIP [2] is the emerging signaling standard for VoIP. The MSc will only deal with SIP as this is the most likely candidate to be used in the future.

One can expect that the security requirements are the same for VoIP as it is for PSTN today. SIP messages may contain information that a user or a server wishes to keep private [9]. It must also be possible to recognize unauthorized changes to a SIP message. Hence confidentiality and integrity should be added to SIP messages. Some of the research questions focus on confidentiality, integrity and availability in SIP and these questions should be fully investigated by the researcher.

The most severe threat in VoIP environment is probably the easy access to the communication channel [9]. SIP is text based and readable to humans making it vulnerable to spoofing, hijacking and message tampering. SIP can use TCP, UDP or SCTP as transport protocol and as a result of this, inherits the vulnerabilities in these protocols. Since the SIP message structure is a straight derivation of the HTTP request/response model, all security mechanisms available for HTTP can also be applied to SIP sessions [10].

The proposed SIP standard, RFC3261 [2], describes several features that can be used to secure SIP. They are all listed below with a short description of strength and weaknesses.

### 2.2.1 Introduction to SIP

This section is intended to give the reader a short introduction to VoIP signaling and the Session Initiation Protocol. An example of a basic call is shown with a call setup and tear down, user ids and network components needed are also identified. Readers already familiar with VoIP and SIP may skip this section.

SIP is used in VoIP as an application-layer signaling protocol for creating, modifying and terminating sessions between one or more participants. It also allows these participants to negotiate how they are going to communicate (also called session negotiation). The majority of these negotiations in systems running today result in voice sessions, but nothing prevents negotiating an Instant Messaging (IM) session or even an online game [11].



The following part identifies and shortly describes the main components involved in VoIP communication.

A *User Agent (UA)* is an endpoint in the infrastructure that terminates the SIP signaling. User Agents has two different parts, an User Agent Client (UAC) and an User Agent Server (UAS).

A *User Agent Client (UAC)* is typically an endpoint, like a VoIP phone where the SIP client can be in the firmware if external hardware is used or software in a computer if a softphone is used. The UAC initiates a SIP request to a User Agent Server (UAS).

A *User Agent Server (UAS)* is a host with an application responsible for receiving a SIP request from a UAC and return a response back to that UAC. The UAS may in some cases issue multiple responses to the UAC.

*Proxy Servers* are often used as mediators in VoIP and forwards requests to UAS, UAC or other proxies. A proxy server is often responsible for a domain that a client is registered to. A Proxy may enforce a policy and for example, verify that an user is allowed to initiate a call. A proxy can also interpret and if necessary, rewrites specific parts of a SIP request before forwarding it.

A *Redirect Server* is an UAS that generates responses (redirects) to requests it receives, redirecting a client to another resource. Redirect Servers allow users to temporarily change geographic location and still be contactable through the same SIP address.

A *Registrar Server* is a server that accepts register requests from a UA and stores the information into a location service in the domain it handles. When an UAC wants to initiate a session with a UAS, UAC must discover the current host (ip address) where the UAS is reachable. This discovery process is often done by SIP proxy servers and redirect servers which are responsible for receiving a request, determining where to send it based on knowledge of the location of the user, and then sending it there. To do this, SIP network elements asks the location service, that responds with a UA address within a particular domain.

Each user in a VoIP system gets a *SIP URI*. An URI is in RFC2396[12] defined as follows:

“A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource.”

The SIP URI is similar to a mail URL and have the following general form:

```
sip:user:password@host:port;uri-parameters?headers
```

A secured connection is indicated with *sips* that otherwise has the same form as *sip*. The following section shortly describes the different parts of the URI:

*User*: identifies a particular VoIP resource and is often a name or a phone number.

*Password*: is a password associated with the user. The SIP specification [2] does however not recommend that this field is used as the information is sent in clear text.

*Host*: identifies the host providing the SIP resource and can be a domain name or an ip address. A domain name is recommended when possible.

*Port*: identifies the port to be used. Default value is 5060 for *sip* and 5061 for *sips*.

*URI-parameters*: are parameters that effects the request constructed from the URI. They are on the form `name = value` and are separated by semi-colons if more than one. They are typically used to indicate routing or time-to-live values.

**Headers:** are fields that are to be included in a request.

The following five examples shows SIP and SIPS URIs where Alice is an user in domain atlanta.com and where Bob is an user in domain biloxi.com:

```
sip:alice@atlanta.com
sip:bob@biloxi.com:5060;transport=tcp
sips:alice@atlanta.com?subject=test
sip:+4712345678@biloxi.com
sip:alice;method=REGISTER@atlanta.com?subject=test
```

More information on SIP URIs can be found in [2].

The interaction between user agents in a SIP session is done with messages. These messages can be requests from a client to a server or responses from a server to a client. Both types of messages consists of a start-line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body[2]. The following example shows an INVITE message where UA Alice initiates a call to Bob:

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 151

v=0
o=Alice 2890844526 2890844526 IN IP4 client.atlanta.com
s=-
c=IN IP4 192.0.1.10
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

The body in this example are used to negotiate communication details between these UA using the Session Description Protocol. The SPD protocol is not discussed further here but a specification of the SDP protocol can be found in [13].

The sip core specification defines six different methods that can be used in a request:

- **INVITE:** Is used to invite a user or service to participate in a session.
- **ACK:** Is only used with INVITE and confirms that the client has received a final response to the INVITE request.
- **BYE:** Is sent by a UA to terminate a call. This message can be sent by either the caller or the callee.

- *CANCEL*: Is used to cancel a pending request.
- *OPTIONS*: Is used to query a server about its capabilities.
- *REGISTER*: Is used by a UAC to register its ip address with a UAS that in this case is called a registrar.

The response messages contain status codes and reason phrases that indicate the current condition of this request. The status code values are divided into six general categories:

- *1xx: Provisional*: The request has been received continuing to process the request.
- *2xx: Success*: An action was successfully received, understood, and accepted.
- *3xx: Redirection*: Further action is required to be able to complete this request.
- *4xx: Client Error*: The request contains bad syntax and cannot be fulfilled at this server.
- *5xx: Server Error*: The server failed to fulfill an apparently valid request.
- *6xx: Global Failure*: The request cannot be fulfilled at any server.

A SIP connection can either be provided directly between User Agents or in a hop-by-hop fashion as shown in figure 2.

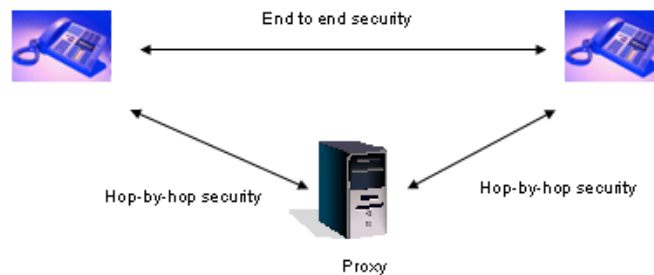


Figure 2: SIP connection

The following examples shows a call set up and tear down for these two scenarios. Example1 in figure 3: Alice, sip:alice@atlanta.com, invites Bob, sip:bob@biloxi.com, into a call and successfully establish a SIP session directly between the two users:

1. Alice (UAC) sends an request (INVITE message) directly to Bob (UAS).
2. The UAS responds with a 100 trying message.
3. The UAS also sends a 180 responds informing the UAC that Bob's phone is ringing.
4. Bob answers the phone and the UAS sends an 200 OK message to UAC
5. The UAC sends an ACK message to inform the UAS that the response was received. A session is now established and voice or other data is transferred with the RTP protocol.

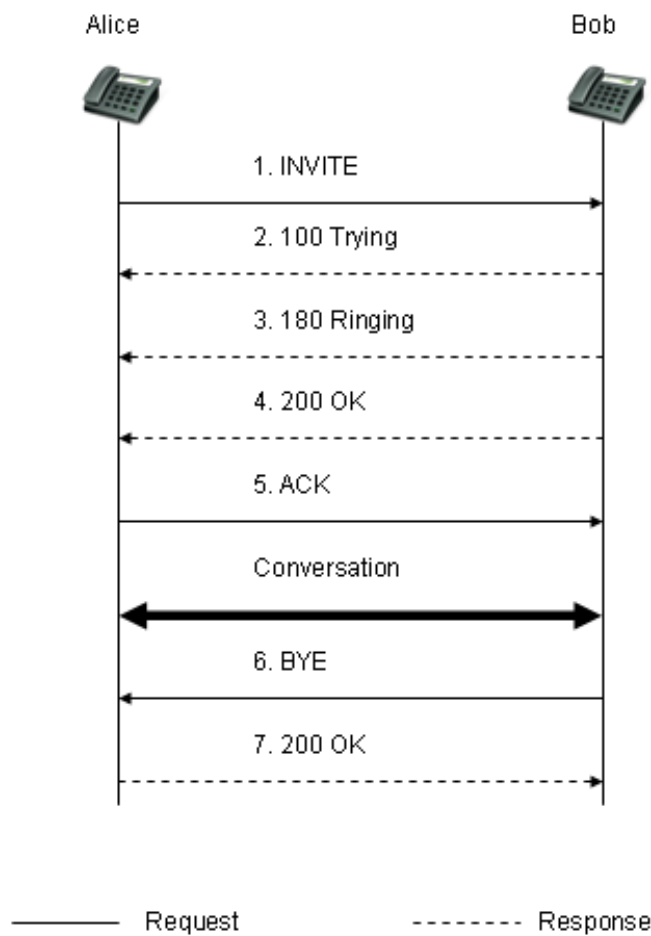


Figure 3: UA to UA

6. Bob ends the call and sends a BYE message
7. The Alice UAC responds with a 200 OK message and the session is finished.

Example2 in figure 4, Alice sip:alice@atlanta.com invites Bob sip:bob@biloxi.com into a call and successfully establish a SIP session between the two users traversing two proxies. User authentication is not shown in this example:

1. Alice (UAC) sends an request (INVITE message) to proxy atlanta
2. Proxy atlanta forwards the INVITE message to Proxy biloxi
3. Proxy atlanta responds with a 100 trying message
4. Proxy biloxi forwards the INVITE to user Bob
5. Proxy biloxi responds with a 100 trying message
6. The UAS also sends a 180 Ringing responds informing the proxy that Bob's phone is ringing.

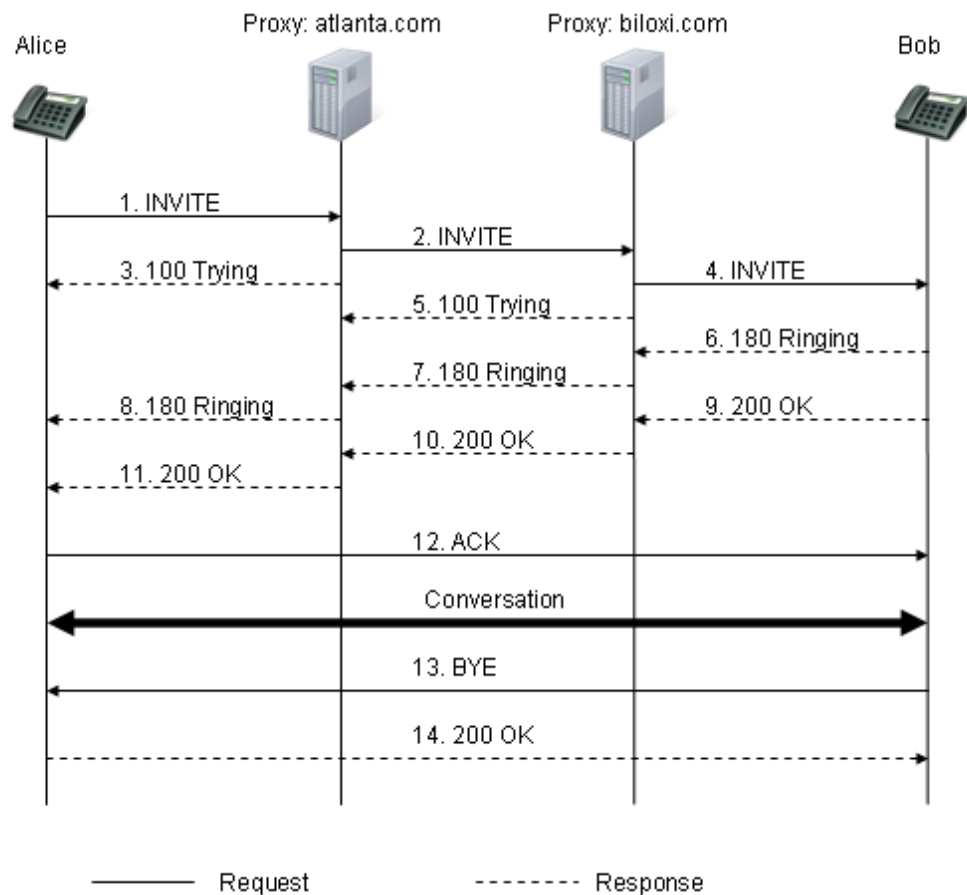


Figure 4: SIP communication using proxies

7. Proxy biloxi forwards the 180 Ringing response to atlanta
8. Proxy atlanta forwards the 180 Ringing response to Alice that will get a tone
9. Bob answers the phone and the UAS sends an 200 OK message to the biloxi proxy
10. Proxy biloxi forwards the 200 OK response to atlanta
11. Proxy atlanta forwards the 200 OK response to Alice
12. The UAC sends an ACK message directly to Bob. A session is now established and voice or other data is transferred with the RTP protocol.
13. Bob ends the call and sends a BYE message directly to Alice
14. Alice responds with a 200 OK message and the session is finished.

A UAC first has to register it self to the "REGISTRAR" server. This is done by sending a "REGISTER" message that binds a particular device contact URI with a SIP user Address of Record (AOR).

The example below shows a REGISTER message sent by Bob's UAC:

```
REGISTER sips:biloxi.com SIP/2.0
From: Bob <sips:bob@biloxi.com>;tag=a73kszlfl
To: Bob <sips:bob@biloxi.com>
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.com
CSeq: 1 REGISTER
Contact: <sips:bob@client.biloxi.com>
Content-Length: 0
```

More examples of basic SIP telephony call flows can be found in [14][15].

### 2.2.2 HTTP Digest Authentication

RFC3261 suggests that a SIP proxy should use HTTP Digest authentication to authenticate a User Agent Client. HTTP digest authentication offers authentication and replay attack protection for INVITE and REGISTER message requests. It does however not protect other messages like CANCEL, BYE and responses to requests. HTTP digest authentication is also vulnerable to dictionary attacks and does not introduce data integrity or confidentiality to a message as shown in [16]. This makes it easy to eavesdrop on messages and use different tools to retrieve a password. Tools like SiVus, Cain and Wireshark that are available on the Internet can be used to achieve this.

The HTTP digest authentication described in RFC3261 does not offer mutual authentication making it impossible for the user to authenticate the SIP proxy server. IETF suggests in their Internet draft [17] that the SIP specification should be updated with new header fields that enables mutual authentication.

The HTTP digest authentication is a challenge-response protocol[18] where nonce is sent in a response from a server to challenge a client that sent an initial request (INVITE). The response is generated from the username, password, nonce value, method and requested URI. This is often done with the hash algorithm MD5, but other hash algorithms can be used. The client send a new request (INVITE) that now also contains an Authorization header with the generated response.

Figure 5 shows an example of user authentication with HTTP digest authentication as described in the SIP specification.

1. Alice (UAC) sends an request (INVITE message) to proxy atlanta
2. Proxy atlanta requires user authentication and responds with a *407 Proxy authentication required* message. This response contains a nonce that will be used in the generation of a authorization response.
3. Alice (UAC) acknowledges the proxy message by sending an ACK.
4. Alice (UAC) sends an new request (INVITE message) to proxy atlanta with authentication information that contains the authorization response.
5. Proxy atlanta verifies the credentials and forwards the INVITE message to Proxy biloxi
6. The rest of the SIP flow is as described before.

Pauli Vesterinen identifies in his paper, User authentication in SIP [19], that a user password must be known by both the user and the proxy that handles the authentication. This must be seen as a weakness in the solution as the server, SIP proxy in this case, may

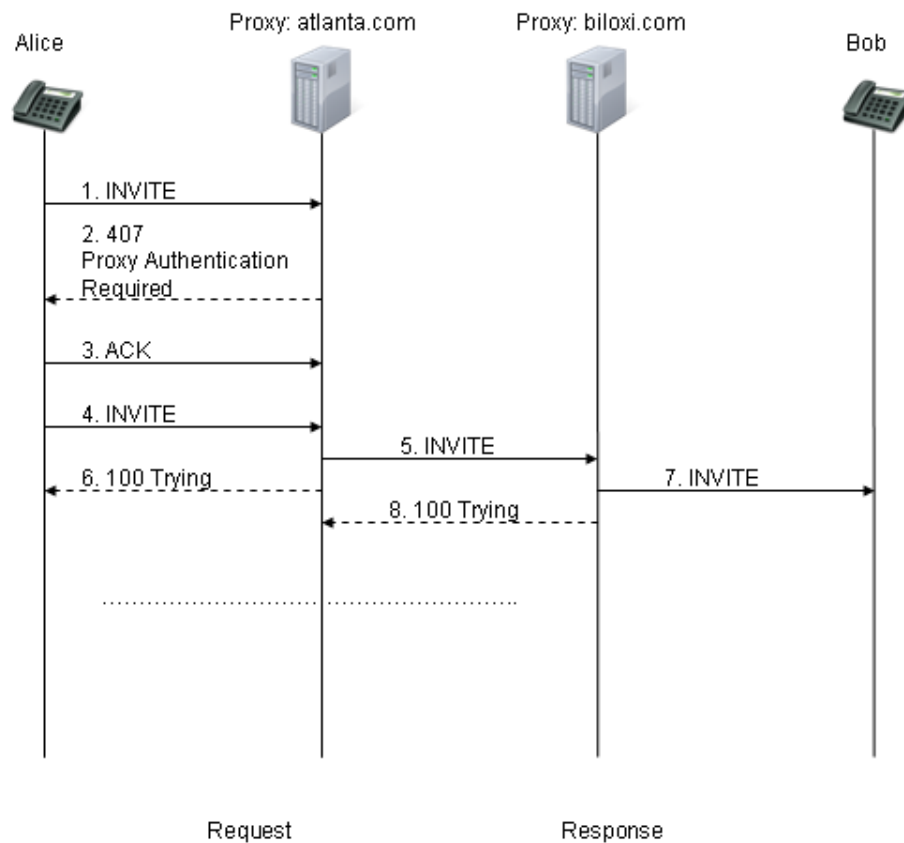


Figure 5: HTTP digest authentication

be exposed to attacks that may give an attacker access to password to all registered users.

