Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Computer Engineering Stream

Securing Confidentiality and Integrity of SIP Based VoIP System in Reduced Call Setup Time

By: Solomon Negussie

Advisor: Dr. Yalemzewd Negash

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Engineering.

June, 2014

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Computer Engineering Stream

Securing Confidentiality and Integrity of SIP Based VoIP System in Reduced Call Setup Time

By: Solomon Negussie

Advisor: Dr. Yalemzewd Negash

Signature of the Board of Examiners for Approval

Chairperson,	Signature	Date
Advisor	Signature	Date
Internal Examiner	Signature	Date
External Advisor	Signature	Date

Declaration

I, the undersigned, declare that this thesis is my original work, has not been presented in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Solomon Negussie

Signature: _____

Place: Addis Ababa Institute of Technology (AAiT), Addis Ababa University, Addis Ababa, Ethiopia

Date: _____

This thesis has been submitted for examination with my approval as a university advisor.

Advisor: Dr. Yalemzewd Negash

Signature: _____

Date: _____

Dedication

To my loving parent.

Abstract

Voice over IP is one of the quickest developing Internet services promised to have growing demand. However this demand is challenged by security pitfalls of the current internet security problems plus none flawlessness of the VoIP protocols themselves. Specially as any telecommunication service requires privacy, confidentiality, integrity and availability protection VoIP needs to assure its customers those information security principle concerns with tolerable quality of service, the concerns not raised frequently in PSTN(Public Switched Telephone Network) and other traditional telephone networks. Even though there are many protocols and security principles made available to be used by developers and service providers, however current applications are more on the quality and the extent of service they provide rather than the concern in security breach. However there are some VoIP application developers and service providers that provides the capability of encrypting the communication, which is not sufficient since the signaling is not protected due to fear of additional overhead.

By this work we proposed new way of setting secure VoIP (Voice over Internet Protocol) session(secure in a context of confidentiality and integrity protection only) in SIP (Session Initiation Protocol) based VoIP system using the current recommended and best practices of securing VoIP so that the newly proposed system will have improved call setup time. We also designed and implemented the proposed secure call setup mechanism as a proof for the practicability of our recommendation and as a test bed to assess the effect of the proposed mechanism in modifying the time required to setup a secure call, i.e. confidentiality and integrity guaranteed session. We have compared our proposed mechanism of establishing a secure conversation and signaling session with a common secure session establishment technique and we have seen that our proposed system requires only 8.0 second to establish secure session, while the other common practices takes about 10.5 second. Hence the proposed system reduces the call setup time by about 2.5 seconds.

Keywords: *VoIP, SIP, TLS, SRTP, ZRTP, MZRTP, Call Setup Time, Confidentiality, Integrity.*

Acknowledgment

Before everything I would like to thank God for the wisdom and perseverance that he bestowed up on me during this thesis work and, indeed throughout my life.

Next, I would like to express my sincere gratitude to my advisor, Dr. Yalemzewd Negash, for his advice, attention and encouragement during my research studies.

My family have been always generous with their encouragement to me. My deepest appreciation goes to my parents, Negussie T. and Bayush D., all together with my sisters and brothers. I am forever indebted to them for their endless love and support.

Finally, I would like to express my sincere gratitude to everybody who helped me in any aspect for successful completion of this thesis work.

Table of Contents

List of Figures

List of Tables

Abbreviations

ACD= Automatic Call Distribution

AES= Advanced Encryption Standard

AVR= Automatic Voice Reply

CA= Certificate Authority

CDR= Call Data Record

CIA= Confidentiality, Integrity and Availability

CSR= Certificate Signing Request

DH= Diffie-Hellman

IM= Instant Messaging

IVR= Interactive Voice Response

JAIN=Java API for Integrated Network

MG= Media Gateway

MGCP= Media Gateway Control Protocol

MIKEY=Multimedia Internet KEYing

MZRTP= Modified ZRTP

MiMT= Man in the Middle Attack

PBX= Private Branch Exchange

POTS= Plain Old Telephone System

PSTN= Public Switched Telephone Network

QoS= Quality of Service

RFC=Reference for Comment

RTCP= Real Time Control Protocol

RTP=Real Time Protocol

S/MIME=Secure/Multipurpose internet Mail Exchange

SAS= Short Authentication String

SDES=Session Description Security Protocol

SDP= Session Description Protocol

SG= Signaling Gateway

SIP= Session Initiation Protocol

SIPS= Session Initiation Protocol(Secure Version)

SMS= Short Messaging Service

SRTP= Secure Real Time Protocol

SSL= Secure Socket Layer

TLS= Transport Layer Security

UA= User Agent

UAC=User Agent Client

UAS=User Agent Server

URI= Universal Resource Locator

VoIP= Voice over Internet Protocol

VoIPSA=VoIP Security Alliance

ZRTP= Zimmermann Real Time Protocol

Chapter One

1. Introduction

Voice over IP (VoIP, or Voice over Internet Protocol) commonly refers to the communication protocols, technologies, methodologies, and transmission techniques involved in the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet[33]. The steps involved in originating a VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, encoding, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side, similar steps (usually in the reverse order) such as reception of the IP packets, decoding of the packets and digital-to-analog conversion reproduce the original voice stream. VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codec's which encode speech allowing transmission over an IP network as digital audio via an audio stream. Because of the bandwidth efficiency and low costs that VoIP technology can provide, businesses are migrating from traditional copper-wire telephone systems to VoIP systems to reduce their monthly phone costs. In 2008, 80% of all new PBX lines installed internationally were VoIP [33].

Generally we can categorize the advantages of VoIP into three major groups [16]. The first is toll by pass. It is saving of money expenses resulted due to a charge for long distance phone call over PSTN by placing the long distance call over pre- established data network or internet.

The second biggest advantage of VoIP is network consolidation. By VoIP network consolidation we mean single internet network can be used for transmission of voice, video and other multimedia services, hence reducing setup and maintenance costs even service charge costs that might be high if each service were made separate like assigning PSTN network for voice and internet or IP for data communication. The third is service convergence. Many multimedia services such as instant messaging (IM), automated voice replay (AVR), SMS, Voice chat, video chat, video conference, etc could be achieved with one complete VoIP software or application.

Like many technology as well as the factor that it is young or new technology, VoIP also has some shortcomings or challenges. Since VoIP is a transmission of real time voice packet over internet protocol it is also a must for the technology to inherit the security threats of the current internet network. The difficulty in employing secure VoIP service is due to the multidirectional threats of VoIP and absence of fully defined and reliable standard protocol that can be used by every VoIP application developer and service vendors [22]. This attack vulnerability encompasses vulnerabilities due to the immaturity of VoIP protocols themselves plus the long lasting security issues of the internet world, the operating platform and the application itself. Moreover even though the protocols are flawless the errors during the implementation of the protocol by the programmer are also another big source of vulnerability.

However, due to the sensitivity of the content of the communication and need of privacy it is a must requirement for VoIP to be secure at least in its confidentiality and integrity from the customer context and availability of the service from service provider/vendor context in order to take over the current PSTN network market coverage. But currently many VoIP vendors worry for the quality of the service rather than the possible security breach due to the perceived overhead and quality reduction to be introduced by the security features alongside the legal issues of some countries prohibiting secure communication more specifically encrypted conversation. This thesis however believes that security is not a feature of VoIP or Telephony in general instead it is a prerequisite for telephony system to be reliable and trustworthily utilized by customers while techniques to reduce the overhead of the security features to bearable level be further analyzed. Hence, in this thesis we design and implement a new technique devised by using current best and recommended internet security mechanisms for establishing secure VoIP communication session, secure in context of confidentiality and integrity of the service, in reduced time thus increased performance and quality of the service.

1.1. Definition of Terms

- Confidentiality: The property that information is not disclosed to system entities (users, processes, devices) unless they have been authorized to access the information (Committee on National Security Systems, 2010).

- Integrity: The property whereby an entity has not been modified in an unauthorized manner (Committee on National Security Systems, 2010).
- Availability: The property of being accessible and useable upon demand by an authorized entity (Committee on National Security Systems, 2010).

1.2. Motivation

Currently in a telecommunication industry there is a huge demand of the following key changes:

- Demand for Multimedia communication
- Demand for integration of Voice and Data networks
- Cost Reduction in long distance telephone calls

And not a demand only but there is a good achievement in satisfying the demands too. As statistics show only by the year 2008 out of the total telecommunication new network installment the 80% was new installation of VoIP infrastructures which means the current PSTN network is sooner than latter to be replaced by VoIP network due to the above demands. However, due to immaturity of the VoIP technology and standardizing bodies that forces the service vendors to follow a unique and uniform standard it is hard for customers to shift from PSTN and relay on VoIP, since there will be plenty of ambiguous protocols and practices who claim to be secure.. Currently service providers and developers focuses on the different features they can employ on their application rather than focusing on the security of their application. Security is left as scarifies pay for quality and technology supported by the vendors for market competition.

Thus the motivation to work on this thesis topic came from the belief that security must precede the quality and technology consolidation worries or concerns since telecommunication service is sensitive service that at least requires guaranty of confidentiality, integrity and availability, CIA in short. Out of these three information security principles we are targeting only the two, confidentiality and integrity, as the scope of this thesis.

1.3. Statement of Problem

As there are too many vulnerability holes for the success of attacks on VoIP network, vulnerabilities on the confidentiality, integrity and availability of the service are intolerable since they mean a lot to both the service vendor and service customer. Thus developing VoIP application without guarantying these three major principles of communication system is letting the privacy, confidentiality, integrity of the customer's communication and infrastructure of the vendor to fraudsters and an intended service takers or attackers. However, implementing strong security feature on the application shouldn't degrade the QoS of the application too. Thus the statement of problem of this thesis is implementing dependable security measure on VoIP service that guaranties two of the three major information security principles these are confidentiality and integrity while reducing overheads and time taken by current best and recommended VoIP security mechanisms to establish secure session. This thesis focuses only on SIP based VoIP system, because SIP is simple and nowadays popular and preferable protocol over the other session initiation protocols due to its less complexity, less hardware and technology requirement. Thus hereafter unless specifically noted out wherever VoIP is mentioned in this document we are referring to SIP based VoIP system which means the signaling protocol is SIP.

1.4. Objectives

General Objectives

In this thesis work there are two main objectives. These are:

- One, designing and implementing VoIP application that can guaranty *complete* protection of the confidentiality and integrity of VoIP service by implementing secure SIP using TLS and secure media session using SRTP protocol where the key exchange for SRTP is a newly proposed protocol which we called it Modified ZRTP ,or in short MZRTP, so that one can establish secure communication session in a reduced call setup time.
- Second we will analyze the effect of implementing MZRTP on improving call setup time of secure session, that is QoS of the proposed system in improving

performance, by measuring the time taken to establish complete secure session and compare it against current widely implemented as well as current more favorable and relatively strong security features.