HTTP Digest Authentication replaces the deprecated authentication method HTTP Basic Authentication. More information about HTTP Digest authentication can be found in [2] [10] [18].

### 2.2.3 SIPS

TLS can be used to introduce integrity and confidentiality to SIP between two points. The *S* in *SIPS* signifies that each hop, to which the request is forwarded until the request reaches the SIP UAS, must be secured with TLS. This is indicated in the SIP URI with *sips*, for example *sips:alice@atlanta.com*. Users that have distributed an address record with *sips* in the SIP URI, can refuse connections over insecure transports. TLS can only be used with a connection oriented protocol like TCP or SCTP, i.e. UDP is not an option.

TLS can be used in a hop-by-hop security architecture where hosts have no pre-existing trust relationship with each other. For example, a UAC can set up a secured connection with its local proxy server, that in its turn sets up a secured connection with the local proxy of the recipient. This proxy may then set up a secured connection with the recipient, as can be seen in figure 6. This indicates the need of some kind of public key infrastructure.

Using TLS to secure the SIP signaling has its limitations as each proxy needs the SIP header in clear text to be able to route the message properly. The result from this is

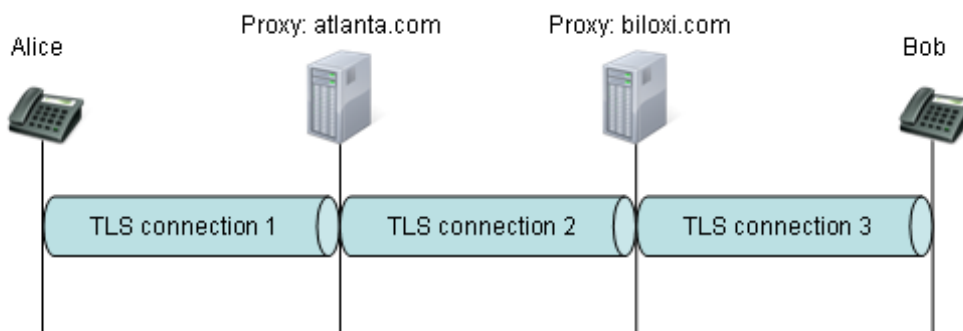


Figure 6: SIP protection using TLS

that all proxies in use in a connection must be trusted as messages are decrypted and encrypted in each node.

As the transport is secured on a hop-by-hop basis, a UAC that sends a request over TLS to a proxy server has no actual assurance that TLS will be used end-to-end, but must trust that the different service providers involved do not read these messages. End users are probably not aware of this and may, when using SIPs, get a false sense of security.

TLS can also be used to protect the negotiation of an encryption key between end users. The lack of end-to-end security may however lead to exposure of this key, as it may be transported in clear in the SIP message body in some part of the network.

This is also no assurance that a SIP message can't be intercepted in the recipient's network as it is no requirement for securing the SIP message there.

This research should indicate if SIPs with TLS is/is not the preferred way of introducing confidentiality and integrity to the signaling in VoIP. The research should also focus on availability and ease of use.

#### 2.2.4 IPsec

IPsec [20] can be introduced to provide confidentiality, integrity, data origin authentication and replay protection to SIP. IPsec can not be used in an end-to-end scenario as proxy servers need to read from and sometimes write to the SIP header. IPsec can however be used to secure the connection between hosts in a hop-by-hop security architecture. An advantage of IPsec is that it is implemented at the operating system level securing connections that can be established without involvement of a UA. IPsec can also be used to secure UDP connections. IPsec assumes however that a pre-established trust relationship is introduced between the communicating parties making it most suited for SIP hosts in a VPN scenario. Further, the SIP specification [2] does not describe how IPsec should be used, neither does it describe how key management should be realized. There is not much to be found on how IPsec should be used to secure SIP.

This research should state if and how IPsec should be used to secure a connection in VoIP.

#### 2.2.5 S/MIME

Multipurpose Internet Mail Extensions (MIME) defines a mechanism for sending different kinds of data in an email. MIME can also be used to secure other applications. The same



security mechanisms that are used to secure a MIME body in an email can be used to secure a SIP body as it also is MIME encoded. The MIME security mechanism is referred to as S/MIME and is specified in RFC 2633[21]. Even more information of S/MIME and S/MIME in SIP can be found in [22] [2].

S/MIME adds security to the message itself and can be used to provide end-to-end security to SIP. A client (UA) can use S/MIME to sign and/or encrypt a SIP message. The message is then signed with the private key of the sender and a certificate with a public key is forwarded to the recipient embedded to the message in a MIME attachment. The signature is then be verified by the recipient.

The encryption of a message is a bit more tricky. It requires that the public key of the recipient is known to the sender. This key must be fetched in advance or be fetched from some kind of central repository, the message body is then encrypted with the public key of the recipient. The message is in the end decrypted with the the private key owned by the recipient.

The following example that is taken from the SIP specification [2], shows a SIP message with an encrypted message body that is marked with \*.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m
Content-Disposition: attachment; filename=smime.p7m handling=required
```

```
*****
* Content-Type: application/sdp *
* * *
* v=0 *
* o=alice 53655765 2353687637 IN IP4 pc33.atlanta.com *
* s=- *
* t=0 0 *
* c=IN IP4 pc33.atlanta.com *
* m=audio 3456 RTP/AVP 0 1 3 99 *
* a=rtpmap:0 PCMU/8000 *
*****
```

The largest outstanding defect with the S/MIME mechanism is the lack of a prevalent public key infrastructure [2]. The solution may also be vulnerable to man-in-the-middle attacks if self signed certificates are used.

The Pretty Good Privacy (PGP) that was described in the previous version of the SIP specification is now deprecated in favor of S/MIME.

The research should indicate if S/MIME can and should be used to sign and/or encrypt a SIP message body.

### 2.3 Identity Based Cryptography

Adi Shamir asked in his publication, Identity-Based Cryptosystems and Signature Schemes [23] for a public key encryption scheme in which the public key can be an arbitrary string. Shamir's original idea was to enable users to communicate securely and to be able to verify each others signatures without exchanging public keys.

The design of such a system was a long-standing open problem in cryptography. Boneh and Franklin proposed in their article Identity-Based Encryption from the Weil Pairing[24] the first secure and practical Identity Based Encryption (IBE) system. The main concept of Identity Based Cryptography and an short description of the solution proposed by Boneh and Franklin is now presented. This information is needed to understand the work done in this master thesis.

In Identity Based Encryption, each user choses an arbitrary string, like an email address as a public key. If for example Alice wants to send an email to bob, she simply encrypts her message using using bob's public key, that in this case is bob's email address bob@biloxi.com. Note that there is no need for Alice to obtain Bob's public key from a certificate, removing the need for certificate handling.

When Bob receives the encrypted mail from Alice, he first has to obtain his private key. The private key is generated by a trusted third party that in IBE is called a Private Key Generator (PKG). Bob has to authenticate him self to the PKG in the same way he would authenticate himself to a CA. The private key is then generated and sent to Bob. Bob can then decrypt and read his email.

It should also be noted that the PKG contains private keys for all users in the system and that key escrow is inherent in IBE.

An IBE scheme can be divided into 4 parts: *Setup*, *Extract*, *Encrypt* and *Decrypt*.

*Setup*: Global system parameters and a master secret is generated out from a security parameter. The system parameters will be publicly known, while the master secret will be known only to the Private Key Generator (PKG).

*Extract*: The system parameters, master secret and an identity string is used by the PKG to generate a private key. The identity string is an arbitrary string that will be used to identify an user.

*Encrypt*: Message is encrypted using system parameters and a calculated public key.

*Decrypt*: System parameters and the corresponding calculated private key is used to decrypt a message.

Identity Based Cryptography can also be used to sign messages. This is done with the following steps [25] where an identity-based signature scheme is used.

*Setup*: Global system parameters and a master secret is generated out from a security parameter. The system parameters will be publicly known, while the master secret will be known only to the Private Key Generator (PKG).

**Extract:** The system parameters, master secret and an identity string is used by the PKG to generate a private key. The identity string is an arbitrary string that will be used to identify an user.

**Signature generation:** A signature is generated for a message using private key and chosen signature scheme.

**Signature verification:** The signature is verified on arrival using senders identity string and PKG public key.

### 2.3.1 An Short introduction to the Boneh-Franklin IBE system

The Identity Based Encryption suggested by Boneh-Franklin[24] uses Elliptic curves. The basic units for this cryptosystem are points  $(x, y)$  on an Elliptic curve,  $E$ , over a finite field,  $F_p$ , denoted  $E(F_p)$  of the form[26]:

$$y^2 = x^3 + ax + b \text{ with } x, y, a, b \in F_p = \{1, 2, 3, \dots, p-2, p-1\}$$

Not all Elliptic curves are suitable for cryptographic purpose. How a curve is chosen it not described here, but more information can be found in [24][27]. In the following part  $+$  and  $*$  (that may be omitted) refer to Elliptic curve addition and multiplication. Capitals like  $P$  and  $Q$  represent points on the curve while lower case represent integers.

Given a message,  $m$ , we must first choose a large prime,  $q$ , and a suitable elliptic curve,  $E(F_q)$ . We must then embed the message  $m$  onto a point,  $P$ , on the curve. The symbol  $\oplus$  denote bitwise exclusive or, XOR.

The fundamental to this system is *bilinear mapping*. An example of such a mapping is the *Weil pairing*,  $\hat{e}$ , that takes two points on  $E(F_p)$ , outputs an integer, and has the property that

$$\hat{e}(xP, yQ) = \hat{e}(P, Q)^{xy} \text{ for any points } P, Q \text{ and for any integers } x, y \text{ [26].}$$

The following list describes the Boneh-Franklin FullIdent scheme[24]:

#### **Setup:**

A security parameter  $k$  is needed that defines the bit strength that the encryption will provide. Generate a prime  $q$ , the two groups  $G_1, G_2$  and define a pairing  $\hat{e}$ . An elliptic curve  $E$  is chosen in advance.

Let  $P$  be an arbitrary point in  $G_1$ . Pick a master secret  $s$  and let it only to be known by the PKG. The length of the plain text is  $n$ , message to be encrypted is  $m \in M$  and the Cipher text to be decrypted is  $c \in C$ . Four cryptographic hash functions  $H_1, H_2, H_3$  and  $H_4$  are defined. Calculate the public key  $P_{pub} = sP$ .

This results in the following list of public system parameters

$$\langle q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4 \rangle.$$

#### **Extract:**

User are identified with an string ID. Users private key  $d_{ID}$  is calculated,  $d_{ID} = sQ_{ID}$  where  $Q_{ID} = H_1(ID)$

#### **Encryption:**

The sender first generates a random  $\sigma$  and sets  $r = H_3(\sigma, m)$  where  $m$  is the message in plain text.

$$C = \langle rP, \sigma \oplus H_2(g_{ID}^r), m \oplus H_4(\sigma) \rangle \text{ where } g_{ID} = \hat{e}(Q_{ID}, P_{pub})$$

The encrypted message  $C = \langle U, V, W \rangle$  is then sent to the receiver.

**Decryption:**

The receiver uses his private key  $d_{ID}$  to decrypt message  $C = \langle U, V, W \rangle$  with the following:

$$V \oplus H_2(\hat{e}(d_{ID}, U)) = \sigma$$

and

$$W \oplus H_4(\sigma) = m$$

where  $m$  is the resulting plain text.

More on the mathematical foundation behind Identity Based Encryption can be found in [27].

### 2.3.2 Identity-based signature schemes

There exists several identity-based signature scheme, one of them is the Sakai et al's identity-based signature scheme [28] which is based on pairing and is described as follows [29]:

**Setup: and Extract:**

The same as for the Boneh-Franklin IBE.

**Signature generation:**

To sign a message  $m$ , the signer produces a signature as  $(Y, Z)$  where

$$Y = yP$$

and

$$Z = yH_2(ID, Y, m) + d_{ID}$$

$y$  is a random generated number.

**Signature verification:**

The signature is verified by checking the following:

$$\hat{e}(Z, P) = \hat{e}(H_2(ID, Y, m), Y) * \hat{e}(H_1(ID), P_{pub})$$

## 2.4 Security issues

### 2.4.1 VoIP threats and Vulnerabilities

When considering VoIP instead of PSTN, users and implementors must be aware of the threats and vulnerabilities that exists in VoIP systems. This section will identify the major threats and vulnerabilities that has been discovered so far. New weaknesses are however expected to be found. Identified threats and vulnerabilities will be used as input to the work done in the research.

As VoIP is using existing IP infrastructure, it will inherit the threats that exists to these environments. It will be vulnerable to traditional network attacks like worms, viruses, trojans and denial of service. This indicates that each layer needs to be secured independently, also referred to as defense in depth.

We will here focus on threats to the VoIP specific protocols in the infrastructure. The supporting infrastructure (routers, switches and other general network components) are not in focus in this master thesis but more information on this can be found in Security engineering[30].

VOIPSA (Voice over IP Security Alliance) is an organization that

“aims to fill the void of VoIP security related resources through a unique collaboration of VoIP and Information Security vendors, providers, and thought leaders.”

VOIPSA has published a paper [31] on their web site ([www.voipsa.org](http://www.voipsa.org)) that defines taxonomy for security threats to VoIP systems. The following categories of threats were identified:

- Eavesdropping threats
- Interception and Modification (Service abuse)
- Intentional Interruption of Service
- Non-Technical threats (Social threats)

The VOIPSA paper [31] contains an extensive list and classification of VoIP specific and non VoIP specific threats and attacks that may be launched on a VoIP system. Other taxonomies are available from other organizations that are not as extensive as the VOIPSA. Internet Engineering Task Force(IETF) has for example published VoIP Security Threats [32] that focus on the VoIP specific threats to the Location Function, Signaling Function and Media Function.

Peter Thermos and Ari Takanen [33] builds on and extends the taxonomy from VOIPSA and IETF and identifies the following categories of threats to VoIP:

- Service disruption and annoyance
- Eavesdropping and traffic analysis
- Masquerading and impersonation
- Unauthorized access
- Fraud

Each of these categories contain groups of attacks that can be launched to the different layers in the protocol stack in a VoIP system. A typically VoIP protocol stack is shown in figure 7.

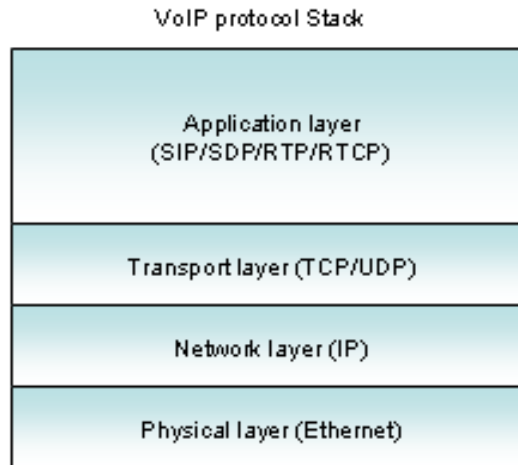


Figure 7: Voice Over IP protocol stack

The following sections will describe the categories from, Securing VoIP networks[33], and list sub-categories of threats and give some example of threats that are identified. This listing is necessary for identifying the requirements that will be used as input to the design of the security solution.

#### Service disruption and annoyance

This category contains attacks that are intended to disrupt the VoIP service or to annoy the users in the system. These attacks can affect any network element that supports the service. Annoyance refers to attacks that annoy the receiver by making calls that one cannot refuse.

This category is divided into sub-categories that are described below.

- VoIP specific Denial of Service, DoS
- General Denial of Service, DoS
- Physical attacks
- Annoyance
- Performance Latency

*VoIP specific Denial of Service, DoS* refers to threats to VoIP specific protocols that makes VoIP services unavailable for a user of a system. This can be done by overwhelming the target with requests that may be valid or invalid. This can also be done by violating the quality of services that has been negotiated in the call setup and for example use another codec, making the service unavailable. SIP specific attacks like *Malformed Packet DoS* and *SIP flooding* makes the signaling system unavailable.

A spoofed message can be inserted into the signaling between parties that among others may tear down a call or fake a response.

VoIP communication between components in a network may be exposed to call hijacking if the security in the system is compromised. Call hijacking occurs when transactions are taken over by an attacker. This may lead to service disruption for the caller. Both the signaling and the media stream may be exposed to hijacking.

*General Denial of Service, DoS* refers to treats that can disrupt the network component that a VoIP network consists of. This can be routers, switches, proxies, DNS servers, modems, computers and other. An attacker can exploit vulnerabilities that exists in a system (like buffer overflow) and get access to network components. Examples of attacks that can cause DoS are SYN flooding, Smurfing, Spoofing attacks, Routing attacks and a number of protocol attacks. A description of these attacks and many more can be found in [30].

Phones, proxies/gateways and other components often runs on an operating system like IOS, Windows or Linux. This makes the VoIP components vulnerable to attacks that these operating systems are exposed to. It is important to understand that a VoIP application is not more secure than the platform it runs on.

In Distributed Denial of Service (DDoS) attacks, network resources are exhausted by a large number of computer simultaneously generates traffic. This can be accomplished by an attacker by subverting a large number of computers with a attack software like a worm. This worm can then be triggered by a specific event or at a predefined time, generating large amount of traffic to a victim. This may among other things block VoIP services and network components that otherwise would have been available.

*Physical attacks:* the components in a VoIP network also has to be physically secured. an adversary may otherwise compromise the system and for example insert packets to a network, listen to network traffic, spoof ip packet, change configurations or just cause a loss of power that will make the service unavailable. An battery backup system should also be considered by service providers.

*Annoyance:* Spam over Internet Telephony or SPIT can be compared to e-mail spam and it often refers to automatically-dialed, pre-recorded phone calls using VoIP. More information of SPIT can be found in [34] [35].

Another way to annoy users is to set the call priority to urgent, forcing a call to come through even if you as a receiver want to to remain unavailable. This can be done by an attacker by manipulating the call priority level and set it to high/urgent or equal.

In SIP an INVITE message can be sent to multiple users making all phones to ring. If the caller id also has been faked, it will result in that all receivers will call/attack the originating source.

*Performance Latency:* Latency is defined in Wikipedia as "a time delay between the moment something is initiated, and the moment one of its effects begins or becomes detectable". ITU-T G.114[36] recommends a maximum of a 150 ms one-way latency to get an acceptable quality.

### Eavesdropping and traffic analysis

In VoIP, eavesdropping and traffic analysis attacks refers to methods for monitoring and analyzing of signaling and media-streams between different endpoints in a VoIP network. The data sent in the monitored network is not altered in any way. The purpose of this is to collect sensitive information that may be used in an attack later. The following sub-categories identified in [31] and are briefly described here:

- Call Pattern Tracking
- Traffic Capture
- Number Harvesting
- Reconstruction of conversation,voicemail,fax,video and text.

*Call Pattern Tracking* refers to the analysis of traffic between nodes in a network. This technique can be used to collect information like identity, presence and usage that can be used in later attacks.

*Traffic Capture* is a basic method for recording a communication without the consent of all the parties[31].

*Number Harvesting* refers to unauthorized gathering of identities like phone numbers or SIP urls.

*Reconstruction of conversation,voicemail,fax,video and text* refers to "unauthorized monitoring, recording, storage, reconstruction, recognition, interpretation, translation and/or feature extraction"[31] of any conversation, voicemail, fax, video or text.

### Masquerading and impersonation

This category contains attacks where the attacker masquerades as a legitime user or device. This can be done with a so-called subscriber impersonation where an attacker uses credentials that are previously captured. This can also be done with device impersonation where an attacker impersonates devices like phones, SIP gateways, RTP gateways, registrars or other network components.

User impersonation can be done by altering the signaling information (SIP message). User impersonation/Caller ID spoofing can for example be done at the end points by generating a false caller id in the SIP message. The same result can be achieved if an attacker has control over a VoIP proxy, by again altering the signaling.

Another example is presence hijacking where an attacker abuse an SIP REGISTER message to alter an IP address to a device (SIP phone), an example of this can be found in [33]. The attackers device is in this case actually registered as an legitimate user device. This will of cause also deny the real user of service as it is the attackers device that is registered.

Other examples of masquerading attacks that has been identified and that must be considered when developing a new solution can be found in [31][33].

### Unauthorized access

This category contains attacks that gets an user unauthorized access to services. Peter Thermos and Ari Takanen has the following definition of unauthorized access in [33]:

The ability to access a service, functionality, or network element without proper au-



thorization.

A telephony system often contains many subsystems that has to be protected for unauthorized access. This can be call control, switches, registration, billing and other systems needed in a complete system [37].

Common methods to gain access to these system includes impersonation and Man-in-the-middle attacks. Impersonation can be accomplished by stealing credentials from a legitimate user. This can be done with dictionary or brute force attacks if a username and password system is in use, or by sniffing signaling and network traffic trying to identify credentials sent in the setup process.

Unauthorized access to a billing system can for example lead to fraud and bad publicity for the service provider that has been exposed to the attack.

In VoIP, the following areas has at least to be secured for unauthorized access:

- VoIP services.
- VoIP infrastructure, network infrastructure and network elements
- End devices like phones and software
- Management systems
- Operation and maintenance

### Fraud

Fraud in telecommunication is today fairly common. The *Communications Fraud Control Association* have done surveys to estimate annual telecoms fraud losses globally[38]. The result shows that the estimated annual global fraud losses in 2005 is in the range of 54.4 - 60 billion (USD). This shows an increase in fraud by 52 per cent from the CFCA Survey results of 2003.

The International Engineering Consortium (IEC) expected that providers will experience new types of fraud when VoIP is introduced. IEC lists in their fraud analysis [39] reasons to why VoIP also is exposed to fraud. They state the following reasons for this:

- They are open and distributed in it's nature
- Lack inherent security mechanisms
- Run mission-critical applications
- Become increasingly complicated
- Are deployed before they have fully matured
- Offer few expert solutions for their effective management
- require time- and cost-consuming integration and configuration

They also state that the fact that there are many tools with Graphical User Interface (GUI), scripts and detailed hacking instructions on the Internet, makes fraud in VoIP networks easier to perform.

Signaling and voice in VoIP is transfered in the same network, giving a user access to

both. This gives an attacker for example, the possibility to eavesdrop and misuse signaling information fetch from a legitimate call setup. In newer PSTN networks, the signaling has been separated from the voice, removing the possibility for making fraudulent calls. The fact that a VoIP network may have gateways against the Internet and different service provider also makes the solution exposed to different Internet attacks.

Phishing attacks are known from Internet and often refers to attempts to acquire sensitive information like credit card numbers by masquerading as a known entity (person, company etc). Phishing attacks are also a threat to VoIP where an attacker provides a user of a service with a phone number. This is typically done by sending out emails to a group of users. The attacker has in advance set up an IVR (Interactive Voice Response) or a voice mail system that is asking for confidential information. IVR and voice mail systems can freely be downloaded from the Internet.

## 3 Securing the VoIP Signaling

This chapter propose a new way to secure the Session Initiation Protocol in VoIP

### 3.1 Requirements

The following part lists and explains requirement that we have on a new security solution to SIP

1. Confidentiality must be provided to the SIP signaling. It should not be possible to eavesdrop on a dialog to get usable information.
2. Integrity must be provided to the SIP signaling. A receiver of must be able to identify if a change is made to a message.
3. Availability: Signaling must not be delayed more that what can be accepted by an human user. The solution should not introduce long delays in the signaling.
4. Non-repudiation of origin: The sender of a message should not be able to deny having sent this message.
5. Regulatory services: Common regulatory services like Lawful Interception must be supported. Lawful Interception: It must be possible to tap the line for signaling and voice data. This is a requirement on service providers to provide legally sanctioned official access to private communications.
6. Flexibility: Users must be able to plug in a telephone or to run softphone from a random location. A user should for example be able to run a softphone from a PC that is connected to a local WLAN.
7. Client authentication: A server must be able to trust the given client identity. Each SIP message contains caller and callee identity. The called party must be able to trust the caller identity in a request.
8. Server authentication: A client must be able to trust the given server identity. Each SIP message contains caller and callee identity. The calling party must be able to trust the callee identity in responses.
9. End to end security should be provided to prevent eavesdropping at any node. Users must be able to trust that messages are secured from one end to another.
10. User friendly: The solution must be easy to use for non technical users. Users should not need to know anything about SIP and information security. The goal is to make a "zero effort" security solution for end users.

### 3.2 Choice of technology

The proposed solution should, as stated in the requirement, provide security services to the SIP messages and also provide end to end security. This naturally excludes the usage of IPSec and transport security like TLS as each node involved in routing has to decrypt and encrypt each message. The message is in the TLS case left in clear text for a short

time of period, making these nodes a target for attackers.

SIP messages are by default MIME encoded and the construction follows the MIME specification[40]. The secured version on MIME is referred to as S/MIME and is today commonly used to secure email services. S/MIME especially provides authentication, confidentiality, integrity and non-repudiation of origin. S/MIME provides security to the message it self and does also secure the message from one endpoint to another. These are properties that are desirable in VoIP making it a good candidate for SIP security.

S/MIME is based on the usage of public key infrastructure and certificates containing public keys. S/MIME requires that a user gets a certificate with a public key for a user that he or she want's to communicate with in advance. Such solution is often considered to be complicated to use by end users and requires some knowledge of the chosen security solution. This property of PKI solutions of today make them hard to use and only a few people actually use PKI in this way.

An alternative and more user friendly solution is needed for securing VoIP. A technology that can help us with this is Identity Based Encryption (IBE). IBE is a type of public key cryptography where the public key is a string identifying a user. This property of IBE removes the need for certificates and PKI solutions.

The hypothesis in this MSc thesis is that the combination of S/MIME and IBE is the ideal solution for securing the signaling protocol SIP.

### 3.3 Usage of S/MIME in SIP

*S/MIME* is introduced to SIP to ensure both integrity and confidentiality to messages. The MIME content type multipart/signed and application/pkcs7-mime can be used. Integrity and Confidentiality can however not be applied to the whole message as this header information has to be parsed by node that are routing the traffic on a network.

#### 3.3.1 Providing integrity

SIP proxy servers may sometimes need to modify the header fields in a SIP message. SIP message header fields that can be modified by proxy servers are: *Request-URI*, *Route*, *Via*, *Record-Route*, *Max-Forwards* and *Proxy-Authorization*. A proxy may also modify the SIP message by adding another *Via* field. The integrity check should therefore not include these fields.

Integrity can be provided to the SIP message by tunneling the original SIP message within a S/MIME body. The content type is in this case "*message/sip*" where the MIME body is signed with a detached signature.

A new outer header is created that contains a copy of the original message header.

The following example, that is taken from the SIP specification[2], shows how tunneling can be used to provide integrity to a SIP message.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
```

Date: Thu, 21 Feb 2002 13:02:03 GMT  
Contact: <sip:alice@pc33.atlanta.com>  
Content-Type: multipart/signed;  
protocol="application/pkcs7-signature";  
micalg=sha1; boundary=boundary42  
Content-Length: 568

--boundary42  
Content-Type: message/sip

INVITE sip:bob@biloxi.com SIP/2.0  
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8  
To: Bob <bob@biloxi.com>  
From: Alice <alice@atlanta.com>;tag=1928301774  
Call-ID: a84b4c76e66710  
CSeq: 314159 INVITE  
Max-Forwards: 70  
Date: Thu, 21 Feb 2002 13:02:03 GMT  
Contact: <sip:alice@pc33.atlanta.com>  
Content-Type: application/sdp  
Content-Length: 147

v=0  
o=UserA 2890844526 2890844526 IN IP4 here.com  
s=Session SDP  
c=IN IP4 pc33.atlanta.com  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

--boundary42  
Content-Type: application/pkcs7-signature; name=smime.p7s  
Content-Transfer-Encoding: base64  
Content-Disposition: attachment; filename=smime.p7s;  
handling=required

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6  
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj  
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4  
7GhIGfHfYT64VQbnj756

--boundary42-

The original message is located in the body part which are completely tunneled and included in the body part with Content-Type: message/sip.

The detached signature located in the second body part with Content-Type: application/pkcs7-

signature, verifies the integrity of the inner (original) SIP message. The outer header can also be verified against the inner header by comparing the fields. The sender and receiver can with sip tunneling, be ensured that the message is not changed by an unauthorized part.

SIP tunneling will also provide authentication of the originator as the *From* field is a part of the integrity check. Any changes to this field will be discovered by the receiver. The originator is in this case a client that initiates a SIP session and a sever that responds to the client messages. A client is signing it's message and a server is signing responses identifying both parts in a session.

Closely related to this is Non-repudiation of origin that will be provided as this message is signed with the private key of the sender. Only a user that posses this key will be able to sign a message. This will cover the client and the server part, i.e caller and callee.

Using clear signing format for messages makes it possible for users to use these messages whatever they have S/MIME software or not.

### 3.3.2 Providing integrity and confidentiality

Proxy servers needs to read some of the headers fields from a message to be able to route the SIP message correctly. The SIP specification defines that the following fields must be in plain text: *To*, *From*, *Call-ID*, *CSeq* and *Contact*. Confidentiality should be provided to the rest of the header fields and to the body part of the SIP message that may contain sensitive information.

Tunneling can provide both integrity and confidentiality to a message. This is done by signing and encrypting the "*message/sip*" MIME body and creating an S/MIME body. The following example, that is partly taken from the SIP specification[2], shows how tunneling can be used to provide integrity and confidentiality to a SIP message. The encrypted part in this example is indicated with \* signs.

Note that a content encryption key is added to the body and that this key is encrypted using Identity Based Encryption.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.com>
From: Anonymous <sip:anonymous@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:pc33.atlanta.com>
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature";
  micalg=sha1; boundary=boundary42
Content-Length: 568

--boundary42
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
```

```

    name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
    handling=required
Content-Length: 231

*****
* Content-Type: message/sip *
* * *
* INVITE sip:bob@biloxi.com SIP/2.0 *
* Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 *
* To: Bob <bob@biloxi.com> *
* From: Alice <alice@atlanta.com>;tag=1928301774 *
* Call-ID: a84b4c76e66710 *
* CSeq: 314159 INVITE *
* Max-Forwards: 70 *
* Date: Thu, 21 Feb 2002 13:02:03 GMT *
* Contact: <sip:alice@pc33.atlanta.com> *
* * *
* Content-Type: application/sdp *
* * *
* v=0 *
* o=alice 53655765 2353687637 IN IP4 pc33.atlanta.com *
* s=Session SDP *
* t=0 0 *
* c=IN IP4 pc33.atlanta.com *
* m=audio 3456 RTP/AVP 0 1 3 99 *
* a=rtpmap:0 PCMU/8000 *
* k=123456789 *
*****
* Content Encryption Key *
*****