Specific Objectives

The specific objectives of this thesis work are:

- Studying and documenting of the current attacks on VoIP service and categorizing them as confidentiality and integrity threats.
- Studying and documenting current VoIP security solutions specifically security features measures targeting confidentiality and integrity of VoIP service.
- Implementing complete and working softphone and VoIP system so that we can implement our proposed security technique for testing purpose.
- Designing and implementing the proposed security feature, MZRTP, on the implemented softphone.
- Analyzing the effect of MZRTP in improving the quality of service while relying on the analysis of other scholars on the security strength of protocols we used in our proposed technique *like TLS, SRTP and ZRTP.*

1.5. Thesis Contribution

This thesis work has the following contributions:

- We have discussed and documented the potential and possible attacks on the confidentiality and integrity of VoIP which can be considered as milestone for further analysis of VoIP vulnerabilities.
- We have proposed, designed, implemented and tested a new way of securing VoIP system or more specifically the call setup part of VoIP which is composed of two phases which are treated independently and secured independently by many scholars and service venders, signaling and media phases, so that those phases be linked to reduce or eliminate the cost of time delay scarified and complexity introduced to achieve security.

- The last but not the least contribution of this thesis work is that we have developed a softphone and working test network that can be taken as starting point for future work, specially for someone who wants to build a more general multimedia application that includes audio and video call, voice chat, video conferencing, AVR (Automated Voice Reply), and if our country's telecommunication service start to work in favor of data communications too alongside its current telephone network to cope the ever increasing demand of the service in the country.

1.6. Document Structures

The thesis document is organized based on the following sequence:

- The first chapter presents the motivation, objectives and statements of the problem.
- In second chapter we discussed in detail the related works done by prior researchers, organizations and VoIP service venders.
- Chapter three introduces in detail what a SIP based VoIP system is and how it works as well as some of it's the current well known or anticipated vulnerabilities and attacks.
- In chapter four we explained our proposed solution and its working principles.
- Chapter five is the section of our document where we described our implementation of the proposed security and tested the implemented system.
- Chapter six is the conclusion and recommendation section. The document also has appendixes.

Chapter Two

2. Literature Review

2.1. Background History

One could simply say telecommunication has evolved through three technological evolutions. The first was a period of **generic telephone network** where a human aid was required to switch and set up the telephone call. The second period was the period where automated telephone switching was made possible and this new telephone system is known as **Plain old Telephone System (POTS)**. The third evolution which was a modification of the second technology employed a digitalization of telephone system which is considered as a big difference to the POTS which was analog system. This third period is known as **Public switched Telephone Network or PSTN**. PSTN became a reason for popping out for more advanced services like fax.

However on the other hand Cellular networks with their completely different architecture, wireless access and mobility issues are treated as a similar (meaning digital system) but independent technology [21].

The most recent evolution of telephony and which we would like to call it the fourth stage in telephone system is VoIP due to its huge difference from PSTN. To mention some of the differences between the two technology; PSTN is circuit switched network while VoIP is packet switched system. In VoIP traffic is independent of geographical distance between the caller and receiver since the packets can follow dynamic route to reach between the two peers. This feature of VoIP technology is a key and primary reason for developing VoIP since the callers pay equal payment no matter how far the called peer is from the caller but this is what PSTN lack. One more difference of the two technology is that VoIP supports advanced features like video call, audio call, Instant messaging, AVR, email, etc allowing merging of a service into a single application and device whereas PSTN is perfected for telecommunication services only.

In the above paragraph we spoke in pro of VoIP which doesn't necessarily mean VoIP should replace PSTN. PSTN has far more advantage over VoIP when question of reliability is asked. Since the PSTN was never intended to support any other type of

traffic, it handles telephony very well. Key virtues of PSTN include pristine quality, nearly 100% uptime, highly private and secure connections, and the ability to scale and support large volumes of traffic. VoIP is designed to work on a data network and, by its nature, cannot match the PSTN in these areas. VoIP technology has come a long way, and while it's good enough for businesses to rely on, there will be some compromises that come with the cost savings, i.e. security. This shows that, just because something is plain and old doesn't necessarily mean it's time to rip and replace it. Table 1 shows the feature by feature comparison of PSTN and current VoIP technology for customers to choose the technology [21], [24].

Table 1. PSTN vs. VoIP: Feature-by-feature comparison

Property	PSTN service	VoIP
switching	circuit switched – bandwidth reservation, resources are used even if there is no information to be transmitted	packet switched – no bandwidth reservation; resources are not used, if there is no information to be transmitted
services	traditional services, like phone calls, faxes, voice mailboxes, caller identification, etc.	almost all traditional services and many more – video, message, data transmission, etc.
quality	Quality of Service is guaranteed, bandwidth reservation – 64 kbps; but QoS limited to standard value and may not be enhanced	no band reservation, quality may be affected if network traffic is too high; but with a sufficient bandwidth available, quality can be better than in PSTN
mobility	originally no mobility option available, but was offered later in mobile networks	calls may be answered and originated from any place with a sufficiently fast Internet connection; user may generate calls from any place of the network, but also receive them
infrastructure	separate telephone infrastructure	shared infrastructure with data

	necessary	network
cost	relatively high, because of additional infrastructure and management	relatively low, as existing data networks may be used for transmission
power supply	independent power supply (48V supplied by the telephone line), telephones work even during power black out	no independent power supply, dependent on household electricity; power infrastructure needed for VoIP switches and every single desktop device
standards	well defined, common standards	no widely adopted standard, many RFCs and competitive standards covering some problems
architecture	centralized, highly complex central architecture	simple core network required, features implemented in end points, but the overall architecture complex
compression	no compression	compression using limitations of eyes and ears to limit bandwidth; audio – silence suppression, video – motion detection
access	Limited	open architecture – almost no limitations
emergency	in case of an emergency call, the caller may be localized, as each client has his/her own subscriber line	no built in emergency localization mechanism
numbering	specific to geographical location, number attached to the closest exchange and identified by the beginning of the number	uses email-similar address structure, geographically independent, identified by server's domain name or IP
scalability	upgrades require purchasing	upgrades usually require more

	more hardware and dedicated lines, which can be very complex and costly.	bandwidth and simple software updates.
call waiting	available at extra cost	most VoIP options offer free call waiting, such as Google Voice and Skype.
call forwarding	available at extra cost	some VoIP options provide free call forwarding (Google Voice), while others offer it for an extra fee or through a subscription (Skype).
call transferring	available at extra cost	some VoIP options provide free call transferring (Google Voice), while others do not support call transferring at all (Skype).

In the following sections we will review some prior scholarly works done by researches and technology vendors of VoIP in securing the service so that the technology can be dependable and yield its expected advantages over PSTN securely and QoS reliably.

2.2. Related Works

As VoIP is a young service perceived to have a bright future many researchers and developers of the service has performed and being performing so many researches and findings to increase the dependability of the service. Of all the concern in the acceptability of VoIP as telecommunication technology, security is a big question and it is the question that most researches or developers try to address. In this section we will see some of the works that are at a closer proximity to this thesis work. The papers and journals we categorized under related works are really not a work that are similar or other version of our work instead they are papers that we used as facts to the reliability and acceptability proof of our works since we couldn't find any prior work that have similarity with our intent.

IP telephony- related protocols were not designed with security as their first priority or as a prime design goal. Some of those protocols added security features when newer protocol versions were introduced and when IETF threatened them by not willing to accept a protocol without security given appropriate consideration [25]. Though this is the fact however currently there are many options of securing VoIP protocols with other pre existing internet security protocols designed for HTTP and SMTP though each of the alternatives has their own drawbacks and shortcomings.

Obviously encryption is a prime countermeasure to be made on services to guaranty confidentiality where as cryptographic signature and authentication mandates protecting integrity threats. Similarly SIP employs, well not directly employ any specific security protocol for SIP but recommends some well known techniques designed previously for other internet security purposes, encryption mechanisms to provide confidentiality and to prevent malicious users from modifying the messages. However, the complete end-to-end encryption of SIP messages is not possible as SIP intermediaries need access to some information in SIP headers. For this reason, SIP supports both end-to-end encryption and hop-to-hop encryption techniques [5], [25].

The end-to-end encryption is supported using S/MIME mechanisms for all the information that is not required to be accessed by the intermediaries. The hop-by-hop

encryption is supported using IPSec or TLS and is used for preserving the confidentiality of the information that needs not to be seen by the intermediaries. The encryption algorithms commonly used with SIP are the Data Encryption Standard (DES) and Advanced Encryption Standard (AES). Note that although encryption provides confidentiality of SIP messages, it can be detrimental to QoS.

SIP employs a cryptographic authentication mechanism, which is based on the HTTP Digest authentication defined in RFC2617, for providing the message integrity and verifying the authenticity of the senders of SIP messages. It is important to note that the authentication allows a UA to verify the identity of another UA but does not provide message integrity and hence the need for cryptographic authentication which combines the encryption and authentication techniques in order to provide authentication, confidentiality, message integrity, and protection against replay attacks. SIP authentication mechanism is based on the HTTP Digest authentication. SIP also supports a scheme called SIPS URI which allows SIP intermediaries as indicator of the need to forward SIP messages using TLS security throughout its path, but it doesn't guaranty that the SIP message is transported on secure channel like TLS throughout its path[15],[25].

As noted above to keep the signaling message secure, that is the SIP content, there is a choice of using S/MIME, TLS and IPSec whether end-to-end or hop-to-hop security is desired. However each of them compromises certain things. For example it is impossible to give complete protection to SIP content using S/MIME encryption and signature capability since intermediaries need to understand the content of the encrypted message to route the request to the desired destination. Which simply means S/MIME doesn't protect the whole message though it was a good and reliable technique for end to end protection, since it leaves some headers unencrypted then leading to traffic analysis attack. And the other choice is using either TLS or IPSec, which are believed to have a serious QoS degradation [3].

These protocols, meaning S/MIME, TLS or IPSec, are all signaling security protocols that cannot, or should not, be used in the actual communication security or encryption.

The media needs independent security phase and SRTP is the de facto standard for the media security. SRTP, when implemented correctly and fully is a reliable security measure for VoIP, however the problem for SRTP is that it doesn't have a means of generating shared encryption or session key, instead it depends on other protocols that can help to exchange keys securely [SRTP RFC 3711]. So the major security concern for media security is the key establishment part. For this there are three major proposed techniques, these are SDES, MIKEY and ZRTP of which ZRTP is believed to be relatively strong and robust though some impractical or inconsiderable attacks are noted by some scholars on ZRTP [7].