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
    handling=required

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

--boundary42-

```

The original message is again totally tunneled and placed in the first body part. Content-Type of the S/MIME body is in this case "application/pkcs7-mime; smime-type= enveloped-data; name= smime.p7m" indicating that the content in this body part is encrypted.

The body may for example include sensitive data like a session key that later can be used to encrypt the actual speech-data between the users. This key is often placed in a *k* parameter in the inner body. It is of cause of great importance that this session key only is known by senders and receivers to prevent eaves dropping of speech.

The signature is again placed in a second body part. This signature can be generated from the message before it is encrypted, ensuring that the plain text message has not been altered. An attacker may however have have changed attributes that are not included in the signature. Another alternative is to encrypt the message first and then signed it, making it possible to verify a signature before decryption.

The two signing solutions has there advantages and disadvantages and it is up to the implementor to choose the appropriate one. This choice depending on the use cases that has to be considered.

The outer header should only contain information that is necessary for routing the message correctly. The example above shows how the identity of the initiator can be kept anonymous to all parts except the receiver by providing an new outer *From* header field that contains no personal information, From: Anonymous <sip:anonymous@atlanta.com>. The inner *From* header field that is only readable to the receiver after decryption contains the true identity of the originator, in this case  
From: Alice <alice@atlanta.com>.

This solution provides Integrity, Confidentiality, Client authentication, Server authentication, Non-repudiation of origin and end to end security. This covers many of the requirements stated above and must be considered to be a good alternative for securing VoIP and SIP

### 3.3.3 Removal of Call-ID

The header field Call-ID is an unique identifier that groups together all messages within a session. The SIP specification states that this is a mandatory field that has to be in clear text. This will however in some cases make it possible to trace user activity. Both INVITE and BYE has the same Call-ID making it possible to identify both parties in a session. The Call-ID header field should be removed from the outer header to remove this weakness in the SIP protocol.

### 3.3.4 Negotiating SMIME capabilities

The *SMIMECapabilities* Attribute is used by the sender of a S/MIME body to inform the receiver of it's capabilities and preferences for further communication.

The SIP specification[2] states the following

S/MIME implementations MUST at a minimum support SHA1 as a digital signature algorithm, and 3DES as an encryption algorithm. All other signature and encryption algorithms MAY be supported.

The *SMIMECapabilities* attribute has to be extended with new values identifying usage of Identity Based Cryptography for signing and or encryption of a S/MIME body.



### 3.4 Identity Based Cryptography and S/MIME

An S/MIME implementation has to follow the S/MIME specification that describes how to add signatures and encryption to MIME data. S/MIME also requires that the Cryptographic Message Syntax (CMS) [41] is followed. In S/MIME and CMS, a content encryption key is used to encrypt/decrypt the message body. The content encryption key is then encrypted with the recipients public key for key transport as stated in (CMS) [41]. Different algorithms that are in use today is specified in RFC3370: Cryptographic Message Syntax (CMS) Algorithms [42]. This specification states that the RSA public-key cryptosystem should to be implemented.

A new specification has to be developed that states how different identity based cryptosystems should be integrated with [41] and [42]. These specifications may also need to be generalized to support public key cryptosystems that are not based on certificate usage.

This new specification is not discussed further here but has to be developed by organizations like "The Internet Engineering Task Force" as this is an extensive amount of work that is out of scope for this Master thesis.

#### 3.4.1 Usage of IBE in S/MIME

IBE is a public key cryptosystems where the public key is a string that identifies an user. The corresponding private key is generated by a trusted third party called a Private Key Generator or just PKG.

The public key cryptosystems that are used today, mainly RSA, has to be replaced by an IBE cryptosystem. There are different IBE algorithms available that can be used in VoIP. An introduction to different Identity Based Encryption algorithms can be found in [27].

The introduction of IBE, changes the way sender and receiver uses public/private key in S/MIME. The following list shows the different steps that the sender and receiver at least has to take:

#### *Sender:*

- Get system parameters for IBE scheme. The sender and receiver has to exchange system parameters in advance.
- Get the the public key for the receiving provider. There is actually one public key / service provider.
- Encrypt the content encryption key with the provider public key and the identity string using chosen IBE scheme.

#### *Receiver:*

- Get system parameters for IBE scheme.
- Calculate private key using system parameters and own identity string.
- Decrypt content encryption key using newly calculated private key using chosen IBE scheme.

### 3.4.2 Identity-based signature schemes in S/MIME

The introduction of identity-based cryptography also changes the signing process in S/MIME. The following list shows the different steps that a signer and a verifier at least has to take:

*Sender:*

- Get system parameters for signature scheme. The sender and receiver has to exchange system parameters in advance.
- Get signers private key.
- Generate a signature with chosen signature scheme and private key.

*Receiver:*

- Get system parameters for signature scheme.
- Get signers public key and identity string.
- Verify signature using chosen signature scheme, public key and signers identity string.

Identity Based Cryptography inherits key escrow where owner of private key and a Private Key Generator is able to sign a message. The PKG can sign a message on behalf of a user without any possibility of being detected. This property makes identity-based systems difficult to use for signature generation. The PKG has to be highly trusted to compensate for the key escrow property.

## 3.5 System architecture

This section identifies and defines the different components that has to be in place in an VoIP system that uses an IBE version of S/MIME for securing the SIP signaling.

Consider the following scenario (figure 8): *Alice* is an user in *Network A* and she want's to talk to *Bob* that is an user in *Network B*. Network A and Network B are separate networks that are interconnected through gateways/proxies. The domain name for *Network A* is *atlanta.com* and *biloxi.com* for *Network B*. Internet is used as a transport medium for traffic between the different domains. Each SIP message is routed through 1 or more nodes on the Internet. These nodes are not supervised by service providers in Network A or B and can not be considered to be secure.

Alice and Bob have their VoIP phone that can be a hardware device or a softphone. Each device can be connected to the local network with a cable or by wireless transmission.

Each network contains a Registrar and a Location server that contains information about each user within a network. These servers typically contains registration and authentication information together with ip addresses and status for each User Agent. Alice is registered in domain *atlanta.com* and Bob is registered in *biloxi.com*. Device ip address is stored in the location server when a phone is connects to the local network.

New component introduced is a Private Key Generator (PKG) server with an attached database. The PKG is responsible for delivering IBE system parameters and for calculat-

ing public and private keys for local users. Parameters and keys are stored in attached database. There should be at least one PKG in each provider network.

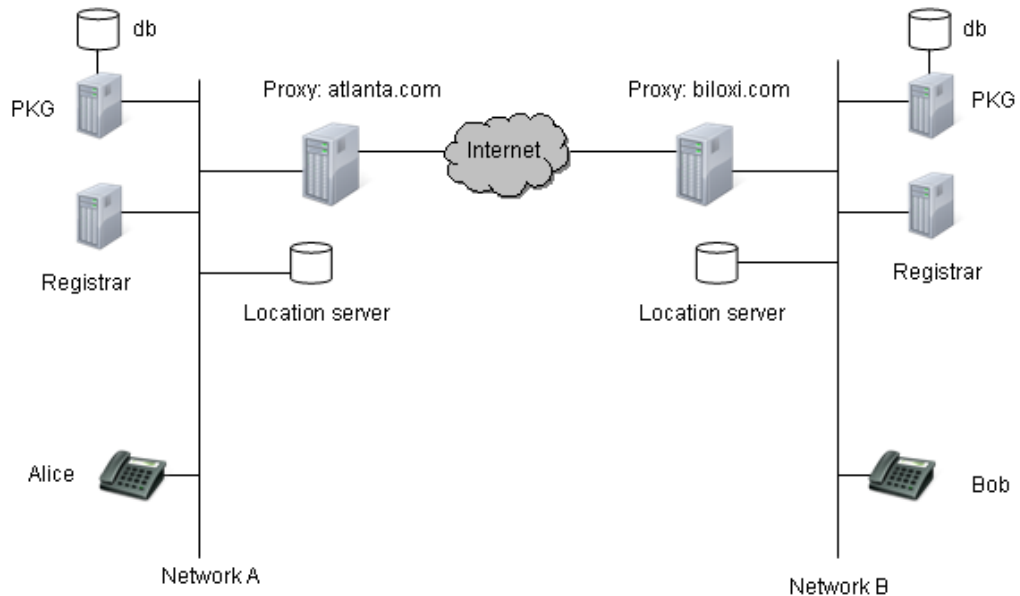


Figure 8: System architecture

An User Agent has to communicate with the PKG to get these system parameters and to get public and private keys. This requires that a new protocol is defined for the communication between an UA and a PKG.

### 3.5.1 Protocol design

The following methods are identified in the protocol between a UA and the local PKG.

- Login user. User needs to be authenticated before executing other commands.
- Logout user. Logout user from PKG server.
- Get system parameters. A phone that is connected to a network for the first time need to get system parameters used by the IBE scheme. There may also be necessary to get updates from time to time from the local PKG.
- Get public key. Get public key for receiving provider.
- Get private key. Get private key for UA.

Communication between a UA and a PKG should if possible be encrypted to prevent an attacker from eavesdropping on the communication that may contain sensitive information. This is normally done by adding a server certificate to the PKG and will also be used to identify the PKG server to the clients.

### Login user

The PKG contains sensitive information like a provider master key that is used to generate all other keys. The PKG may also contain all private keys calculated and public keys for different service providers. The PKG server must be placed in a secure zone and must not be available from remote networks or the Internet.

Only users that are registered and has proved its identity should be able to communicate with the PKG. This indicates that some kind of authentication method must be used. It is however up to the different providers to decide level of security in PKG. Conceivable authentication methods are username/password or a smart cards with identity information/certificate that proves a users identity. An UA authenticate it self by providing necessary credential, figure9. Result of the authentication is returned and a session between UA and PKG is created.

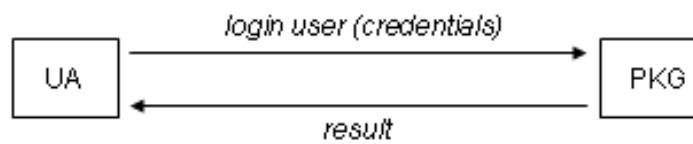


Figure 9: Login user to PKG

A user that has logged in should stay active for a predefined time of period to be able to run additional queries.

### Logout user

A user that signs out should also be logged out from the PKG server to prevent misuse. This is done by the UA by running a logout command, figure10. Result from command is returned to UA.

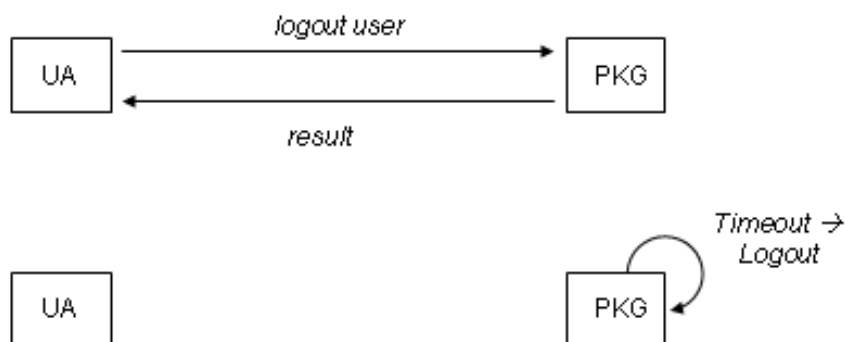


Figure 10: Logout user from PKG

Users should automatically be logged out after a predefined time of inactivity.

### Get system parameters

A user that want's to encrypt a message needs to get the system parameters from the local PKG, figure 11. These parameters varies for different IBE algorithms but a PKG must return all parameters needed for encryption/decryption of messages.

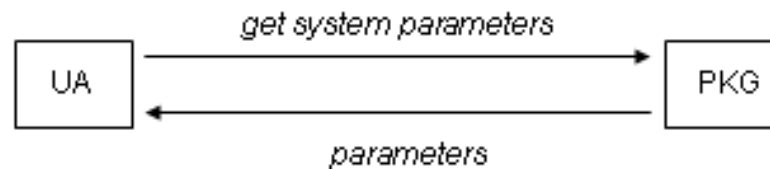


Figure 11: Get system parameters from PKG

These parameters are static and are the same for all service providers using the same IBE algorithm and are only needed to fetched once. A UA should then store these parameters locally and use them together with provider specific information when encrypting/decrypting messages.

### Get public key

The public key is in this case a service provider specific key. This key should, when encrypting an message or when validating a signature, be used together with the string that identifies the recipient of a message. The combination service a provider public key and identifier string must globally point out the message recipient. Only the recipient with corresponding private key is able decrypt this message.

An UA ask the local PKG for a service provider specific key when accounting this provider for the first time, figure12. A public key is returned and should be stored locally by the UA for further use.

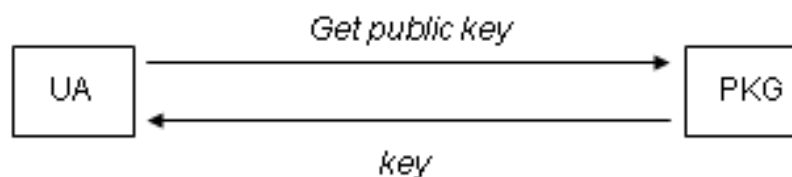


Figure 12: Get public key for service provider

### Get private key

A user that wants to decrypt a received message or sign a message to send, needs a private key. This key is calculated by the PKG on request from an UA, figure13. Needed input is at least an identifier string that identifies requesting user. The private key is returned to the UA and should be safely stored for further use. This key should only be known to the owner and to the PKG.

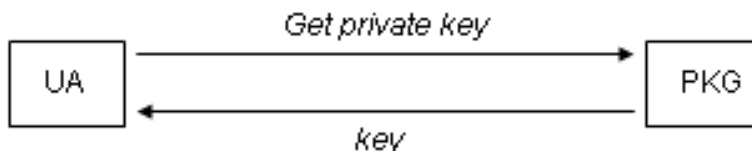


Figure 13: Get private key from PKG

### 3.5.2 User friendly

A security solution in VoIP system must be very easy to use as most people know very little about VoIP and information security. VoIP services of today mostly have a non existing or in best case a weak security solution that probably is based on proprietary technology.

The introduction of a certificate based PKI solution would improve the security but must be considered to be hard to use for ordinary people. PKI solutions requires that a user retrieves a certificate with a public key for the recipient prior to the call setup. This certificate has to be stored locally and updated when certificated expires.

Identity Based Cryptography will make the security solution much easier to use to the end user as no extra effort is required. Information needed by the IBE system is fetch from the local PKG automatically and the recipient SIP address is used as an identity string.

This would indicate that this almost is a *zero effort* solution. Almost as user initially must add their credentials to the phone. The credentials should be distributed by service providers when new users are registered.

The only thing a user needs to do after that is to decide if the call should be secured or not.

## 4 Prototype

A prototype is developed to investigate if the suggested design works and if it fulfills the availability and user friendliness requirements. The prototype should focus on central part in this MSc. Shortcuts may be made if they are of a less important nature and not inflict on the result.

The prototype should if possible implement a realistic VoIP scenario that indicates if the mixture of IBE, S/MIME and SIP is a good combination in practice. The prototype should focus on securing the SIP signaling between two User Agents. No work is put into securing the speech data as this must be considered to be solved by others.

The result from this work should be a trace of the SIP signaling showing session establishment and session tear down. Time used for encryption and decryption of messages should also be presented.

### 4.1 Development plan

The following list shows a overall list of activities that are going to be made in the prototype work.

No	Activity	Comment
1	Start prototype design	Define and describe the prototype
2	Find Soft phone software	Search for and test different soft phone software. Decide which one to use