In this thesis work we will be using TLS to secure the SIP and SRTP to secure the media, which is a common best practice for completely securing VoIP, however one problem with this technique is performance degradation, which we solved considerably with our new proposition of key establishment or exchange technique which also depend on one well known and highly acknowledge protocol called ZRTP. By this proposition we can obtain the advantage of TLS, SRTP and ZRTP while reducing the call setup time, which is one challenge as QoS parameter.

Chapter Three

3. SIP Based VoIP System

3.1. VoIP Overview

VoIP is a set of technologies that enable voice calls to be carried over the Internet (or other networks designed for data), rather than the traditional telephone landline system—the Public Switched Telephone Network, or PSTN.

The term VoIP was coined by the VoIP Forum which was set up in May 1996 as an industry group concerned with promoting and developing product interoperability and a high quality of service for Internet telephony products. Initially, one of the main drivers in developing VoIP was the potential to cut the cost of telephone calls. Traditional voice calls, running over the PSTN, are made using circuit switching, where a dedicated circuit or channel is set up between two points before the users talk to one another—just like old-fashioned operators, plugging in the wires to connect two callers. The advantage of this is that once the circuit is set up, the call quality is very good, because it is running over a dedicated line. But this type of switching is expensive because the network needs a great deal of (mostly under-used) capacity.

The development of VoIP represents a major change in telecommunications. VoIP uses IP protocols, originally designed for the Internet, to break voice calls up into digital 'packets'. In order for a call to take place the separate packets travel over an IP network and are reassembled at the far end. The breakthrough was in being able to transmit voice calls, which are much more sensitive to any time delays or problems on the network, in the same way as data.

Whereas calls over the PSTN are metered, so the user pays for the amount of time taken by their call, Internet usage is not metered. The user pays a set fee for their Internet service and their VoIP service and can then use the Internet to get free phone calls to other users on the same VoIP service, or pay a small fee to call users on other VoIP services or on the PSTN.

Packetized voice also enables much more efficient use of the network because bandwidth is only used when something is actually being transmitted. Also, the network can handle connections from many applications and many users at the same time, unlike the dedicated circuit-switch approach. This greater efficiency is one of the main reasons that all major carriers, such as BT with its 21CN (21st Century Network) project, are changing their own networks so that they are IP-enabled.

Voice over IP (VoIP) uses the Internet Protocol (IP) to transmit voice as packets over an IP network. So VOIP can be achieved on any data network that uses IP, like Internet, Intranets and Local Area Networks (LAN). Here the voice signal is digitized, compressed and converted to IP packets and then transmitted over the IP network. Signaling protocols are used to set up and tear down calls, carry information required to locate users and negotiate capabilities. One of the main motivations for Internet telephony is the very low cost required calling as compared to the current traditional telephone network. Some other motivations are:

- Demand for multimedia communication
- Demand for integration of voice and data networks

A commercial VoIP and complete VoIP system should have the following components: end-user equipment, network components, call processors, gateways and protocols. Figure 1 shows simplified version of a working VoIP infrastructures.

Figure 1. Simplified VoIP infrastructure

End-user equipment is used to access the VoIP system to communicate with another end point. Connection to the network may be physically cabled or may be wireless. The end-user equipment may be a phone that sits on a desk or a softphone that is installed on a PC. Functions include voice and possibly video communication, and may contain instant messaging, monitoring and surveillance capabilities.

Network components include cabling, routers, switches and firewalls. Usually the existing IP network is where a new VoIP system is installed.

Call processor functions can include phone number to IP translation, call setup, call monitoring, user authorization, signal coordination, and may help control bandwidth. Call processors are usually software that runs on a popular OS.

Gateways can be categorized into three functional types: Signaling Gateways (SG), Media Gateways (MG) and Media Controllers. In general, they handle call origination and detection and analog to digital conversion. Signaling gateways manage the signal traffic between an IP network and a switched circuit network, while media gateways manage media signals between the two. Media Gateway Controllers manage traffic between SGs and MGs. The most common gateway protocols are MGCP [17] and Megaco.

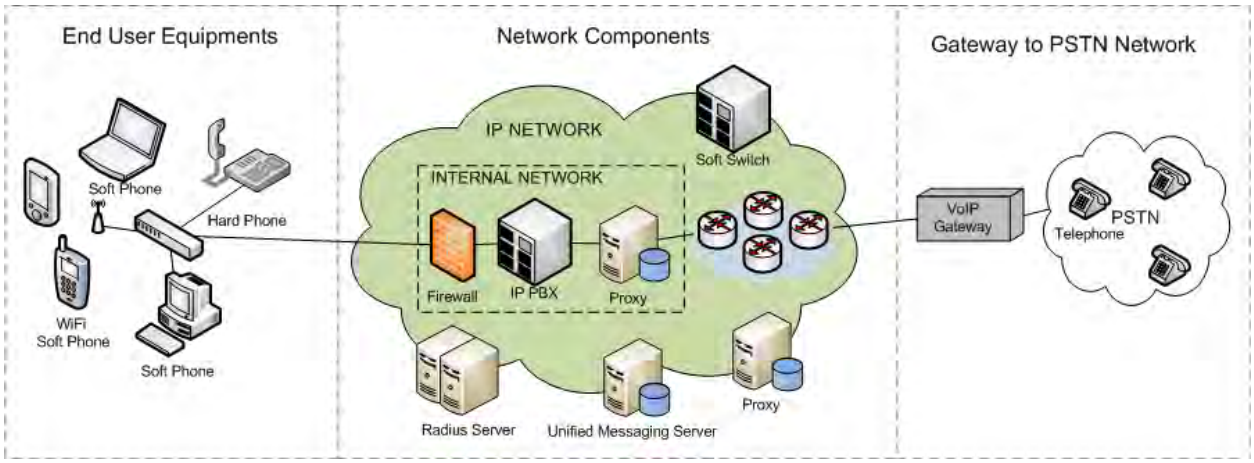


Figure 2. VoIP components and their interconnections [26]

3.2. VoIP protocols

Data communications over the Internet relies on dozens of protocols—rules defining the required steps for a communications task. For example, part of the protocol for telephone calls includes (1) waiting for dial tone before dialing and (2) the different sounds that indicate that the called telephone is either ringing or busy. The internet protocol (IP) is the basic protocol at the heart of the Internet. An IP message is the data communications equivalent of a postcard—it carries the recipient's and sender's addresses, a block of data, and little else. Other protocols, such as TCP and HTTP, build on IP to create additional capabilities, see figure 3.

Naturally enough, one kind of data that can be carried by IP is digitally encoded voice. Thus, voice over IP or VoIP. But, VoIP is not just one thing—it is many different things. It is a product/end product/ of many protocols, hence, VoIP by itself is not a protocol or a service or a device, instead it is implementation of a set of protocol that make telephony possible on an already established data communication infrastructures.

In genera VoIP or IP telephony consists two phases—***signaling phase*** and ***media/conversation/ phase,*** like any other telephone system such as PSTN where SS7 is the signaling phase and the next phase is the exchange of communications over pre-established line or channel. As a signaling phase protocol current VoIP systems use either a proprietary protocol, or one of two standards, H.323 and the Session Initiation Protocol (SIP). Although SIP seems to be gaining in popularity, neither of these protocols has become dominant in the market yet, so it is essential to understand both protocols. And in the conversation phase mainly RTP protocol is used for transporting digitized voice packets across networks between the two or more communicating parties.

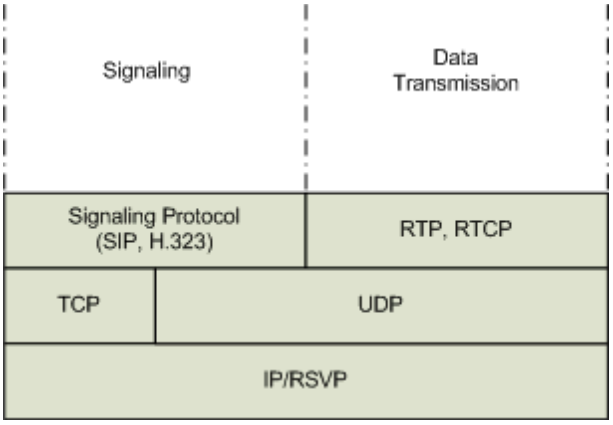


Figure 3. Essential protocols in a VoIP protocol stack

Signaling Protocols

H.323 is a set of protocols recommended by the International Telecommunication Union –Telecommunication Standardization Sector (ITU-T) and consists of family of protocols that are used for call setup, call termination, registration, authentication, and other functions (International Telecommunication Union, 2000). H.323 is widely adopted in the enterprise environment because it is a binary protocol which can be easily integrated with PSTN. An H.323 network consists of several components including Gatekeeper, Gateway, Multipoint Control Unit (MCU), and Back End Service (BES). Figure 4 shows H.323 major protocol stacks [26].

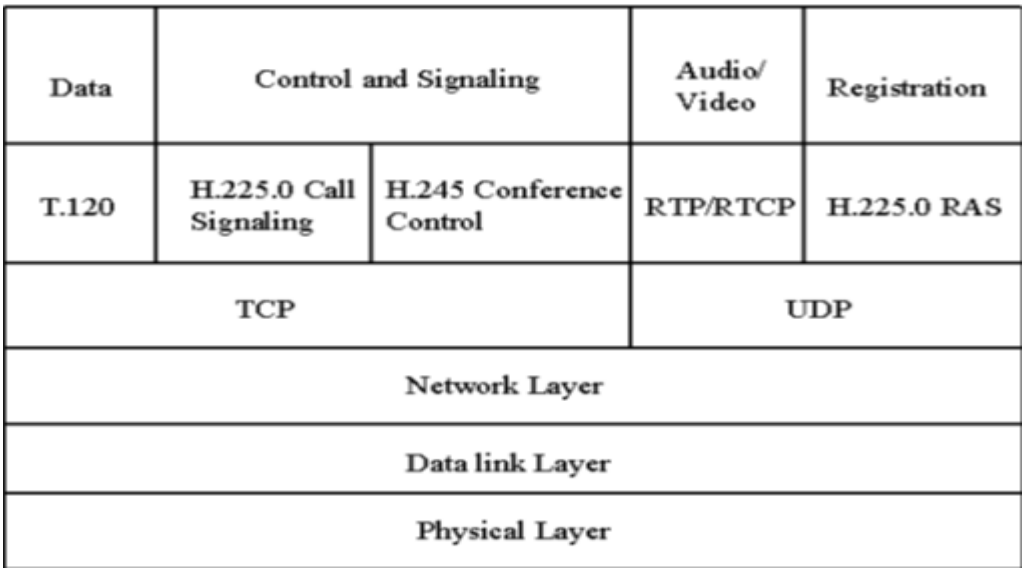


Figure 4. The protocol stack of H.323

SIP (Session Initiation Protocol) is the Internet Engineering Task Force (IETF) specified protocol for creating, modifying, and terminating unicast or multicast sessions. SIP is a text-based protocol and can transfer different types of payload with different encodings. SIP supports both UDP and TCP as transports. The architecture of a SIP network is different from the H.323 structure. A SIP network is composed of Endpoints, Proxy servers, Location servers, and Registrar servers. In section 3.3 we will see detail of SIP protocol but for now let's compare SIP and H.323 signaling protocols.