3	Find proxy/registrar software	Search for and test different proxy/registrar software. Decide which one to use
4	Design IBE S/MIME	Decide which IBE algorithm to use and design an simplified IBE version of S/MIME
5	Find or develop IBE software	Search for and test IBE software. Decide if it is usable and develop own version if no one is found
6	Soft phone development	Integrate IBE and simplified S/MIME software into soft phone
7	Develop IBE PKG	Develop an IBE PKG if time
8	Test software and measure performance	Test if software behaves as expected and if performance is acceptable

### 4.2 Prototype design

The prototype is only one implementations of many possible and the following design decisions has been made:

- The prototype should be a fully functional security solution for SIP signaling.

- The programming language Java should be used as far as possible.
- The VoIP phone is software that runs on a standard PC that is running a commonly used operating system, windows XP in this case.
- The proxy is software that is run on a dedicated server.
- Software used should if possible be Open Source.
- The Boneh-Franklin IBE algorithm is used for encryption and decryption of content encryption keys. This is however just one of several different IBE algorithms that can be used in such a system. A overview of different IBE algorithms can be found in the book *Introduction to Identity-Based encryption*[27].

The following restrictions has been made:

- The SIP messages are not signed as this will require more development time. It should however be possible to estimate the time added by the IBE part.
- S/MIME is not fully implemented as this requires a large amount of work and is beyond the scope of this MSc. A simplified version of S/MIME is implemented to support the IBE part. This implementation does not follow the S/MIME specification and should just be considered to be a *proof of concept*.
- The PKG is not implemented as this requires to much work. Parameters needed by the IBE software is read from a local file at start up. This should not affect the prototype as this can be considered to be the normal scenario.

The prototype consists of many software components that works together. Different open source component has been used as a base for the development. These component and design decisions made are described in some detail below.

### 4.3 SIP, Softphone and Proxy software

The prototype will be based on the open source software Mjsip. MjSip is available under the terms of the GNU GPL license (General Public License) as published by the Free Software Foundation. The MjSip has earlier been used in research activities by the department of Information Engineering at University of Parma and by the University of Roma "Tor Vergata".

Mjsip is a fully compliant Java implementation of the SIP specification and contains the complete layered stack architecture. The Mjsip website <http://www.mjsip.org/> also states the following:

Moreover it includes higher level interfaces for Call Control and User Agent implementations. MjSip comes with a core package implementation that includes:

- all standard SIP layers and components,
- various SIP extensions (already defined within IETF),
- some useful Call Control APIs (e.g. Call-Control, UserAgent, etc.),
- a reference implementation of some SIP systems (proxy servers and UAs).



A reference implementation of a simple but fully functional User Agent (UA) and a Proxy server is also provided by MjSip and will be used as a base for the Softphone and proxy server. The Mjsip implementation is described to some extent in [43].

#### 4.4 Usage of Boneh-Franklin IBE algorithm

There are many different Identity Based encryption algorithms available. The Cooks IBE, Boneh-Franklin IBE, Boneh-Boyen IBE and Sakai-Kasahara are all introduced in [27].

The natural choice in this master thesis is the Boneh-Franklin algorithm as it is well documented and reference implementation are available.

Note however that Boneh-Franklin is a so called full-domain hash IBE scheme and requires fairly expensive calculations compared to other IBE schemes. This would suggest that a commercial application should be based on other technology than full-domain hash.

A solution that is based on the Boneh-Franklin IBE algorithm needs to do the following:

- Decide which Elliptic curve to use
- Define the bit strength
- Generate system parameters
- Pick a master secret key
- Calculate public key
- Calculate private keys
- Encrypt and decrypt strings

The following parameters are common to all service providers

- chosen elliptic curve  $E$
- generated prime  $q$
- group  $G1$  and  $G2$  of order  $q$
- bilinear map  $\hat{e}$
- cryptographic hash function  $H1$
- cryptographic hash function  $H2$
- cryptographic hash function  $H3$
- cryptographic hash function  $H4$
- the length of plaintext in bits  $n$

The following is unique to each service provider:

- Picks random master secret  $s$ . This key is used to generate all other keys and must be

kept safe in the PKG.

- Randomly pick a point  $P$  on the elliptic curve  $E$ .
- Calculate a service provider specific public key with  $P_{pub} = sP$
- Distribute the service provider specific point  $P$  and calculated public key  $P_{pub}$  to other service providers.
- Calculate the private key  $d_{ID}$  for each user that has subscribed to this provider;  $d_{ID} = sQ_{ID}$  where  $Q_{ID} = H_1(ID)$ .  $d_{ID}$  must also be stored safely in the PKG.

More information about the Boneh-Franklin IBE algorithm, the *BasicIdent* and *FullIdent* scheme can be found in [24][27].

#### 4.5 Development of IBE software

The development of the IBE software are a bit complicated as it requires the usage of libraries for elliptic curves and pairing algorithms. The calculation that are needed to be made are intensive and requires much resources and can take some time.

The NUIM Computer Science Crypto Group has developed an Java based Identity Based Encryption provider[26][44] that can be used in this project. The IBE provider implements the Boneh-Franklin IBE algorithm together with an elliptic curve API. The IBE package make use of a super-singular curve on the form  $y^2 = x^3 + 1$ . More information about the IBE provider and elliptic curve API can be found in page <http://www.crypto.cs.nuim.ie/>. The Java code is freely provided and can be downloaded from this site.

The performance of the Java based provider is however not very good. Some preliminary test have been made with the following result:

- Encryption of a string "my secret message": 8844 ms
- Decryption of encrypted string: 6734 ms

These times vary depending on available resources.

The provider works fine but uses too much time. We have instead decided to implement an encryption and decryption routine in the C programming language where we use the MIRACL library. The MIRACL library can be used freely for academic, non-profit making or non-commercial use. This library also contains example C code for IBE encryption and decryption that we have used as a base and modified. This has resulted in two programs called *voipIBE\_encrypt.exe* and *voipIBE\_decrypt.exe*.

These routines are called from the Java code using the Java language feature *ProcessBuilder*. A singleton Java class called *IBECryptoUtil* is developed and implements encryption and decryption methods that calls the C code. These methods are also responsible for setting the system parameters, sending in parameters and handling the result from the C code.

The encryption method takes the following input:

- A Public key to a recipient's service provider. This key should be fetched from the local PKG.

- A string identifying an User Agent. This string is in this case the SIP address.
- The string to encrypt. This string has a maximum length that corresponds to the hash algorithm used that in this case is SHA-1. SHA-1 produces an output of 160 bits or 20 bytes.

The C code returns the  $\langle U, V, W \rangle$  parameters that represents the encrypted text. These parameters are returned as strings.

The decryption method takes the following input:

- A Private key to the recipient. This key has to be fetched from the local PKG in advance.
- The  $\langle U, V, W \rangle$  parameter representing the encrypted string.

The output from this method is a string with the decrypted message.

The IBE implementation, that is based on the MIRACL library, uses the super singular curve  $y^2 = x^3 + 1$  with a 512 bits prime.

## 4.6 S/MIME replacement

Development of an IBE version of S/MIME is a big task and is not the scope of this MSc. An S/MIME alike solution must however be developed to be able to test the design and to get an fully functional VoIP prototype.

The S/MIME replacement works as S/MIME but does not follow the specification when it comes to details. The big picture is however this: The information that is going to be encrypted is transformed into a string. A content encryption key (or session key) is generated. This string is then encrypted with suitable symmetric key encryption algorithm and a newly generated key.

The content encryption key must be known by both parts and is encrypted using Identity Based Encryption. The encrypted key is then attached to the encrypted message body, figure14.

Message is then sent to the recipient that knows the structure of the message. The encrypted content encryption key is extracted and decrypted with the private key of the recipient using IBE. Message body is then decrypted and the original message is returned in clear text, figure15.

The content encryption key is stored by both parts and used in the rest of the communication within this session.

We have decided to use Advanced Encryption Standard (AES) for symmetric encryption with 128 bits keys. The key length is however configurable and allowing a developer to chose length.

The AES code is placed in a Java class called *AESCryptoUtil* that is a singleton. This class has several methods for encrypting or decrypting strings.

The class *SmimeUtil* encapsulate the SMIME alike functionality and uses *IBECryptoUtil* for public key encryption and *AESCryptoUtil* for the symmetric encryption.

S/MIME, IBE and AES code is placed in package *org.zoolu.sip.smime*.

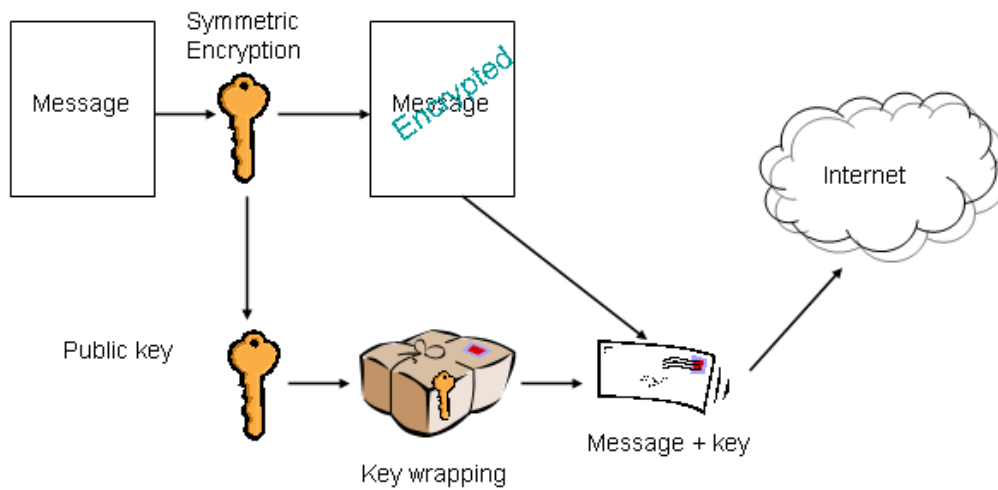


Figure 14: Encrypt message

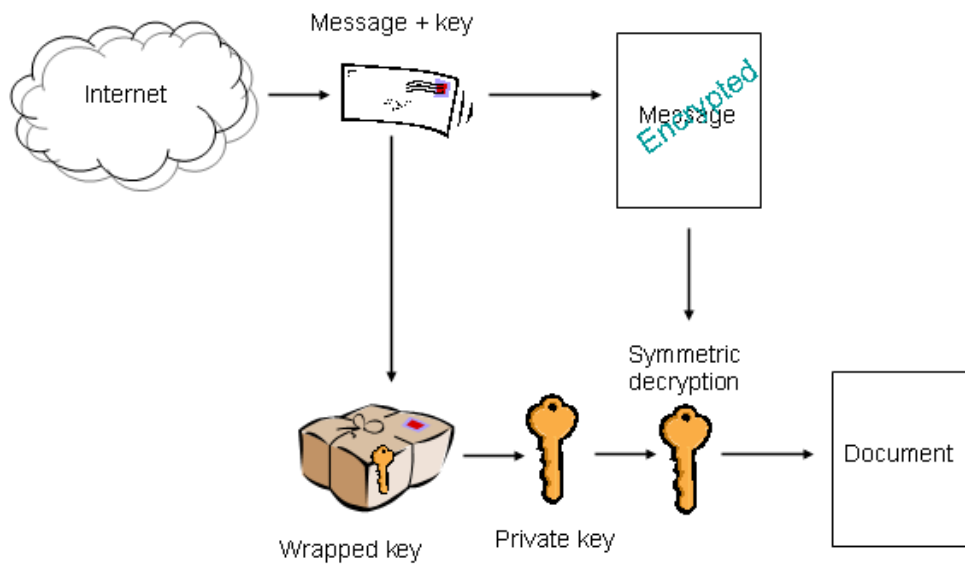


Figure 15: Decrypt message

#### 4.7 Changes made to soft phone software

Mjsip is, as stated before, a Java implementation of the SIP stack. The Sip signaling is however not secured in any way. This is an excellent base for further development and introduction of IBE, AES software.

The IBE and AES software has been integrated to the soft phone where the Java code is placed in package *org.zoolu.sip.smime*. C code an needed libraries are also integrated

and placed in the *clib* catalog.

The *org.zoolu.sip.smime* classes are then used to secure the following SIP messages:

- INVITE
- Trying
- Ringing
- OK
- ACK
- BYE

These messages are all included in an ordinary SIP session as described in the Literature survey.

The configuration of the software is found in appendix A. A description for building and running the software is found in appendix B and C.

#### 4.8 Test scenarios

The test scenario is based on set up and tear down of a call between two participants that runs the softphone software on a ordinary personal computer available to most people today. No extra resources (like memory and CPU) are added on the machines used. The three machines used have the following characteristics:

*PC1:*

- System: Windows XP Home edition 2002 SP3
- CPU: Pentium 4, 3.00 GHz
- Memory: 1.00 GB RAM
- Connection: LAN

*PC2:*

- System: Windows XP Professional 2002 SP2
- CPU: Pentium M, 1.7 GHz
- Memory: 1.00 GB RAM
- Connection: WLAN

*PC3:*

- System: Windows XP professional 2002 SP3
- CPU: Pentium 4, 1.66 GHz
- Memory: 1.99 GB RAM
- Connection: WLAN

The prototype consists of two softphones with integrated IBE support that are placed on separate pc:s. The users of the soft phones are referred to as *Alice* and *Bob* and they are registered in the domain *atlanta.com*. The proxy/registrar and location server are

executed on the same machine. The prototype deployment is shown in figure16.

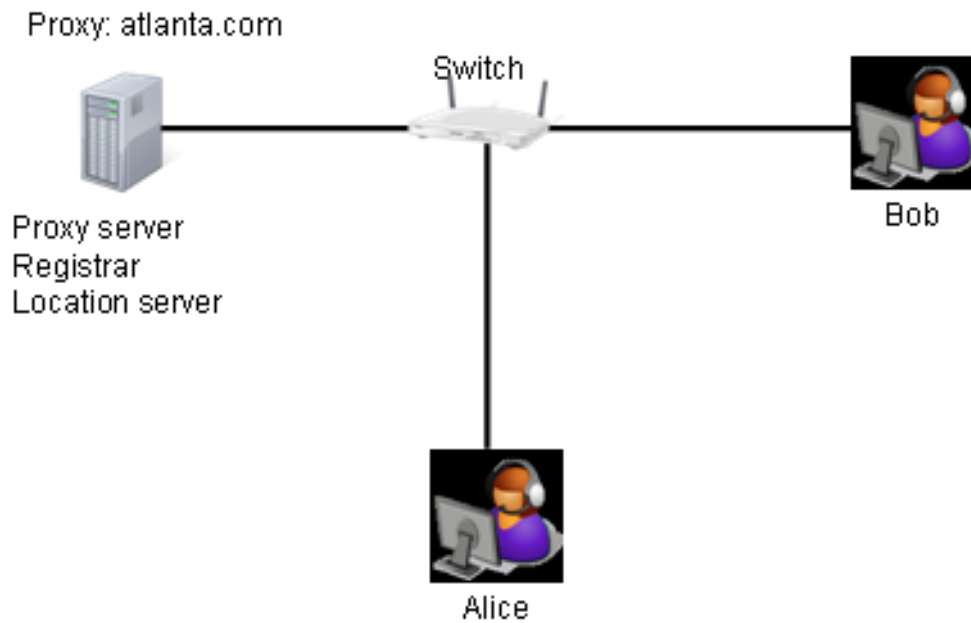


Figure 16: Deployment in prototype

The test scenario should indicate whatever proposed solution works and if it is simple enough to use for non technical users.

#### 4.8.1 Test scenario 1: Proof of Concept

As more and more people works on wireless solutions today it is interesting to test proposed solution in this environment. The proxy/registrar and the location server is placed on PC1 that is connected to a WLAN switch with a cable. User Agent Alice is placed on PC2 and Bob is placed on PC3 and these machines are connected to the WLAN switch with a wireless connection as shown in figure 17. The messages should in this case also be secured secured when transmitted wireless (WLAN).

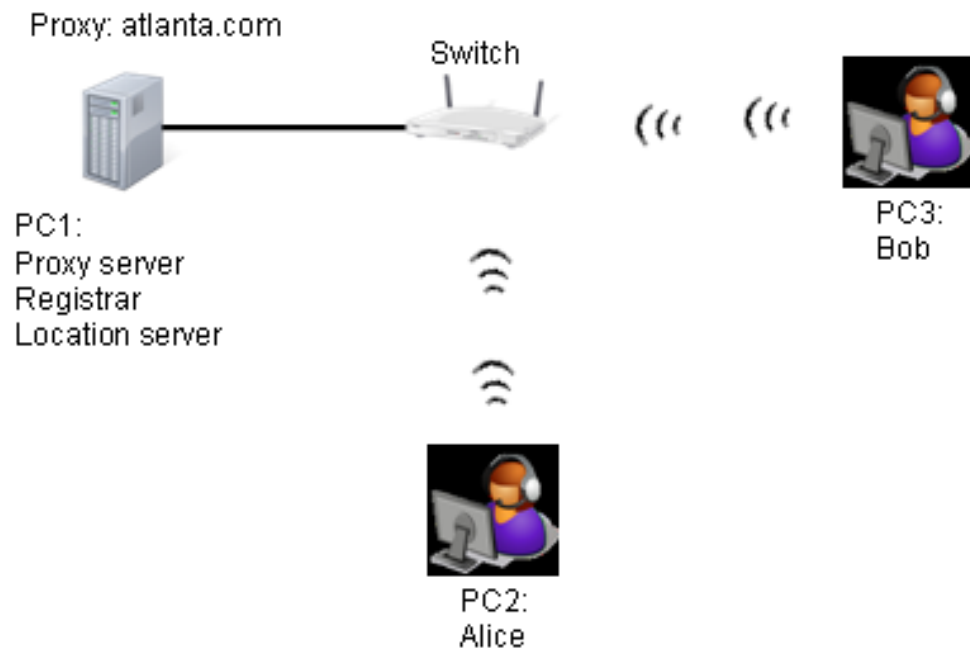


Figure 17: Test scenario

The Server and User Agents are first built on a development machine as described in appendix B and distributed to each machine in the test scenario. The Server and UAs are then configured as shown in appendix A. A Java Runtime Environment (JRE 1.6 or later) must also be available on each machine. The test is then run as described in Appendix C.

#### 4.8.2 Test scenario 2: Measuring signaling overhead

It is also interesting to measure the overhead in signaling when IBE and AES are in use.

This test will measure the total time and the amount of signaling data in call setup for an unsecured and secured connection.

The same test deployment is used where Alice is placed on PC2 and Bob is placed on PC3. PC1 is again used as a proxy that also contains the registrar and location server. PC1 is connected to the switch with a LAN cable and PC2 and PC3 with WLAN connections, figure 17.

## 4.9 Test result

The network protocol analyzer *Wireshark* is used to analysis the SIP signaling in the test scenario. *Wireshark*[45] is licensed under the GNU General Public License and can be freely downloaded from the Internet from <http://www.wireshark.org/>.

The timing presented below is made with the Java command *System.currentTimeMillis()*; and some calculations that are integrated into the softphone application.

All tests are run on a network with very little traffic.

### 4.9.1 Result from test scenario 1

The following shows the SIP signaling for the call setup test run that is secured with AES and IBE. A new SIP message is created for each message that is going to be encrypted and only information needed in routing is available in the new header part. The original message is encrypted and placed in the body of the new message. The encrypted content encryption key is added to the first message body and sent to the receiver. This key is only sent in the first message and all parts in the communication stores this key for the entire session. A new key is generated for each session.

The result from a test run is shown in the following trace.