Comparison of SIP and H.323

The proponents of SIP claim that since H.323 was designed with ATM and ISDN signaling in mind, so H.323 is not well suited for controlling the voice over IP systems [26]. H.323 is inherently complex, has overheads and thus inefficient for VOIP. Additionally H.323 lacks the extensibility required of the signaling protocol for VOIP. SIP has been designed by keeping the Internet in mind, so it avoids both the complexity and extensibility pitfalls [9]. SIP reuses most of the header fields, encoding rules, error codes and authentication mechanisms of HTTP. H.323 defines hundreds of elements while SIP has only 37 headers, each with a small number of values and parameters. H.323 uses a binary representation for its messages, which is based on ASN.1 while SIP encodes its messages as text, similar to HTTP. H.323 is not very scalable as it was designed for use on a single LAN and so has some problems in scaling though newer versions have suggested techniques to get around the problem. H.323 is still limited when performing loop detection in complex multi-domain searches. It can be done state-fully by storing messages but this technique is not very scalable. On the other hand, SIP uses a loop detection method by checking the history of the message in the header fields, which can be done in a stateless manner. The advantage of SIP is that it is backed up by IETF, one of the most important standard bodies while the advantage of H.323 is that it has a much larger chunk of the market presently [9]. Table 2 lists the differences in a tabular form.

Table 2. Comparing H.323 with SIP

H.323	SIP
Complex protocol	Comparatively simpler
Binary representation for its messages	Textual representation
Requires full backward compatibility	Does not require full backward compatibility
Not very modular	Very modular
Not very scalable	Highly scalable
Complex signaling	Simple signaling
Large share of market	Backed by IETF
Hundreds of elements	Only 37 headers
Loop detection is difficult	Loop detection is comparatively easy

Media transport protocol

RTP supports the transfer of real-time media (audio and video) over packet switched networks [8]. It is used by both SIP and H.323. The transport protocol must allow the receiver to detect any losses in packets and also provide timing information so that the receiver can correctly compensate for delay jitter. The RTP header contains information that assists the receiver to reconstruct the media and also contains information specifying how the codec bit streams are broken up into packets. RTP does not reserve resources in the network but instead it provides information so that the receiver can recover in the presence of loss and jitter.

The functions provided by RTP include:

- Sequencing: The sequence number in the RTP packet is used for detecting lost packets
- Payload Identification: In the Internet, it is often required to change the encoding of the media dynamically to adjust to changing bandwidth availability. To provide this functionality, a payload identifier is included in each RTP packet to describe the encoding of the media

- Frame Indication: Video and audio are sent in logical units called frames. To indicate the beginning and end of the frame, a frame marker bit has been provided
- Source Identification: In a multicast session, we have many participants. So an identifier is required to determine the originator of the frame. For this Synchronization Source (SSRC) identifier has been provided.
- Intramedia Synchronization: To compensate for the different delay jitter for packets within the same stream, RTP provides timestamps which are needed by the play-out buffers.

RTCP is a control protocol and works in conjunction with RTP. In a RTP session, participants periodically send RTCP packets to obtain useful information about QoS etc. The additional services that RTCP provides to the participants are:

- QoS feedback: RTCP is used to report the quality of service. The information provided includes number of lost packets, Round Trip Time, jitter and this information is used by the sources to adjust their data rate.
- Session Control: By the use of the BYE packet, RTCP allows participants to indicate that they are leaving a session
- Identification: Information such as email address, name and phone number are included in the RTCP packets so that all the users can know the identities of the other users for that session.
- Intermedia Synchronization: Even though video and audio are normally sent over different streams, we need to synchronize them at the receiver so that they play together. RTCP provides the information that is required for synchronizing the streams.

Figure 5 shows the header format of RTP protocol.

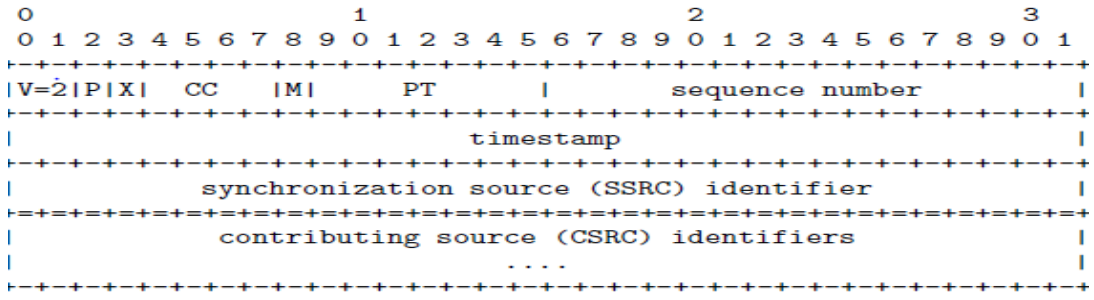


Figure 5. RTP Header format [RFC-3550]

3.3. SIP based VoIP System

3.3.1. Introduction to SIP

SIP is an application layer signaling protocol, developed by the Internet Engineering Task Force (IETF), to enable the creation, modification, and termination of sessions that are independent of the underlying transport protocols and session type being established. SIP is a client-server protocol that closely resembles two other internet protocols, HTTP and SMTP, in which requests are issued by the client and responses are managed by the servers. The messages exchanged between them contain the information required for establishing a session. SIP makes communication possible by using protocols like RTP/RTCP and SDP. The RTP protocol is used for carrying voice data in real time, while the SDP protocol is used for the description of capabilities of participants, type of encoding, ports, etc.

3.3.2. SIP Components

For the above mentioned functionalities of SIP, the protocol has two components: User Agent (UA) and SIP Servers. Let's see each component shortly.

User Agent

A user agent is an end system acting on behalf of a user. A User Agent is an application that can initiate, receive, and terminate a call in a SIP session. Or simply UA is the softphone installed on user's computer or the hard-phones that are capable of VoIP service. There are two parts to it: a client and a server. The client portion is called the User Agent Client (UAC) while the server portion is called User Agent Server (UAS). The UAC is used to initiate a SIP request while the UAS is used to receive requests and return responses on behalf of the user.

SIP Servers

The server components of the SIP infrastructure can be divided into three types:

- Proxy Server: A proxy server is an intermediate entity that receives SIP requests and forwards them on behalf of the requester. It works as a client and server to enable the call establishment between users. This server has the functionality to enclose route the requests which receives from another entities closer to the destination. There are two different types of Proxy Severs:

- — Statefull Proxy: it keeps the transaction states while requests are being processed. It lets break the requests down into several (forking), with the aim of the parallel location of the call, to obtain the best response to be send to the user who made the call.
- — Stateless Proxy: it does not keep the transaction state while requests are being processed, it only resends messages.

- Registrar Server: This is a server that accepts register requests from the user and keeps information of about this requester to offer a localization service and addresses translation in the domain controlled by it. It also supports authentication. A client must register with the registrar server each time a user turns on the SIP user client.
- Redirect Server: This is a server that creates redirect responses to the received requests. It redirects the requests through the next server. A redirect server on receiving requests determines the next-hop server and returns the address of the next-hop server to the client instead of forwarding the request.

Generally this SIP components referred as SIP architecture, will communicate to each other so that a call can be established or torn down. One simple example that illustrate the call setup between Alice and Bob is shown in figure 6 below. The figure is meant to show only the interactions and steps of call setup while a detail of SIP call flow is a bit larger than the steps shown in figure 6.

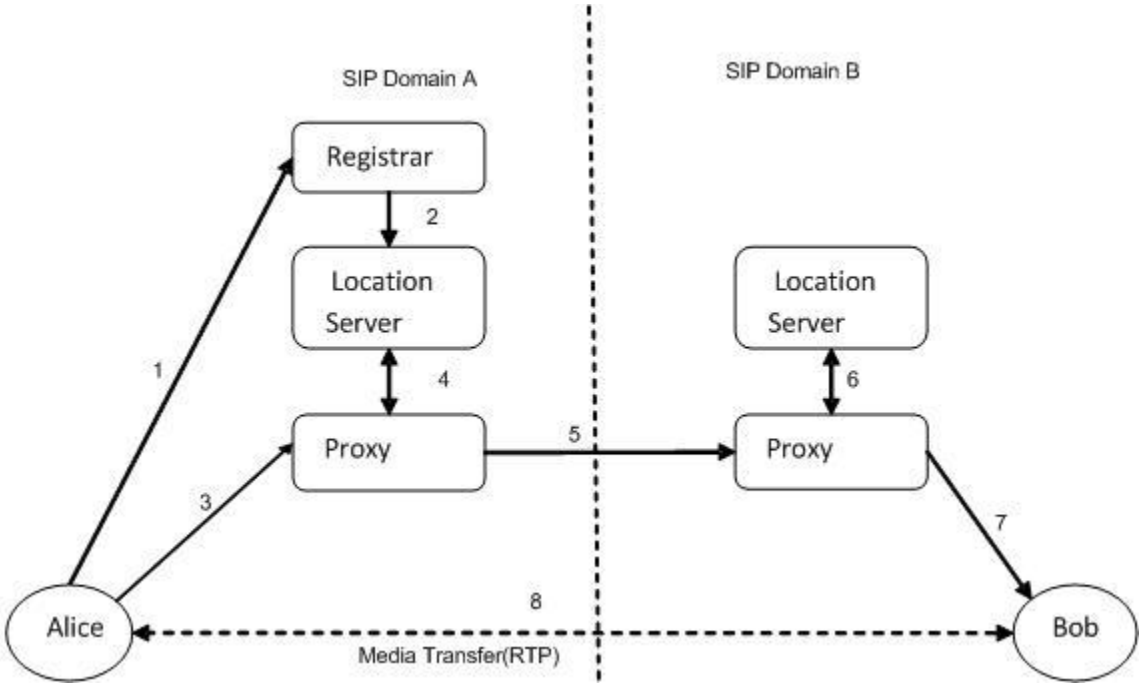


Figure 6. SIP Architecture (or interaction of SIP entities)

The steps denoted on the figure can be explained as follow: User Alice registers with her domain's registrar (1), which stores the information in the location server (2). When placing a call, Alice contacts her local proxy server(3), which then consults location server to assess Alice registration before forwarding the call(4). If success then call will be forwarded to another proxy (5) in hop by hop basis till the location server containing bobs port is found (6). The proxy who found Bob then forward the call to Bob (7). If Bob is willing to accept the call it will notify Alice by replying with an OK response. And then start RTP automatically directly between the two devices (Alice's and Bob's devices) or even through their proxies if the devices are unable to communicate directly due to their network setup.

3.3.3.SIP Message Exchanges

SIP messages come in two flavors:

- *Request*: sent from client to a server and defines the operation sought by the client.
- *Response*: sent from server to a client, and provides the status of that request.

Request

A SIP request is characterized as a method much like HTTP, and is considered a 'verb', since it requests actions to be performed by other User Agents or servers. RFC3261 defines six methods (the first six in Table 3), with subsequent standards defining the remaining extension methods (from INFO onwards).

Table 3. SIP methods

Method	Description
INVITE	Used to set up an SIP session. Session parameters are negotiated.
REGISTER	Authenticates the User Agent and provides a current location to the network.
BYE	Terminates an open session.
ACK	Confirms a success response to an INVITE. The third part to a three-way-handshake.
CANCEL	Cancels an open request. BYE should be used to cancel (tear down) an existing request.
OPTIONS	Queries the capabilities of correspondents.
Extension Methods	
INFO	Provides mid-call session-related information. It is rarely used.
MESSAGE	Used to transfer Instant Messages.
NOTIFY	Publishes the outcome of events. Used in combination with SUBSCRIBE requests.

PRACK	A Provisional Response Acknowledgment. Confirms receipt of a provisional response.
PUBLISH	Publishes status information. Used for Instant Messaging presence services.
REFER	Mechanism to pass a request to someone more appropriate to deal with it.
SUBSCRIBE	Used to request receipt of future NOTIFY or PUBLISH requests.
UPDATE	Modifies session parameters in mid-call.

Responses

SIP Response messages are always sent in reply to a request. They convey status updates, confirmations, directions, and error codes back to the UAC originating the request. Responses are characterized as either provisional or final, and every response must be identified by a 3-digit code.

Response Types

Six classes of response have been defined, and are categorized using the 3-digit code. The first five are borrowed from HTTP; the sixth is new to SIP.

Table 4. SIP Response classes

Class	**Description**
1xx Provisional	Confirms receipt of request and processing is continuing. Provisional responses to INVITEs are never ACKed.
2xx Success	The request was received, processed, and accepted.
3xx Redirection	Provides location information or alternative services to try.
4xx Request Failure	The request contained an error or cannot be processed by the server.
5xx Server Failure	The server is unable to fulfill the request because of an internal error.
6xx Global Failure	No service can be found to fulfill the request.

Within each class, numerous response codes have been predetermined - some copied from HTTP.

Table 5. Sample response codes (change this table to single lined table)

#	**Reason Phrase**	**Description**
100	Trying	The next hop received the request.
180	Ringing	Attempting to alert the user.
182	Queued	Temporarily unavailable and request is in a queue (not rejected).
200	OK	The request has succeeded.
301	Moved Permanently	User is no longer available at the address given in the Request URI.
302	Moved Temporarily	Retry the request at a new address given in the Contact header.
400	Bad Request	Could not understand or process correctly the request.
401	Unauthorized	The request either failed authentication or needs more information.
403	Forbidden	The server is refusing to process the request. Do not retry.
404	Not Found	The server cannot identify the user in its domain.
408	Request Timeout	The server could not process the request in a reasonable time.
415	Unsupported Media	The format is not supported by the server.
480	Temporarily Unavailable	The called party is currently unavailable.
485	Ambiguous	The Request URI is ambiguous.
486	Busy Here	The called party is currently not willing or able to take the call.
500	Server Internal Error	The server encountered an unexpected condition.
513	Message Too Large	The message length exceeded a determined limit.
603	Decline	The user explicitly refused to accept the request.

Warning Header Field

The *Warning* header field is used to carry additional information about the status of the response. The header defines a 3-digit code between 300 and 399, the host name, and a warning text.

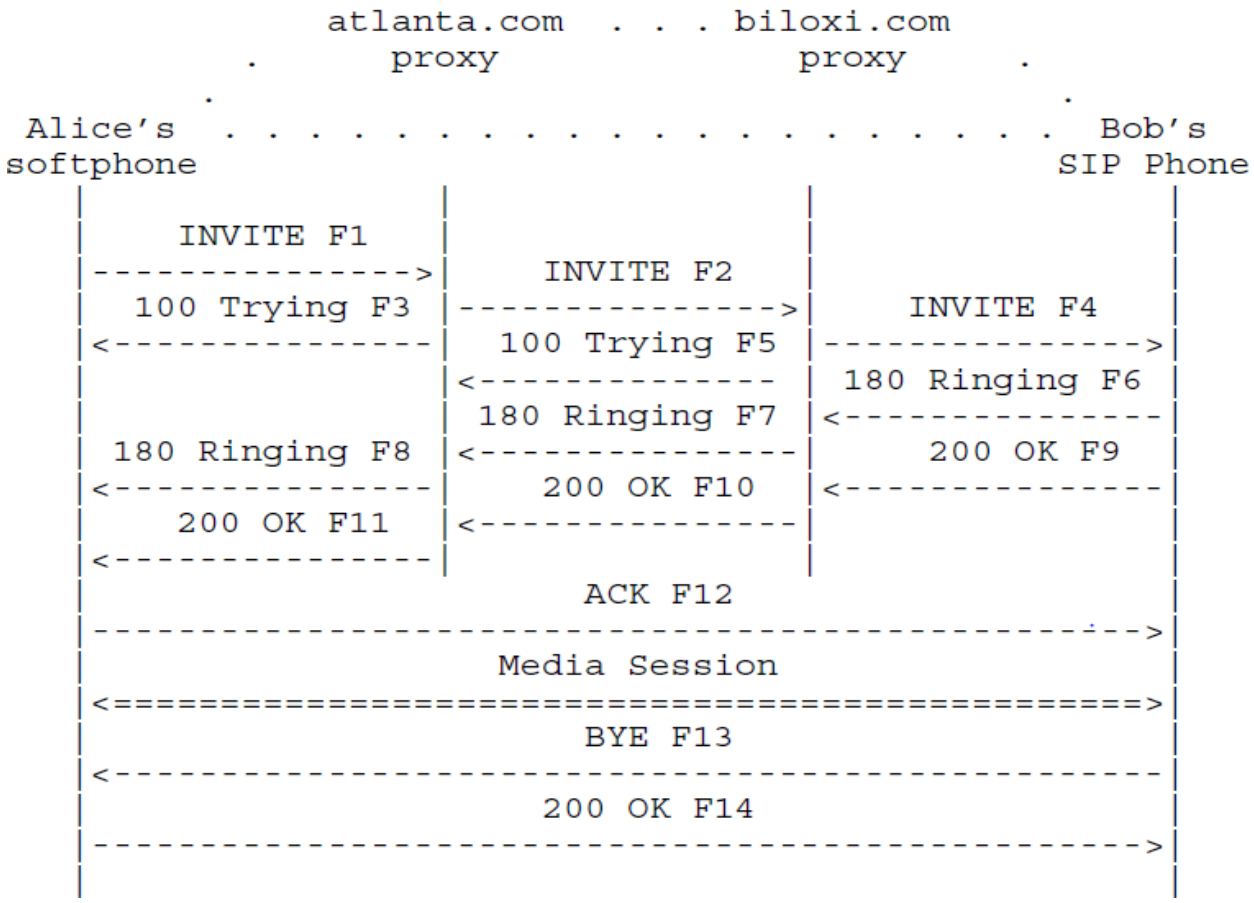


Figure 7. SIP Call Setup [RFC 3261]

Figure 7 shows a typical example of a SIP message exchange between two users, Alice and Bob. (Each message is labeled with the letter "F" and a number for reference by the text.) In this example, Alice uses a SIP application on her PC (referred to as a softphone) to call Bob on his SIP phone over the Internet. Also shown are two SIP proxy servers that act on behalf of Alice and Bob to facilitate the session establishment. This typical arrangement is often referred to as the "SIP trapezoid" as shown by the geometric shape of the dotted lines at the top portion of Figure 7. SIP uses a Uniform Resource Identifier

(URI) called SIP URIs that looks like an email address to identify users as a phone number. SIP URI is made from username and host proxy or IP-PBX's address the user register its phone. For example according to figure 7 Alice's softphone address is written as sip:alice@atlanta.com where as Bob's address could be sip:bob@biloxy.com whereas alice and bob are usernames of Alice at atlanta.com proxy server and Bob's username at biloxy.com. Just like a secure connection to internet using HTTP protocol is identified by https label instead of the http label or tag SIP URI also replaces the *sip* label by *sips* to show a secure connection. However the actual security cannot be obtained by changing the sip to sips, instead the underlying callee and caller must implement another protocol like TLS or S/MIME to carry the SIP messages securely. Generally the sip and sips label in SIP URI doesn't guaranty instead they simply tell us whether this call is secure i.e. sips label or insecure i.e. labeled by sip where as the caller must implement another security protocol to secure its connection.

As an example the INVITE (message F1 in Figure 7) might look like this:

```
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: 142
```

This example contains a minimum required set of invite request. The SDP part of Alice's INVITE message is not shown which means with this message at least Bob is unable to know what codec and RTP stream address (the IP and port of Alice's phone) that Alice intend to use for this call. That information could be included to SIP message by implementing a separate protocol called Session Description Protocol (SDP).

3.4. Vulnerabilities and Current Attacks

Before going any further one should understand the difference between the two terms, vulnerability and attack. **Vulnerability** is a risk of exposure for an attack to occur due to presence of exploitable weakness. The weakness could be a result of many contributing problems like incorrect implementation, protocol problem or underlying platform and supporting applications and protocols. **Attack** is an action taken by some malicious intruder that involves exploitation of certain vulnerabilities in order to cause an existing threat to occur.

Attackers typically target the most popular and well-publicized systems and applications. VoIP has become one of such application. Several VoIP weaknesses have been revealed recently, thus protocol designers need to address it before successfully deploying VoIP on the global scale. Not only protocol designers but mainly the implementers should implement protocols with a mandatory care for security than considering it as an added feature or advanced feature. In this section we will describe some of known or anticipated vulnerabilities or attacks on current VoIP service based on the two phases of VoIP we discussed earlier in this chapter.