```
INVITE sip:bob@atlanta.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.36:5070;rport;branch=z9hG4bK33765
Max-Forwards: 70
To: "Bob" <sip:bob@atlanta.com>
From: "Anonymous" <sip:Anonymous@atlanta.com>;tag=z9hG4bK38306281
Call-ID: 779333070336@192.168.1.36
CSeq: 1 INVITE
Contact: <sip:192.168.1.36:5070>
Expires: 3600
User-Agent: mjsip stack 1.6
Content-Length: 1045
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m
```

```
Nz1FREE3MjU1MTQ5QUVDRjdCQkRFNORGNEJCOEVFMTc2NOM2MTUxMjg3MUNDMOE1MzRERDdDMjM2
RkYzQTFEND4NjM4NzNmMzRBM0YzNjBEMzJGNTIyRUQyNDg5NkI4Q0EOMOUzQUE2MzA3RDkyMkE3
QTAYNjZFYTYOEZGNEMgMjNCRtAYMTVCMEI2QzA3ODYxRTQ4RDBFNDg4QTVBRkYzRTIxNjU5RiAZ
QUEyRUFBMDRFQjNGMjkyRDM2MjJCNTBERUYyQ0ZBQTNENEVCQzQ5IA==
KYcWSHb2fsSWoUP9QCcbdfKmODqNuKgaRaBKGTb9Wf00rw2tB1k1amZUffd4L3/a2Af2ulcvFDC0
1b8qWrp4p4SqbKWsJFVJDHrWfjv0msEYr2vugrFV/16JcYPNTB81XguvdJlkpycEmrtpAL6crEmR
vrrpJQ214RN7cLQbvUewvKcArY8S6bpMng4/WxX1UxNcCCvFxGdl6vzUdgdAZq9EJMW+F2L9Kn18w
uPRfkCmOFNthMFy4K7tIxIh9ppzrmCghri0tHPiBulnkdVtiXJUXzs95S20p4uhS2/BercYqHuwV
tL9djiq4pcckQabQ7WHvX7hfbeE/fEENAFiuew5wCR+DNThG1YoNQUkrNvIL4MYyLc+ZJ7QhRrP6
V3/pecV7GrA9pP4MxhQ4r25c9Hu0mi95KoYcahsS0yWqYHUHInAtmROBKqDqxP6NFDpQUjnX1MSW
Ixr9TgJJP7L5clan9StIpt0XZagPIEAenKmMJA7DkbjmvNm0Udwwu718ZKid5vylj0ic/See3n8M
eH7xvpCy+V2fWTi6J5+PQQheVpI5ct/odGIL3XQrXeKYh3w4pXtitHRTN/ZQz4jQ01cBhKJp5Gg5
mHrB1PLV9uxp1UR8sQeMb1odHMoyt2sDb6Vu0EFVJ/RfJbq1D0NstLHbf13HWuxbRsXRn0QtvPLi
5TuLRUIq07kX0Gion/AAZCmvWr5GYrVy1/j0Mfc2txkE5iSPFhlCg69mA70cYcg=
```



SIP/2.0 100 Trying  
Via: SIP/2.0/UDP 192.168.1.36:5070;branch=z9hG4bK33765;rport=5070  
To: "Bob" <sip:bob@atlanta.com>  
From: "Anonymous" <sip:Anonymous@atlanta.com>  
Call-ID: 779333070336@192.168.1.36  
CSeq: 1 INVITE  
Server: mjsip stack 1.6  
Content-Length: 498  
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

X78adXj4mUBGwg9hhT3MoHwSbxqM7ZqJW750gDuzGkLYB/a6Vy8UMI7VvypauninIcpmkcTExncB  
ZsSH251cYK/6MZm3NeSchBBL7ah6nYhdGJFfHrwwbDI+PXwEYrnz7kg++OLW+ayUSEMNZNjQJVnf  
6lwFqoxA84W+g5iUtkSWKjgYX0d34J2UWjuGqEVE/4Hcno4/IFWnq/0mVmo3Q0Ki8DUukMqu84C4  
uwWnVRncQwAZ16UHt/Zw+ozJqQJ/i9Zh/C9xe2GC0sWtoNVfR90NAzPVA9VZ2PvVnSy4BSWZGI9Y  
/gnlN1axhYHY785oZXdiI3w8ivmqmHUUYf6wju4oUy0bizrkt5ZcFvcnEbUA2/uCo6seHys7YgVX  
WzkERH+jAC7S3w2m5FYWDhLgrcjKswtAo6jwaHvjdn4ngeAXMfVuS40/HCWQ5cgwJT0jGsXoS1J  
UZyW9mCdkPeLSm90kgvt1kIVBCfm0/GdfMw=

SIP/2.0 180 Ringing  
Via: SIP/2.0/UDP 192.168.1.36:5070;branch=z9hG4bK33765;rport=5070  
To: "Bob" <sip:bob@atlanta.com>  
From: "Anonymous" <sip:Anonymous@atlanta.com>  
Call-ID: 779333070336@192.168.1.36  
CSeq: 1 INVITE  
Server: mjsip stack 1.6  
Content-Length: 498  
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

NzvoTKkI1Ebu+Ck3mRP9wLv9zwL9+so+EtaB0k8vj8XT9WxSi98deU0I Jpzc9fkU11k5VUIJelk  
sj/d4rmIhx6M4FPI/V6IpnZo3HfFQDJG269qqsKLbsuKECLg5fLFN1RCLraqYlq8hMBQ9swj3Vuk  
032gCw2NcPXQow77V+LV1//Wlp12l+yyy0xmSOSPT5NgpCABs/hcJv99oIVp9VomEMMX204krE5F  
phUe5hPFjpcI82SRV54paw+u6S0I1WTE5+FMCRkyjbKfW763/djDNuT3ZtDIy2mZFUbMFteVpI5  
ct/odGIL3XQrXeKYAS7f1ALpGpBtfYl6qKpINPQGCLjys2TI6dj sm5B8dVeKngyZIFRDNc6+fc9J  
2AH/KCuswfYxXH7hohzsmM4oPQw7zZaJ1xcruAMNXrRR6jWJ3Lksd5PifivW3k9SwSwthKfhMMe4  
fCt6FLXuq1Qd/b+Ye2xolBVwIbzgfFSj4SE=

SIP/2.0 200 OK  
Via: SIP/2.0/UDP 192.168.1.36:5070;branch=z9hG4bK33765;rport=5070  
To: "Bob" <sip:bob@atlanta.com>;tag=73b296d98ec846bc  
From: "Anonymous" <sip:Anonymous@atlanta.com>  
Call-ID: 779333070336@192.168.1.36  
CSeq: 1 INVITE  
Contact: <sip:192.168.1.37:5080>

Server: mjsip stack 1.6  
Content-Length: 822  
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

hcH0nyWl1HDnySHWuD/Kk3TdBOZWTpPCByQieQa4SCPA4ajYVdNVashncnuRFHpSzprCVunCxRcv  
EhL06VjmxDZfwQ92BBFaGuGu+BmzhmGE1Q5ImnlOCjnIQUUsSap6TuhKRM7jEmaHx6M8W+fHZEE0  
ox4FJZ86VqvlhzTXe9GShgkZzThfqZekzRbRg/aniAl7zWzMC1LcqMrjZZnr4ibQzYDbU79CzKsT  
YQvev7/CvrQ+mJHRVPQj/2swVgpxxFqQNV4oFsc7seSoIy/ULINLbdub1d4mS1CYX1ac8ggFCDe0  
IOXr95P8H+MkzXyJyZGAzcY3wS4b5V/qlSGpZMyFpKcye/T4YYkwf+dRZBfXi1td14uxvAK+FzV/  
3It+dpDQtISyB4WqlmFj7Dg24o2rgX9Ietk9JgpC2NZFY80yULBJARTEBzi+xB+FskGFJ+B4p9A  
ZScXbyIQGUD5oKoC6+1T+ZPDwbBil1YExdV6u49taX0EhCmt6sEqMk/du6xRsyKCqQaz6dQJkI6v  
tgmpYKEOHslcPZ7/s80qSa9n6UFyb59mWZasEoLq+PMSa+XNdyABaFY+ZkPJlGSQ2WH84ruAKSnK  
1B2fpnN6EIXS2K59K49s7YZ0Ie2BdoBxi840Jjb194CQCQ3o1ogW36C0Lwsj3UkNX60p4RbeX04  
me0V19n3d0APgcJPKwCcqx2fklGmjAZFoDyATA3GV0W0CNJXNrbdoUnNLE13NnZTZHyoaTRMFsn  
DsL1uUtZRa90zrp6AT+VC2EdRRfxIpFPJV55aBwQP4sc90MKipE=

ACK sip:192.168.1.37:5080 SIP/2.0  
Via: SIP/2.0/UDP 192.168.1.36:5070;rport;branch=z9hG4bK06595  
Max-Forwards: 70  
To: "Bob" <sip:bob@atlanta.com>;tag=73b296d98ec846bc  
From: "Anonymous" <sip:Anonymous@atlanta.com>;tag=z9hG4bK38306281  
Call-ID: 779333070336@192.168.1.36  
CSeq: 1 ACK  
Contact: <sip:sip:192.168.1.36:5070>  
Expires: 3600  
User-Agent: mjsip stack 1.6  
Content-Length: 543  
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

K3UhZo/tnzR5xoeH9KxmUJ7RUn/B18NNLq/wAmC/36yz7IvANdhk0JyPaf2MtqSQ0dGNay874Qn  
WSa25LtVfj8WrRuiTRQmcOnqgrGcmHoKv0NjoJR3AkQUqVzK9Em0H5VZGdqkY/5Jxq6VKHh2Fh/4  
jo2MfK42N8nWZfaAZx8KYsxwJ5jGhBraJd0sG1TiV99iKIAMBMeqCKCdY9jZpu/gRSwCQH7a4ENJ  
pQS3WrFR6R1B/L1OpaffwUYZXschVLOSD56fUaEKuw9g4D5wLo0UGqWExnP8XQb2T+cArXmZbSES  
20A5VzuZZgrzZkvFoswmNQCwclF1uch1QLZE2ky8tfdh5JoU1BVF38EAfg5vQMMKKU1PLICDOLTy  
V4osDL5qR0LPLs1aXLI0bs1vuFi+rqlmuIjBFYpLy0PuvRpVARYAvLoZAxp4ucVmmdlbFk47N46R  
ovz3lQfzW644rmtDdbECzxtSQRTAk22RALDAXXfeWuLDEVbM/ynytL6dUFChvN1tfnCfSVuiaJg0  
kg==

**The following shows the SIP signaling for the tear down of the call in the test run.  
The original message is again encrypted and placed in a new SIP message body.**

BYE sip:alice@192.168.1.36:5070 SIP/2.0  
Via: SIP/2.0/UDP 192.168.1.33:5060;rport;branch=z9hG4bKe5587a6ccf0d6ce870a3b2ba  
Via: SIP/2.0/UDP 192.168.1.37:5080;branch=z9hG4bK77025;rport=5080

Max-Forwards: 69  
 To: "Alice" <sip:alice@atlanta.com>;tag=z9hG4bK38306281  
 From: "Anonymous" <sip:Anonymous@atlanta.com>;tag=73b296d98ec846bc  
 Call-ID: 779333070336@192.168.1.36  
 CSeq: 1 BYE  
 User-Agent: mjsip stack 1.6  
 Content-Length: 518  
 Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

6azreiWNu1soupLlFM+aDHPjVhdpsVslduVTlXpPHhwcSsXfmDSVEMzcjsNrsj7s4ZxvKA0I6Mv3  
 KQZQ5X0qoGF9LPDpkn40QHxqJr5UXnj1adG4LxwdHVS0c22qv3SEbHIxYeRc7+uIhiLaAjsJ60S0  
 nnhpW7qnmvQGQr7NTqTGF/2auuxXmPWP+FmkmbP/WxrL0QuYyOh8dKBCiQQ6XErFBehWz2Zd2Dpb  
 B/dmvpJR2S0+EFCYcmRMSk20hT1/LynAK2PEum6TJ40P1sV5VCQzUz68GprpPz793VY0xZhJk4Ac  
 mrYwmGXNZ7ucYRngxip4MmSBUQzX0vn3PSdgB/ygrrMH2MVx+4aIc7Jj0KD2fh0JpXacKb3S9M/Gu  
 hjIGkerZqQotZDZ56uEtqW4cv5JvijJz3ijEC7m4ccW0FU82FmaEs2sPr7DcqemymJntUyHfdAey  
 rEWLoPq2WkRaNXKMWSyv5xsugEPNePjwsPRNpnhLZKWrkPAkKeWZeKz

SIP/2.0 200 OK  
 Via: SIP/2.0/UDP 192.168.1.33:5060;branch=z9hG4bKe5587a6ccf0d6ce870a3b2ba;rport=5060  
 Via: SIP/2.0/UDP 192.168.1.37:5080;branch=z9hG4bK77025;rport=5080  
 To: "Alice" <sip:alice@atlanta.com>;tag=z9hG4bK38306281  
 From: "Anonymous" <sip:Anonymous@atlanta.com>  
 Call-ID: 779333070336@192.168.1.36  
 CSeq: 1 BYE  
 Server: mjsip stack 1.6  
 Content-Length: 518  
 Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

hcH0nyWI1HDnySHWuD/Kk3TdBOZWTpPCByQieQa4SCPA4ajYVdNVashncnuRFHpSzprCVunCxRcv  
 EhL06VjmxGK1pFHoFyICYRpoEhLpdL7c8WDQNwXwqTFsDYXkv1oITuhKRM7jEmaHx6M8W+fHZEE0  
 ox4FJZ86VqvlhzTYe9GXyMoP2KRQbqScVH5B3jJc4k6fJ1x+qRBTeh0Ppj0v1wlf6Cd8h0tE3F7iK  
 +tsd2ILGF/2auuxXmPWP+FmkmbP/WxrL0QuYyOh8dKBCiQQ6XErFBehWz2Zd2DpbB/dmvpJR2S0+  
 EFCYcmRMSk20hT1/LynAK2PEum6TJ40P1sV5VCQzUz68GprpPz793VY0xZhJk4AcmrYwmGXNZ7uc  
 YRngxip4MmSBUQzX0vn3PSdgB/ygrrMH2MVx+4aIc7Jj0KD08pJPeGAwGq1fCRMGXuw17KLT7viRd  
 u902t2W6ckYY/KYe9+hGSHSAr10E3gQtliH3oV1QNG5Ej4tm7T/RYZEu

The time used in the IBE encryption, IBE decryption, AES encryption, AES decryption and the total overhead of securing the signaling in 10 different test runs are shown in the tables below.

Alice on PC2 is calling Bob on PC3. Traffic is routed through the proxy on PC1. Table 1 shows the time (in milliseconds) used by Alice in the AES and IBE encryption and the total time used in the encryption classes. The timing includes processing of input, encryption and processing of result. It does not include initiation of AES and IBE algorithms as this is done when the softphone is started.

Table 2 shows the time used by Bob to decrypt the message that is encrypted by Alice.

Test No	AES Encrypt(ms)	IBE Encrypt(ms)	Total Encrypt(ms)
1	50	101	601
2	30	91	571
3	30	90	591
4	40	90	571
5	40	100	651

Table 1: Alice on PC2 is calling Bob on PC3, time used in encryption on PC2

The decryption time is divided into AES decryption, IBE decryption and total time used in the decryption classes.

Test No	AES Decrypt(ms)	IBE Decrypt(ms)	Total Decrypt(ms)
1	63	156	861
2	63	94	485
3	62	78	437
4	62	63	437
5	47	78	438

Table 2: Alice on PC2 is calling Bob on PC3, time used in decryption on PC3

Alice configuration now installed on PC3 and Bob configuration on PC. This is done to be able to compare different PCs. Traffic is again routed through the proxy on PC1. Table 3 shows the time (in milliseconds) used by Alice in the AES and IBE encryption and the total time used in the encryption classes.

Test No	AES Encrypt(ms)	IBE Encrypt(ms)	Total Encrypt(ms)
1	63	109	601
2	46	63	531
3	62	78	484
4	62	78	515
5	63	62	547

Table 3: Alice on PC3 is calling Bob on PC2, time used in encryption on PC3

Table 4 shows the time used by Bob to decrypt the message encrypted by Alice. The decryption time is divided into AES decryption, IBE decryption and total time used in the decryption classes.

Test No	AES Decrypt(ms)	IBE Decrypt(ms)	Total Decrypt(ms)
1	40	120	761
2	40	91	541
3	40	80	581
4	40	80	631
5	60	120	510

Table 4: Alice on PC3 is calling Bob on PC2, time used in decryption on PC2

#### 4.9.2 Result from test scenario 2

The total size are measured by recording the traffic on the network with Wireshark[45], figure 18, and calculating the number of bytes within a session. The messages included in the calculation are the ones that are used in a call setup: *INVITE*, *Trying*, *Ringng*, *OK*, *ACK*, *BYE* and *OK* number 2.

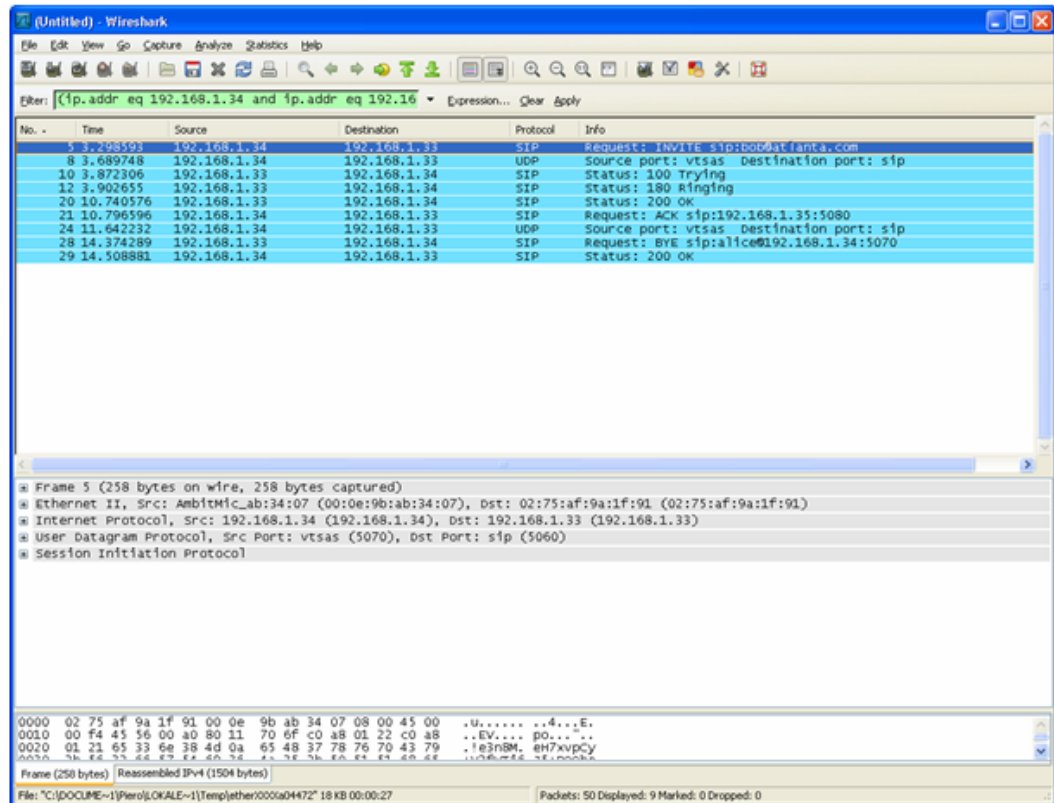


Figure 18: Using Wireshark to record network traffic

Timing statements in the Java code are used to get the timing for the setup. Timing is seen from the initiators point of view, i.e, the time it takes from that Alice pushes the call button until she gets a ring tone plus the time it takes after Bob has pressed the answer button until the call is established. Timing includes the following SIP messages: *INVITE*, *Trying*, *Ringng*, *OK* and *ACK*.

The test scenario is divided into two parts, one without security and one with IBE security. The result is presented in table 5 and in table 6.

Part1: The signaling is in this case unsecured and message can be eavesdropped or changed by an attacker. The total size of the messages within a session and the time used in the setup are measured and shown in table 5.

Test No	Total Message size(Bytes)	Setup time(ms)
1	2666	182
2	2670	254
3	2668	157
4	2668	166
5	2668	178

Table 5: Alice on PC2 is calling Bob on PC3, Signaling for a unsecured connection

Part2: Confidentiality is then added to the message with IBE and AES. Only a receiver with the corresponding private key will be able to read the original message. The total size of the messages within a session and the time used in the setup are measured and shown in table 6.

Test No	Total Message size(Bytes)	Setup time(ms)
1	7429	1347
2	7429	1438
3	7429	1145
4	7429	1411
5	7429	1408

Table 6: Alice on PC2 is calling Bob on PC3, Signaling for a secured connection

## 5 Discussion

This chapter will discuss the design proposed in chapter 3, Securing the VoIP Signaling, in regards to the stated requirements in section 3.1. It will also discuss the prototype and the test result in section 4.9.

This MSc proposes a security solution to VoIP that is based on a development of an extension to S/MIME. This new version has integrated support for public key cryptography that is based on Identity Based Encryption. Content encryption is achieved by using a symmetric key encryption algorithm.

### 5.1 Requirements

*Requirement 1:* Confidentiality is provided by encrypting the original messages and add the result to a new message. The original message is tunneled within an S/MIME body. The encryption is done with an content encryption algorithm (AES in the prototype) where the key used is encrypted with IBE and attached to encrypted content. This provides confidentiality to the original message where only parameters used in routing is readable on eavesdropping.

This requirement must be considered to be fulfilled by proposed solution.

*Requirement 2:* Message integrity is provided by adding a signature to each message in the dialog. This will protect the messages for unauthorized changes by an adversary. The signature includes the information in the original message plus header parameters that are not changed in transport.

Identity Based Cryptography inherits key escrow where the PKG can sign a message on behalf of a user without any possibility of being detected. This property makes identity-based systems difficult to use for signature generation. The PKG has to be highly trusted to compensate for the key escrow property.

This requirement is only partly fulfilled by proposed solution due to key escrow.

*Requirement 3:* Availability is an important requirement in VoIP. People expect that a VoIP telephone is just as available as their old pots telephone. This is however not very easy to achieve in VoIP as many new network components are involved. Routers, switches, modems and proxy-server often don't have the availability requirements that are needed. One must therefore expect that a VoIP solution is less available than a pots solution. Network components are however not in scope and this discussion will focus on the issues introduced by proposed solution.

The proposed solution requires that the PKG is available when new subscribers connects to a provider network for the first time. A new user needs to download IBE system parameters and getting it's private key. These parameters should be stored locally so that the solution can work even if the PKG is unavailable. The PKG is a central component in this solution with high availability requirements.

Both Confidentiality and Integrity will give some overhead in both processor time and data transfer time. It is however important that this overhead is kept small to get a

functional solution.

The test results from the prototype shows that the IBE encryption of a key takes ~100 ms on a standard PC. The decryption of this key also takes ~100 ms. IBE encrypt and decrypt is however only done in the first message when the content encryption key is exchanged. This means that IBE encrypt/decrypt will add ~200 ms in calculation time to each session. The signaling overhead is measured in test scenario 2 and the result is presented in 4.9.2.

The SIP messages are encrypted/decrypted with a symmetric key encryption algorithm that is much more effective than identity based encryption. The prototype uses a Java implementation of AES with 128 bit keys. Java is however not very effective when it comes to calculations and the time used by the AES can be shortened considerably.

There exist different identity-based signature schemes that can be used in conjunction with the Boneh-Franklin IBE algorithm. The result from the signing and verification process depends on the chosen signature scheme. The signature/verification process is not tested in the prototype. The Sakai et al.'s pairing based and identity based signature scheme [28] uses the same parameters, hash functions and bilinear mappings as the Boneh-Franklin IBE scheme. One can expect the same amount of time used in the signing process as in encryption.

The prototype is far from being optimized and one could expect the time used to encrypt to be considerably shorter in a commercial application.

This requirement must be considered to be fulfilled by the proposed solution and the prototype.

*Requirement 4: Non-repudiation of origin.* All messages are signed by the sender of the message and the signature will be used to verify the claimed identity. The key escrow property makes identity-based systems difficult to use for signature generation. The result from this is that the PKG has to be highly trusted.

This requirement is only partly fulfilled by the proposed solution due to key escrow.

*Requirement 5: Regulatory services:* It must be possible to tap the line for signaling and voice data. This is a requirement on service providers to provide legally sanctioned official access to private communications. Public and Private keys should after generation be stored in the PKG to archive this. IBE is said to have inherent key escrow. This also indicates that the PKG must be properly secured as this server contains sensitive data.

This requirement must be considered to be fulfilled by the proposed solution.

*Requirement 6:* A VoIP system is in itself very flexible as User Agents register their IP address when connected to a network. A user can in this way connect a telephone in any network that has a connection to the chosen provider and use it in a normal way. This will however require that the User Agent can be authenticated in a reliable way. This would suggest that a more secure solution than username and password should be used. Smartcards with stored user information may be a good solution to achieve this.

A properly authenticated UA should also be able to contact the PKG server and download needed information.

This requirement must be considered to be fulfilled by the proposed solution.



*Requirement 7 and 8:* Client and server authentication. Messages that contains information about sender and receiver are signed. This information together with a verified signature can be used to identify both client and server, i.e caller and callee. The key escrow property makes identity-based systems difficult to use for signature generation. The result from this is that the PKG has to be highly trusted.