To start from the signaling phase we explain the different possible or already happened attacks due to usage of SIP signaling protocol. Almost all the entities (i.e., proxy servers, registrar servers and end-user devices) of SIP based VOIP Networks are potential targets for standard Internet attacks as well as some more types of attacks that are unique to SIP. For instance, SIP Registration Server cannot challenge the authenticity of an UA which can cause registration hijacking. This can be done by altering the 'from' header of a SIP registration message, an intruder can portray as a valid UA and can register himself. An Attacker can also launch dictionary attacks to retrieve the password of an UA, and can be able to perform redirection, eavesdropping or monitoring of multimedia sessions. Any SIP server can be duplicated by sending the similar message responses to a soft-phone, the attacker can forward incoming calls to himself. This enables an attacker to track the call and to perform selective DoS attacks.

Fake Proxy Server that is capable of reading and modifying the message header and body can be used to tamper SIP messages, DoS attacks can also be launched by redirecting the messages to non-existing server. A malicious user can forward 'BYE' and 'CANCEL' messages to close down the SIP sessions.

These and many more attacks [25], [31] are to be prevalent sooner than later as VoIP become an alternative to the current PSTN. The second phase of VoIP, the actual phase that matters to the dialer, has also some formidable attacks that can be easily anticipated if security is not dealt as a must requirement than a feature.

Since our objective is mainly to address the confidentiality and integrity breaches in this thesis then it will be good move if we categorize the current attacks or vulnerabilities based on the two information security principles. That is what we will do in the next sections.

3.4.1. Confidentiality Threats

Confidentiality means that the information cannot be accessed by unauthorized parties. The confidential information of end users includes private documentation, financial information, security information like password, conversation content, conversation history or pattern, etc. The confidential information for network components includes operation systems, IP addresses, protocols used, address mapping, user records, etc. Leak of this information might make attackers' jobs easier. Some of the major confidentiality threats are discussed as follows.

Eavesdropping of phone conversation

Conventional telephone eavesdropping requires either physical access to tap a line, or penetration of a switch. With VoIP, opportunities for eavesdroppers increase dramatically because of the large number of nodes in the path between two conservation entities. If the attacker compromises any of these nodes, he can access the IP packets flowing through that node. There are many free network analyzers and packet capture tools that can convert VoIP traffic to wave files [27]. These tools allow the attackers to save the

conversation into the files and play them back on a computer. VoMIT (Voice over Misconfigured Internet Telephones) is an example of such a tool [23]. Ethereal or Wire-shark can also be used to record SIP packets and retrieve voice message in wav file format.

Unauthorized access attack

Unauthorized access means that the attacker(s) can access resources on a network that they do not have the authority. Shawn Merdinger reported multiple undocumented ports and services in certain VoIP phones [29]. There are also vulnerabilities due to implementation issues [11]. There are systems for call control, administration, billing and other voice telephone functions. Repositories in these systems may contain passwords, user identities, phone numbers, and private personal information. Lots of gateways and switches are shipped with default well-known passwords. If these passwords are left without changes, the attackers can easily break in. Some switches still use TELNET for remote access. The clear-text protocol exposes everything to anyone who can sniff the network traffic. Some of the gateways or switches might have a web server interfaces for remote control. The attacker might sniff the HTTP traffic in local network to steal sensitive information. Attackers can also use ARP cache poisoning to forward all traffic through their machines to capture network traffic.

3.4.2. Integrity Threats

Integrity of information means that information remains unaltered by unauthorized users. A legitimate user may perform an incorrect or unauthorized operations function and may cause delirious modification, destruction, deletion or disclosure of switch software and data. An intruder may masquerade as a legitimate user and access an operation port of the switch.

Caller Identification spoofing

Caller ID (Caller Identification) is usually a service provided by most telephone companies that tell users the phone number of an incoming call. Caller Identification spoofing is setting the Caller ID on the outgoing calls to a 10 digit number of the caller's choice. Several websites provide Caller ID spoofing service eliminating the need for any

special hardware. The list includes www.spooftel.com, www.telespoof.com, www.callnotes.net, www.spoofcard.com, etc.

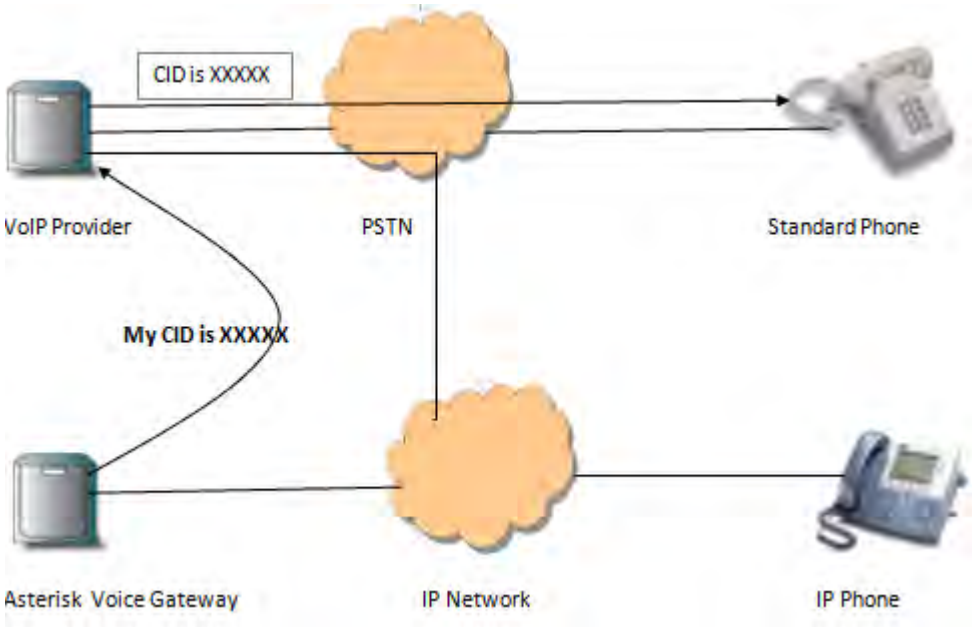


Figure 8. Illustration of Caller ID spoofing

For VoIP, Caller ID spoofing is simpler than traditional telephone as illustrated in Figure 8. Suppose Alice sent an "INVITE" message to Bob and the message is like the following (the SDP of Alice is not shown here):

```
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: 142
```

The caller ID service relies on the "From" header to supply the identity. Call-ID contains a globally unique identifier for this call and has nothing to do with Caller ID. If the attacker can control the gateway server, he can arbitrarily change "From" header to anything that he wants. The recipient will send back acknowledgment to this proxy

server, which is the same as "via" field. The proxy server can then forward the acknowledgment to Alice since it knows the real IP address of Alice's phone.

Automated or manual caller ID verification systems such as used by credit card companies can be sent false information. Lance James, chief scientist at security company Secure Science Corp., said Caller ID spoofing Web sites are used by people who buy stolen credit card numbers. They will call a service such as Western Union, setting Caller ID to appear to originate from the card holder's home, and use the credit card number to order cash transfers that they then pick up [23]. Spammers can use this feature to spam or run phishing attacks posing as banks or other trusted parties.

Registration hijacking

Registration hijacking happens when an attacker replace the legitimate registration of the victim with his address. The attack causes all incoming calls for the victim to be sent to the attacker's address. Registration is normally performed using UDP, which make it easy to spoof registration requests. For example, Alice wants to register her address at registrar using SIP protocol. The "REGISTER" message looks like the following:

```
REGISTER sip:alice@atlanta.com SIP/2.0
Via: SIP/2.0/UDP 192.168.2.10;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Alice <sip:alice@atlanta.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@192.168.2.3
CSeq: 314159 INVITE
Contact: Alice <sip:alice@192.168.1.11:5061>;expire=60
Content-Type: application/sdp
Content-Length: 142
```

In this message, the "To" and "From" fields use the same user information. The "contact" field contains a SIP URI that represents a direct route to the device. In this example, it is IP address 192.168.1.11 and the port is 5061. "expires=60" means that the registration will expire in 60 seconds. Another REGISTER request should be sent to refresh the user's registration. The attacker can construct a similar REGISTER message with modified "contact" header. A possible example is like this message:

```
REGISTER sip:alice@atlanta.com SIP/2.0
Via: SIP/2.0/UDP 192.168.2.10;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Alice <sip:alice@atlanta.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@192.168.2.3
CSeq: 314159 INVITE
Contact: Alice <sip:alice@192.168.1.22:5061>;expire=60
Content-Type: application/sdp
Content-Length: 142
```

In this example, the registered IP address for Alice is changed from 192.168.1.11 to 192.168.1.22. Fearing the victim's legitimate register requests might replace his registration, the attacker can use denial of service attack to disable the victim. The attacker can also send spoofed requests at a higher frequency than the victim.

Proxy Impersonation

Proxy impersonation attack tricks the victim into communicating with a rogue proxy set up by the attacker. Once an attacker impersonates a proxy, he has complete control of the call. Figure 9 illustrates proxy impersonation. The attacker tricks Alice to communicate with the rogue proxy server instead of the legitimate proxy server. The UAs and proxies normally communicate using UDP and do not require strong authentication to communicate with another proxy. The attack can work by several means, including DNS (Domain Name Service) spoofing, ARP (Address Resolution Protocol) cache spoofing, DHCP spoofing, or changing proxy address for a SIP phone.

Call redirection or hijacking

Call redirection occurs when a call is intercepted and rerouted through a different path before reaching the destination. Possible methods include proxy impersonation and registration spoofing. The attacker can also spoof the response from the recipient and trick the requestor to talk with the attacker.