This requirement is only partly fulfilled by proposed solution due to key escrow.

*Requirement 9:* S/MIME is chosen to introduce end to end security in the solution. This will secure the SIP messages from one UA to another. No other means of security should be needed to protect the signaling data. User must however be authenticated to be able to initiate an conversation.

This requirement is fulfilled by proposed solution.

*Requirement 10:* It is very important to introduce a secured solution that people with no or little knowledge of security actually can use. Many of the solution today fail on this point. The consequence of this is that VoIP today mostly are based on weak or proprietary security solutions.

A new security solution that is based on IBE will automatically handle the security of the signaling to the user. The design in chapter 3 and the test result in chapter 4 indicates that this actually is the case.

These requirements must be considered to be fulfilled by proposed solution.

## 5.2 Test result

The results presented in this section only applies to the configuration described in section 4.8 and will vary depending on chosen algorithms, implementation, programming language, hardware, traffic situation and more.

The SIP trace in test scenario 1 shows the result of the prototype. All messages sent between UA Alice and Bob are tunneled and the original messages are encrypted and placed in the S/MIME body. This result clearly shows that SIP messages can be encrypted with IBE and AES. Note that no information of the caller is revealed, making it hard for attackers to trace call patterns.

Information about the receiver is in the message header and is readable to everyone that analyzes a VoIP network. There is however hard to go around this problem and still use end to end security. A possible solution would be to anonymize the SIP address to the recipient. This will require that each provider encrypts the SIP address and adds a anonymous address in the header on messages that are sent out. The receiving provider must decrypt the original SIP address to be able to route the message to the recipient.

Test results indicates that the IBE encryption takes ~100 ms to execute on a ordinary PC. The total time used in a client is ~600 ms. The same amount of time is used on a server (recipient). This will give us a short delay i the call setup that will be ~1200 ms in the prototype. The prototype is however not optimized in any way and one can expect that this time used is a bit high.

Note that the signature part not is implemented in the prototype due to time limitations in the project.

The result in test scenario 2 shows that the total amount of bytes within a normal session in an unsecured solution is 2668 bytes against 7429 bytes in a solution where confidentiality is added. This means that ~2.8 times more signaling data has to be transferred in a solution when confidentiality is added.

## 6 Conclusion

This master thesis designs a new security solution and introduces end to end security to VoIP systems. The research questions asked in section 1.5 will here be discussed and determined to what extent they have been answered.

*Question 1: How can the Signaling protocol used in VoIP be secured?*

The work done in this master thesis has been on the Session Initiation Protocol as this is the most likely candidate to a future standard. SIP is a application-layer signaling protocol that can be secured with different techniques.

TLS can be used to provide transport level security to signaling from one node to another. This solution will however leave the message in clear text in each node and must be considered to be a weak security solution.

IPSec is implemented at the operating system level securing connections that can be established without involvement of an UA. IPSec assumes that a pre-established trust relationship is introduced between the communicating parties making it most suited for SIP hosts in a VPN scenario. IPSec will as TLS only secure a connection from node to another and security must be provided in a hop-by-hop fashion.

S/MIME adds security to the message itself and can be used to provide end-to-end security to SIP. S/MIME can provide confidentiality or integrity or both. S/MIME is based on the usage of public key infrastructure and certificates containing public keys. Such solution is often considered to be complicated to be use by end users.

The introduction of Identity Based Encryption as a public key cryptosystem solves this problem as shown in chapter 3. IBE introduces a secure solution to VoIP without the need of certificates making the solution much easier to use.

The rest of the questions will only deal with the IBE, S/MIME solution.

*Question 2: Is confidentiality needed and if so, how can this be provided?*

Yes, confidentiality must be added to the SIP signaling to prevent the many different forms of attacks that are described in section 2.4. Confidentiality is provided by encrypting the original message and add the result to a new S/MIME encoded message. The original message is tunneled within an S/MIME body and only parameters used in routing is readable on eavesdropping.

Message encryption is done with an symmetric encryption algorithm of choice. The symmetric key in use is encrypted with an Identity Based Encryption algorithm and transported as a part of the S/MIME body. The SIP address is used as an identity string for user identification.

*Question 3: How can integrity be provided?*

Both SIP messages in plain text and encrypted messages can be exposed to changes by an attacker. It is therefore necessary to provide integrity to all SIP messages. Integrity can be added to a SIP message by tunneling the original message within a S/MIME body and add a detached signature as shown in section 3.3.1.

A message is signed with the identity-based signature scheme and the private key of the sender and should be verified by the receiver by using corresponding public key and identity string. Both confidentiality and integrity should be provided to each message in a session.

It should be noted that the PKG must be highly trusted to compensate for the key escrow property of identity-based cryptography.

*Question 4: How can availability be assured in a VoIP system?*

A new identity based encryption system will add some overhead to the SIP signaling as shown in the prototype in section 4.9. The prototype indicates that the overhead is within what must be seen as acceptable limits.

New network components introduced with the IBE are critical for the solution. The PKG is a central component in this solution with high availability requirements and careful adjustments and optimization is needed to get high availability in the system.

A system that is secured with IBE may also increase the availability as it prevents many of the different attacks that are described in section 2.4.

*Question 5: How can a client be assured that message is secured from one phone to another?*

A user of a VoIP service expects that the communication is secured from one phone to another. This is however not the case in many solutions today.

The introduction of IBE and S/MIME will add end to end security to the VoIP services assuring confidentiality and integrity. The VoIP phone must however be upgraded to support IBE and s/MIME.

*Question 6: How can a recipient be assured that a message really is from the stated sender?*

Messages used in a SIP session must be signed to provide user authentication and no-repudiation of origin. A message is signed with a identity-based signature scheme and the private key of the sender and should be verified by the receiver by using corresponding public key and identity string. This will identify the sender of the message as only the holder of the private key can sign a message in a correct way. It should be noted that the PKG must be highly trusted to compensate for the key escrow property of identity-based cryptography.

This solution require that the User Agent is authenticated in a reliable way before getting a private key. This suggest that a more secure solution than username and password should be used. Smartcards with stored user information may be a good solution to achieve this.

*Question 7: Is it possible to introduce a "zero effort" security solution that can be used by non technical users?*

A security solution in VoIP system must be very easy to use as most people know very little about VoIP and information security.

Identity Based Encryption will make the security solution much easier to use than a solution that is based on PKI and certificates. Information needed by the IBE system is fetch from the local PKG automatically and the recipient SIP address is used as an identity string. This would indicate that this almost is a zero effort solution. Almost as

user initially must add their credentials to the VoIP phone. The credentials should be distributed by the provider when a new user is registered. The only thing a user needs to do after that is to decide if the call should be secured or not.



## 7 Further work

The work done in this master thesis indicates that the combination of identity-based cryptography and S/MIME is a ideal solution for securing VoIP. There are however more work that has to be done to get an working solution in a commercial application.

It must be possible to use phones from several supplier with different software in the same provider network and between different provider networks. This indicates that the new combination of identity-based cryptography and S/MIME has to be described in detail. A new specification has to be developed that states how different identity-based cryptosystem should be integrated with CMS specification [36] and [37]. These specifications may also need to be generalized to support public key cryptosystems that are not based on certificate usage. Specifications in this area are often developed by organizations like "The Internet Engineering Task Force".

The introduction of identity-based cryptography in S/MIME changes the way sender and receiver uses public/private keys. A new specification has to be developed that describes the steps that a sender and receiver has to take. This is discussed in section 3.4.1.

identity-based algorithms suitable for encryption and signing in VoIP should be pointed out and tested. This work should also indicate Elliptic curves and system parameters that may be used in a VoIP system. An reference implementation of a S/MIME version that supports IBE and that is based on these specifications should then be implemented.

The prototype developed in this project shows that a solution based on IBE and S/MIME can work in practice. However, some shortcuts has been made and we will here suggest further activities.

The prototype is based on a simplified version of S/MIME as no specification exists in the area today. A real implementation of S/MIME with IBE support should be integrated into the prototype when available.

The Boneh-Franklin IBE algorithm is used in the prototype. Other IBE algorithms should be investigated, implemented and integrated to be able to find a suitable algorithm for VoIP.

A Private Key Generator (PKG) should be developed that supports the functions defined in section 3.5.1. The UA should then connect to the local PKG to collect system parameters and a private key. A UA should download it's private key and store it locally in a secure way using some kind of encrypted key store.

Confidentiality is added to the prototype by encryption of the original message, no integrity is provided due to time limitations. Integrity should be added to the prototype in the form of a signature in each SIP message. Verification of this signature should also be implemented to get a complete solution. This will also add user authentication and non-repudiation of origin to the solution.

The prototype is not as stable as desired. Work should be done to remove known bugs and to get stable prototype that can be used in further work in the area.





## Bibliography

- [1] Kuhn, D., J. Walsh, T., & Fries, S. JANUARY 2005. Security considerations for voice over ip system. *Recommendations of the National Institute of Standards and Technology*, 1(1), 93.
- [2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. 2002. Rfc3261 - sip: Session initiation protocol.
- [3] Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. 2003. Rfc 3550 - rtp: A transport protocol for real-time applications.
- [4] Varshney, U., Snow, A., McGivern, M., & Howard, C. 2002. Voice over ip. *Commun. ACM*, 45(1), 89–96.
- [5] Leedy, P. D. & Ormrod, J. E. 2005. *Practical Research, planning and design*. Pearson Merrill Prentice Hall, 8th edition.
- [6] Yin, R. K. 2003. *Case Study Research: Design and Methods*. Sage Publications Inc, third edition.
- [7] Arkin, O. November 2002. Why e.t. can't phone home?, security risk factors with ip telephony based networks.
- [8] Sipera. Comprehensive voip security for the enterprise: Not just encryption and authentication. Technical report, Sipera systems, mars 2006.
- [9] Geneiatakis, D., Kambourakis, G., Dagiuklas, T., Lambrinouidakis, C., & Gritzalis, S. 2005. Sip security mechanisms: A state-of-the-art review.
- [10] Steffen, A., Kaufmann, D., & Stricker, A. 2004. Sip security. Security Group.
- [11] Sparks, R. 2007. Sip: basics and beyond. *Queue*, 5(2), 22–33.
- [12] Berners-Lee, T., Fielding, R., Irvine, U., & Masinter, L. August 1998. Rfc 2396 - uniform resource identifiers (uri): Generic syntax. Internet.
- [13] Handley, M., Jacobson, V., & Perkins, C. July 2006. Rfc 4566 - sdp: Session description protocol. Internet.
- [14] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., Willis, D., Rosenberg, J., Summers, K., & Schulzrinne, H. June 2001. Sip telephony call flow examples. Internet.
- [15] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., & Summers, K. December 2003. Rfc 3665 - session initiation protocol (sip) basic call flow examples. Internet.
- [16] Qiu, Q. Master's project report, study of digest authentication for session initiation protocol (sip). Master's thesis, SITE, University of Ottawa, 2003.

- [17] Dotson, S., Hoggan, S., & Channabasappa, S. February 2008. Proxy mutual authentication in sip, draft-dotson-sip-mutual-auth-01. Internet.
- [18] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., & Stewart, L. June 1999. Http authentication: Basic and digest access authentication. Internet.
- [19] Vesterinen, P. December 2006. User authentication in sip. Helsinki University of Technology. TKK T-110.5290 Seminar on Network Security.
- [20] Kent, S. & Atkinson, R. 1998. Security architecture for the internet protocol.
- [21] Ramsdell, B. June 1999. S/mime version 3 message specification, request for comments: 2633. Network Working Group.
- [22] Stallings, W. 2003. *Cryptography and Network Security, Principles and practices*. Prentice Hall, third edition.
- [23] Shamir, A. 1984. Identity-based cryptosystems and signature schemes. *Lecture Notes in Computer Science*, 196, 47–53.
- [24] Boneh, D. & Franklin, M. 2003. Identity-based encryption from the weil pairing. *SIAM J. of Computing*, 32, No 3, 586–615.
- [25] Baek, J., Newmarch, J., Safavi-Naini, R., & Susilo, W. 2004. A survey of identity-based cryptography.
- [26] Owens, L., Duffy, A., & Dowling, T. 2004. An identity based encryption system. Internet.
- [27] Martin, L. 2008. *Introduction to Identity-Based Encryption*. Artech house.
- [28] Sakai, Ohgishi, & Kasahara. 2000. Cryptosystems based on pairings.
- [29] Barbosa, M. B. June 2005. Identity based cryptography from bilinear pairings.
- [30] Anderson, R. J. 2001. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, first edition.
- [31] VOIPSA. Voip security and privacy threat taxonomy. Technical report, VOIPSA, October 2005.
- [32] Niccolini, S. & Chen, E. Voip security threats relevant to speermint draft-niccolini-speermint-voipthreats-02. Technical report, The IETF Trust, SPEERMINT Working Group, August 2007.
- [33] Thermos, P & Takanen, A. 2008. *SECURING VoIP NETWORKS Threats, Vulnerabilities, and Countermeasures*. Addison Wesley.
- [34] Baumann, R., Cavin, S., & Schmid, S. Voice over ip - security and spit, swiss army, fu br 41, kryptdet report. Technical report, University of Berne, September 2006.
- [35] Zandi, M., Martin, M. V., & Hung, P. C. Mars 2007. Overview of security issues of voip.

- [36] ITU-T. May 2003. Itu-t recommendation g.114 one-way transmission time.
- [37] Juniper Networks, I. Enterprise voip security, best practices. Technical report, Juniper Networks, Inc., 2006.
- [38] CFCA. 2006. Cfca where communications professionals go to know, fraud control, revenue assurance, risk management. *CFCA webpage*, none, 1.
- [39] Consortium, T. I. E. Fraud analysis in ip and next-generation networks. web.
- [40] Freed, N. & Borenstein, N. November 1996. Multipurpose internet mail extensions (mime) part one: Format of internet message bodies. Internet.
- [41] Housley, R. & Security, V. July 2004. Rfc 3852: Cryptographic message syntax (cms). Internet.
- [42] Housley, R. August 2002. Rfc3370: Cryptographic message syntax (cms) algorithms. Internet.
- [43] Veltri, L. Mjsip-mini-tutorial. version 0.1, April 2005.
- [44] Duffy, A. & Dowling, T. 2004. An object oriented approach to an identity based encryption cryptosystem. Internet.
- [45] Lamping, U., Sharpe, R., & Warnicke, E. *Wireshark User's Guide, 26165 for Wireshark 1.0.0*, 2004-2008.



## A Configuring the prototype

The soft phone can be run in *normal* or *secure* mode. The parameter *ibe* has to be set to *yes* to use *secure mode*.

Username and password and proxy server address is also added. The following shows the user *Alice* configuration file *alice.cfg* that can be found in the *config* catalog:

```
#           MjSip UA configuration file
# -----
#

#via_addr=127.0.0.7
host_port=5070
transport_protocols=udp tcp
#outbound_proxy=127.0.0.2:5060
#outbound_proxy=169.254.25.129:5060
outbound_proxy=atlanta.com

#from_url="Alice" <sip:alice@127.0.0.2>
#from_url="Alice" <sip:alice@169.254.25.129>
from_url="Alice" <sip:alice@atlanta.com>
username=alice
#realm=127.0.0.2
#realm=169.254.25.129
realm=atlanta.com
passwd=abc123
ibe=yes

debug_level=8
log_path=log
max_logsize=2000

contacts_file=config/contacts.lst

do_register=yes
#do_unregister=yes
#do_unregister_all=yes

keepalive_time=8000

#call_to=sip:127.0.0.9:5090
#accept_time=0
#hangup_time=20
```

```
#redirect_to=sip:127.0.0.9:5090
#no_offer=yes
#transfer_to=sip:127.0.0.9:5090
#transfer_time=10
#re_invite_time=5

#recv_only=yes
#send_only=yes
#send_tone=yes
#send_file=yes

audio=yes
audio_port=3001
audio_avp=0
audio_codec=PCMU
audio_sample_rate=8000
audio_sample_size=1
audio_frame_size=500

#video=yes
#video_port=3002
#video_avp=101

use_jmf=yes
#use_rat=yes
#use_vic=yes
#bin_rat="c:\program files\mbone\rat"
#bin_vic="c:\program files\mbone\vic"
bin_rat=rat
bin_vic=vic
```

An working configuration files for Alice and Bob that is also used in the test runs are found in the *config* catalog. More information on other MjSip User Agent parameters can be found in [43].

The proxy server does also need to be configured before starting up. The following shows the configuration file *server.cfg* that also can be found in the catalog *config*. This configuration is used in the prototype test runs.

```
#      MjSip Server configuration file
# -----
#
# ***** sip *****
#via_addr=127.0.0.2
#via_addr=192.168.1.33
```

```
#via_addr=169.254.25.129
host_port=5060
transport_protocols=udp tcp
#outbound_proxy=127.0.0.3:5069

# ***** server *****
#on_route=yes
#loose_route=yes
call_log=yes
#domain_names=wonderland.net neverland.net
domain_names=atlanta.com
is_registrar=yes
register_new_users=yes
is_open_proxy=yes
location_service=local
location_db=config/location.db

do_authentication=no
do_proxy_authentication=no
#authentication_realm=wonderland.net
#authentication_service=ldap
#authentication_scheme=local.authentication.AkaServerImpl
authentication_db=config/auth.db

# ***** static routing *****
#phone_routing_rules={prefix=06,nextthop=127.0.0.1:7001}
# {prefix=0521,nextthop=127.0.0.2:7002} {prefix=*,nextthop=127.0.0.3:7003}

# ***** debugging *****
debug_level=6
log_path=log
max_logsize=2000
log_rotations=7
log_rotation_time=1 DAYS
```

More information on the MjSip server configuration can be found in [43].





## B Building the prototype

A Java SE development kit 1.6 (or later) has to be installed on a development machine to be able to build the prototype. The Java compiler can be freely downloaded from the Sun microsystems web site <http://java.sun.com/javase/>.

The project can be built with your favorite IDE or with the *make* command. The IDE eclipse is used in this project for development. The *make* command builds a release that consists of the three jar files *sip.jar*, *server.jar*, and *ua.jar*. A build with the *make* command is necessary if changes is made to the code and when you want to build a new release.

You can use the following commands:

*make all* - build all jar files

*make ua* - build User Agent jar file

*make sip* - build sip jar file

*make server* - build server jar file

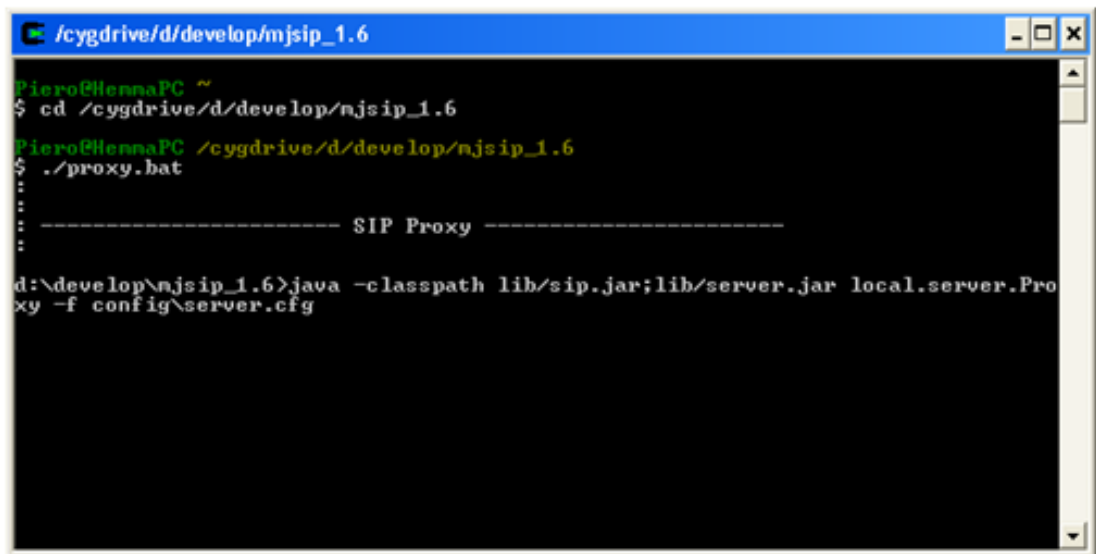
The make file may have to be installed on your development PC.



## C Running the prototype

The MjSip server has to be started first as User Agents has to authenticate and register them self to this server. The MjSip server can be started by running a start script or by executing an Java class with a main method.

You can start the MjSip server from a Cygwin shell by running the start script *proxy.bat* that is located in the main catalog:

A screenshot of a Cygwin terminal window titled "/cygdrive/d/develop/mjsip\_1.6". The terminal shows the following commands and output:

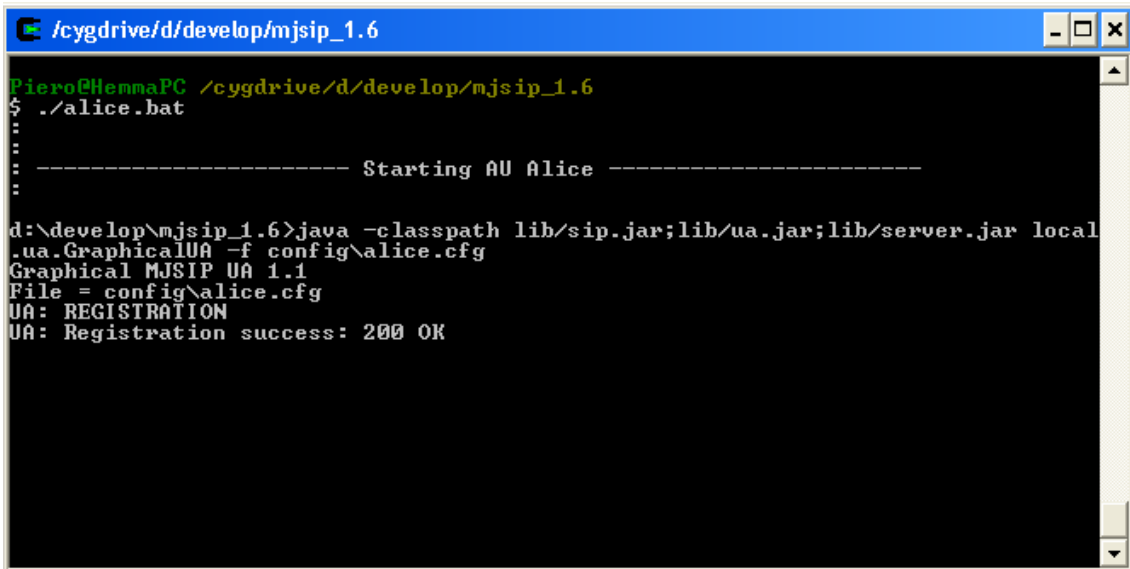
```
Piero@HennaPC ~
$ cd /cygdrive/d/develop/mjsip_1.6
Piero@HennaPC /cygdrive/d/develop/mjsip_1.6
$ ./proxy.bat
:
: ----- SIP Proxy -----
:
d:\develop\njsip_1.6>java -classpath lib/sip.jar;lib/server.jar local.server.Proxy -f config\server.cfg
```

Figure 19: Starting MjSip Server from shell

You can also start the mjSip server from your favorite IDE or from command line using the Java command shown in figure19.

The MjSip UA's are after that started for the different participants. An UA can be started by running a start script or by executing a Java class with a main method.

You can start the MjSip UA Alice from a Cygwin shell by running the start script *alice.bat* as shown in figure 20 and you can start the MjSip UA Bob from a Cygwin shell by running the start script *bob.bat* as shown in figure 21:

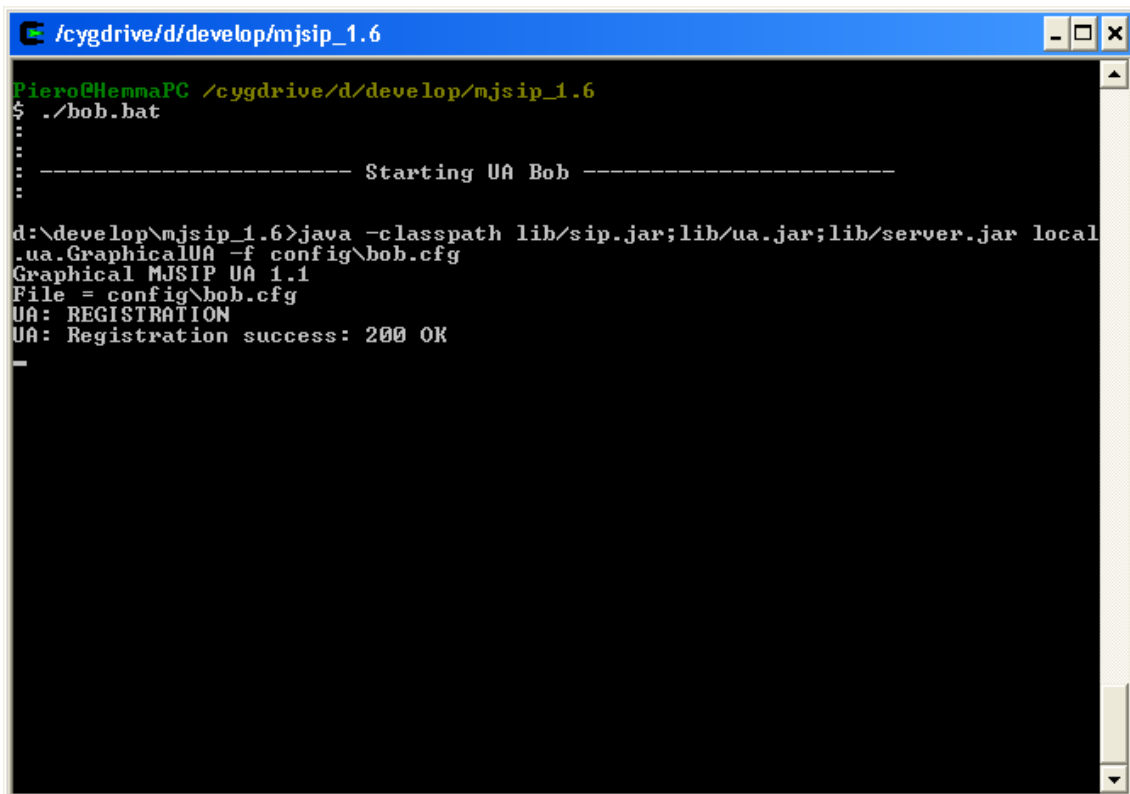


```

C:/cygdrive/d/develop/mjsip_1.6
Piero@HemmaPC /cygdrive/d/develop/mjsip_1.6
$ ./alice.bat
:
: ----- Starting AU Alice -----
:
d:\develop\mjsip_1.6>java -classpath lib/sip.jar;lib/ua.jar;lib/server.jar local
.ua.GraphicalUA -f config\alice.cfg
Graphical MJSIP UA 1.1
File = config\alice.cfg
UA: REGISTRATION
UA: Registration success: 200 OK

```

Figure 20: Starting MjSip UA Alice



```

C:/cygdrive/d/develop/mjsip_1.6
Piero@HemmaPC /cygdrive/d/develop/mjsip_1.6
$ ./bob.bat
:
: ----- Starting UA Bob -----
:
d:\develop\mjsip_1.6>java -classpath lib/sip.jar;lib/ua.jar;lib/server.jar local
.ua.GraphicalUA -f config\bob.cfg
Graphical MJSIP UA 1.1
File = config\bob.cfg
UA: REGISTRATION
UA: Registration success: 200 OK
-

```

Figure 21: Starting MjSip UA Bob

A server and User Agents are then started and a small Java swing client pops up for each UA as shown in figure22 and 23.

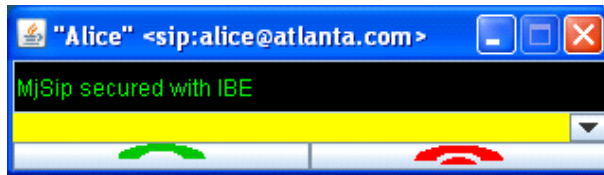


Figure 22: Alice User Agent in idle state



Figure 23: Bob User Agent in idle state

Alice starts a call by entering Bob SIP address and presses the green call button, figure 24. The SIP address can also be stored in advance in the file *contacts.lst* in catalog *config*.

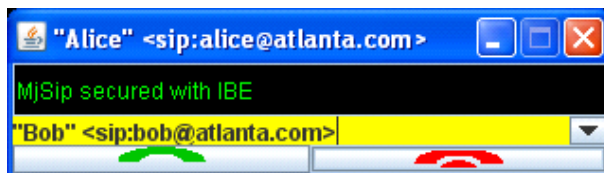


Figure 24: Alice enters Bobs SIP address and makes a call

Alice UA changes state to *ringing*, figure 25 and Alice is waiting for Bob to answer. Bobs UA changes state to *incoming call* and starts to ring, figure 26.

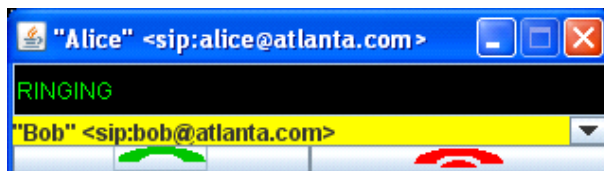


Figure 25: Alice UA changes state

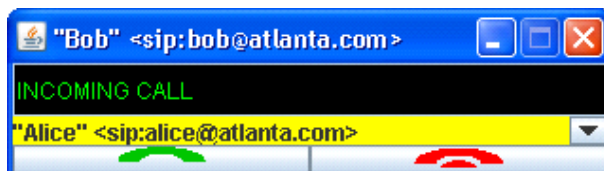


Figure 26: Bobs phone is ringing

Bob answers the call by pressing the green button and a call is established. Both UAs

changes state to *on call* as shown in figure 27 and 28.



Figure 27: Alice UA is on call



Figure 28: Bobs UA is on call

Bob ends the call by pressing the red button and the connection is teared down, figure 29 and 30. Alice UA receives the hangup shows this to the user and goes to idle state.



Figure 29: Bob ends the call by hanging up



Figure 30: Alice is informed