In this example, Alice wants to talk with Bob about some business secrets, which interests Tim. Tim is in the same LAN with Alice. The man-in-the-middle attack by Tim follows the below steps:

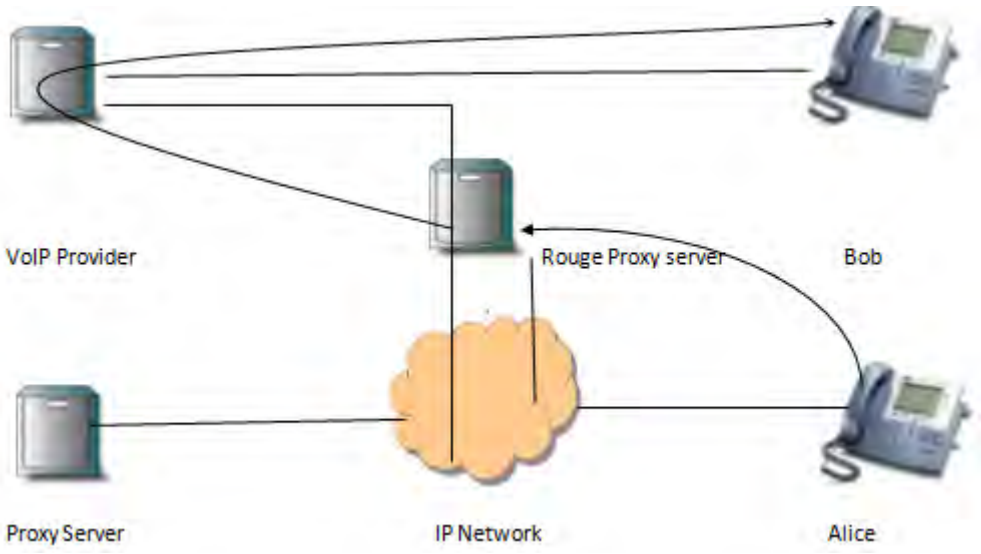


Figure 9. Illustration of Proxy Impersonation attack

1. Alice sends an Invite message to Bob and this message is detected by Tim.
2. Tim sends a response message to Alice spoofing from Bob with 301 Moved Permanently code. In the response, Tim set the new address of Alice to his computer.
3. Alice sends a new Invite message to Tim in belief that she is connecting to Bob
4. Tim sends back an Acknowledgement to establish the connection between him and Alice
5. At the same time, Tim sends an Invite message to Bob and he can also fake the caller ID of Alice.
6. Tim replies with a 200 OK and the connections between Bob and Tim is established.

Tim can use specific software to forward voice messages between two connections. He can also record the conversion contents. This is a man-in-the-middle attack. Even encryption is deployed in both connections; Tim can still access whole conversation in certain conditions, like ZRTP key exchange is used and Tim manage to fool ZRTP even though the chances are too small to succeed for this attack.

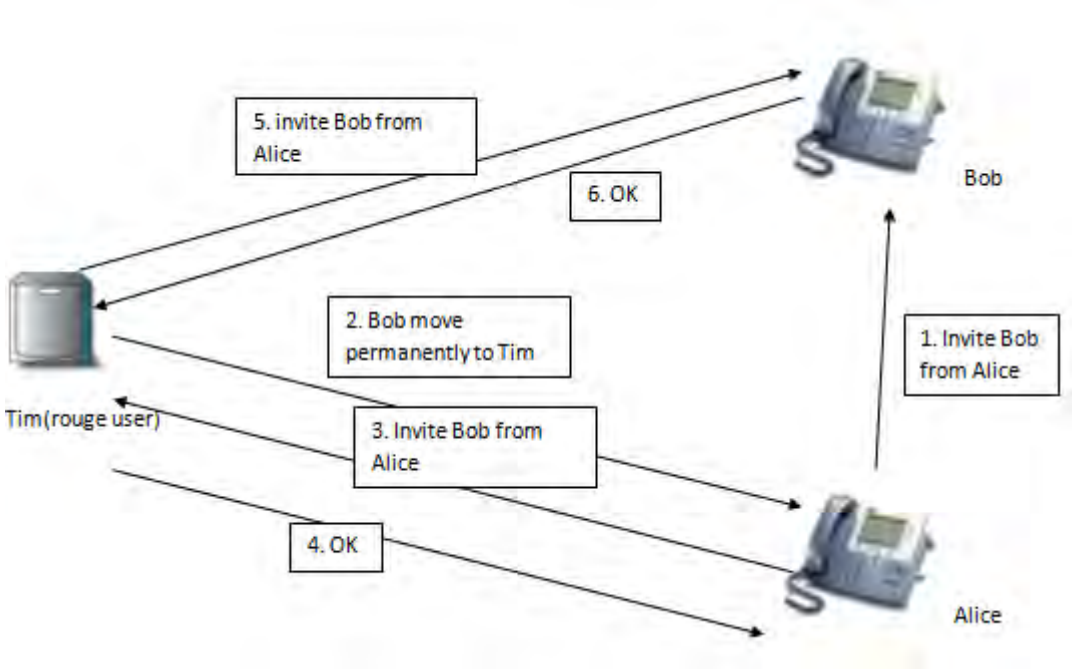


Figure 10. Illustration of Man-in-middle attack using spoof response

Call redirection or hijacking enables the attacker to eardrop even encrypted voice conversation. The attacker also can tamper the voice message sent both ways. They can also carry out replay attacks.

The third and may be the most likely than the confidentiality and integrity attack and dangerous threat to VoIP according to VoIPSA (Voice over IP Security Alliance) is an attack on the ***availability*** of VoIP service. VoIPSA is a vendor-neutral, not for profit organization composed of VoIP and security vendors, organizations and individuals with an interest in securing VoIP protocols, products and installations[12],[2]. VoIPSA categorized and analyzed VoIP vulnerabilities in six categories (i.e. social threats, denial of service, physical access, Eavesdropping and hijacking, service abuse and interruption of service) in which denial of service covers the 57% of the threats, which is an availability attack. Again the vulnerability breakdown according to the traditional (Confidentiality, Integrity, Availability), figure 11, security concerns reflects the observation depicted in figure 11 as disclosed by VoIPSA. Both classification shows that denial of service, which is an availability attack, is a predominant vulnerability concern for VoIP.

However, due to the limitation in the scope of this thesis as explained in chapter one our objective is to propose, design and implement a security countermeasure for confidentiality and integrity vulnerabilities of VoIP while improving performance or quality of service by leaving availability to be dealt in future works.

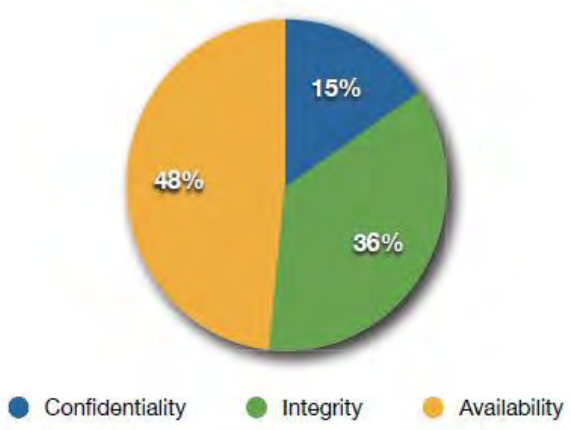


Figure 11. VoIPSA VoIP threat categorization based on Confidentiality, Integrity and availability

Generally are different kinds of classification of VoIP current vulnerabilities as well as attacks. In [28] we can find similar kind of classification as we did in this section except that, this classification also includes Availability vulnerabilities plus the vulnerabilities are categorized based on TCP/IP network model layers. Figure 12, shows the classification in [28].

	Vulnerability	Confidentiality	Integrity	Availability
Data Link	Physical Attacks	x		x
	ARP cache	x	x	x
	ARP flood			x
	MAC spoofing	x	x	x
Internet	IP spoofing			
	Registration server, IP phone, MGCP, DNS, etc	x	x	x
	Redirect via IP spoof	x	x	x
	Malformed packets	x	x	x
	IP frag	x	x	x
	Jolt			x
Transport	TCP / UDP flood			x
	TCP / UDP replay	x	x	
Application	TFTP server insertion		x	
	DHCP server insertion (redirect)		x	
	DHCP IP address starvation			x
	ICMP flood			x
	SIP			
	Registration Hijacking	x	x	x
	Call Hijacking (MGCP NotifiedEntitiy parameter)	x	x	x
	Message body modification	x	x	
	RTP insertion			
	Spoof via header	x	x	x
	Cancel / bye attack			x
	Malformed method			x
	Redirect method	x		x
	RTP			
	SDP redirect			x
	RTP payload			x
	RTP message tampering	x	x	x
	Encryption	x	x	x
	Default settings / passwords	x	x	x
	Disable unnecessary services HTTP, FTP, etc	x	x	x
	Buffer overflow	x	x	x
	Legacy Network Interaction	x	x	x
	DNS Availability			x

Figure 12: VoIP vulnerabilities based on protocol layers

3.5. Security Consideration

3.5.1. Signaling Security Consideration

Encrypting a plain text message/data/ is a countermeasure taken to protect the confidentiality/privacy/ of a message while in some cases it also gives information about whether the data's integrity altered or not. We have described that SIP is a text based protocol like HTTP and SMTP. Thus best security countermeasures proposed for guarantying confidentiality and data integrity in HTTP and SMTP can also be applied to SIP security. Those security mechanisms include IPsec, S/MIME and TLS. All the three mechanisms has their own advantage and disadvantages, however we will describe TLS protocol in detail since it is the preferred protocol by this thesis work due to the following reasons:

- Provides complete protection of data integrity and privacy same as IPsec with reduced infrastructure and processing capacity; easy to implement than IPsec since IPsec requires kernel level manipulation to implement.
- S/MIME leaves some confidential headers of SIP unprotected; To, From, Contact, Call ID, CSeq and Via header are not encrypted by S/MIME leaving those information open for unintended third body where as TLS encrypts the whole SIP header and provide extended confidentiality

TLS

TLS, an acronym for **T**ransport **L**ayer **S**ecurity, is a feature of mail servers designed to secure the transmission of electronic mail from one server to another using encryption technology. TLS can reduce the risk of eavesdropping, tampering, and message forgery mail communications. TLS is a security protocol from the Internet Engineering Task Force (IETF) that is based on the Secure Sockets Layer (SSL) 3.0 protocol developed by Netscape.

The TLS protocol is made up of two layers.

- The *TLS record protocol* is designed to protect confidentiality by using symmetric data encryption.

- The *TLS handshake protocol* allows authentication between the server and client and the negotiation of an encryption algorithm and cryptographic keys before the application protocol transmits or receives any data.

TLS is the successor to Secure Sockets Layer (SSL). SSL and TLS are frameworks that include cryptographic protocols which are intended to provide secure communications on the Internet.

They use X.509 certificates and hence asymmetric cryptography to assure the counterparty with whom they are communicating, and to exchange a symmetric key. This session key is then used to encrypt data flowing between the parties. This allows for data/message confidentiality and message authentication codes for message integrity and as a by-product, message authentication. Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, Internet faxing, instant messaging, and voice-over-IP (VoIP). An important property in this context is forward secrecy, so the short term session key cannot be derived from the long term asymmetric secret key [30].

In the TCP/IP model view, TLS and SSL encrypt the data of network connections at a lower sub layer of its application layer. In OSI model equivalences, TLS/SSL is initialized at layer 5 (the session layer) then works at layer 6 (the presentation layer): first the session layer has a handshake using an asymmetric cipher in order to establish cipher settings and a shared key for that session; then the presentation layer encrypts the rest of the communication using a symmetric cipher and that session key. In both models, TLS and SSL work on behalf of the underlying transport layer, whose segments carry encrypted data[22].

Once the client and server have decided to use TLS, may be by dedicating a specific ports for TLS which is commonly port 5061, then they negotiate a statefull connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters used to establish the connection's security:

1. The client sends the server the client's SSL version number, cipher settings, session-specific data, and other information that the server needs to communicate with the client using SSL.
2. The server sends the client the server's SSL version number, cipher settings, session-specific data, and other information that the client needs to communicate with the server over SSL. The server also sends its own certificate, and if the client is requesting a server resource that requires client authentication, the server requests the client's certificate.
3. The client uses the information sent by the server to authenticate the server—e.g., in the case of a web browser connecting to a web server, the browser checks whether the received certificate's subject name actually matches the name of the server being contacted, whether the issuer of the certificate is a trusted certificate authority, whether the certificate has expired, and, ideally, whether the certificate has been revoked. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client proceeds to the next step.
4. Using all data generated in the handshake thus far, the client (with the cooperation of the server, depending on the cipher in use) creates the pre-master secret for the session, encrypts it with the server's public key (obtained from the server's certificate, sent in step 2), and then sends the encrypted pre-master secret to the server.
5. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case, the client sends both the signed data and the client's own certificate to the server along with the encrypted pre-master secret.
6. If the server has requested client authentication, the server attempts to authenticate the client. If the client cannot be authenticated, the session ends. If the client can be successfully authenticated, the server uses its private key to decrypt the pre-master secret, and then performs a series of steps (which the client

also performs, starting from the same pre-master secret) to generate the master secret.

7. Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity (that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL connection).
8. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.
9. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.

Figure 13 shows these steps in simplified way.

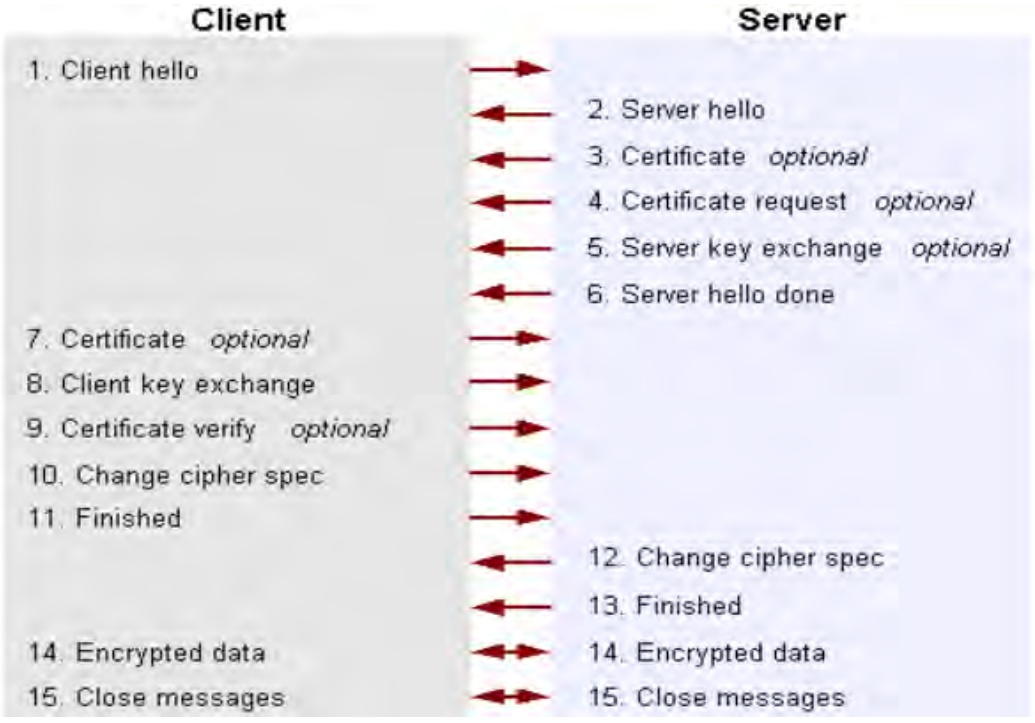


Figure 13. SSL/TLS handshake

The SSL handshake is now complete and the session begins. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

This is the normal operation condition of the secure channel. At any time, due to internal or external stimulus (either automation or user intervention), either side may renegotiate the connection, in which case, the process repeats itself.

This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes.

If any one of the above steps fails, the TLS handshake fails and the connection is not created.

In step 3, the client must check a chain of "signatures" from a "root of trust" built into, or added to, the client. The client must *also* check that none of these have been revoked; this is not often implemented correctly, but is a requirement of any public-key authentication system. If the particular signer beginning this server's chain is trusted, and all signatures in the chain remain trusted, then the Certificate (thus the server) is trusted.

3.5.2. Media Security Consideration

The media, in this case the audio data, is transported end to end using Real Time Transport Protocol (RTP). RTP exchanges packets in clear text, so there were many security issues as the confidentiality of the voice data was at serious risk. Hence a Secure RTP (SRTP) to provide message authentication, replay protection and confidentiality to the clear text RTP traffic has been developed. Conceptually, SRTP can be seen as a "bump in stack" implementation as shown in Figure 14, that exists between the RTP layer and the transport layer. SRTP intercepts RTP packets and then forwards SRTP packet on the sender side and intercepts SRTP packets and passes an equivalent RTP packet up the stack on the receiver side [7].

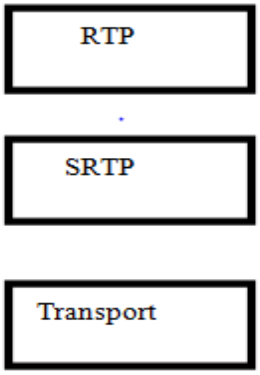


Figure 14. SRTP as "bump in stack"

In order to achieve the goals of confidentiality, replay protection and message authentication, SRTP defines extensions to the RTP packet format to encrypt the RTP payload. Figure 14 shows the SRTP packet format. To see the changes made to RTP protocol by SRTP protocol you may compare figure 14 with figure 5. The SRTP stream requires that the sender and receiver to maintain crypto-graphic state information. The cryptographic context provides all the necessary parameters such as the chosen cipher, its mode of operatio.n and the block size, the master key and the session key etc.

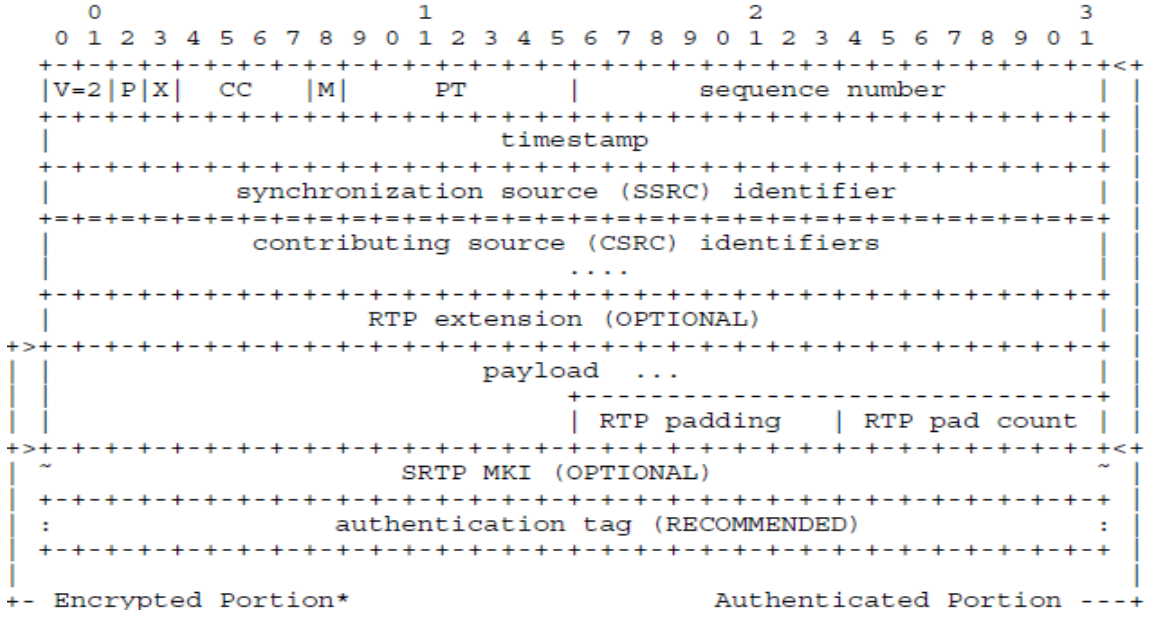


Figure 15. SRTP protocol format

There are two types of keys used by SRTP one is the session key and other is the master key. The session key is used directly in the payload encryption or message authentication and the random bit string provided by the key management protocol from which the session keys are derived in a cryptographically secure manner. The salt key, master key and other parameters in the cryptographic context are provided by the various key management mechanisms such as SDES, ZRTP and MIKEY.

Generally, SRTP provides confidentiality, integrity and authentication of message. It provides protections against replay attacks for both RTP and RTCP. It supports AES which allows for out-of-order packet reception and processing. It also minimizes computation and resource consumption for generation cryptographic keys. However, the security threat of SRTP doesn't come, well primarily, from the lack of strength of SRTP itself instead the vulnerability is induced due to the problems on the key exchange mechanisms since SRTP doesn't have a mechanism of generating its own shared secret key between the peers unless the keys are derived prior the start of SRTP. As stated in the above paragraph some of most widely used key exchange or establishment mechanisms include SDES, MIKEY and ZRTP. We will shortly describe these key exchange or establishment mechanisms since our proposed technique is slight change of one of those key exchange mechanisms.

SDES

SDES is the simplest form of key management protocols to understand. SDES is the key transport extension of the SDP protocol. SDES defines a new SDP attribute called "crypto" which is used to signal and negotiate cryptographic parameters for SRTP media streams [6]. This attribute transports the encryption and the authentication algorithms, master key and salt of the sender (i.e. the receiver should use the said master key and salt to derive session keys for decryption .The crypto attribute for SRTP is defined as: ***a= crypto :(tag) (crypto-suite) (key-params) [(session-params)].*** The most important component is key-params, which specifies one or more cryptographic keys as (key-info).

In this simplest form, the UAC inserts this parameter in the SDP of the INVITE request and send it to UAS; the UAS inserts this parameter in the 200 OK responses and

transmits it to the UAC. Consequently SDES provides unique keys for each media stream in each direction. For example when Alice calls Bob if she decide to make secure media session using SDES key exchange then the SDP part of her INVITE message might look like the following:

```
...
v=0
o=alice 2890844526 2890842807 IN IP4 10.47.16.5
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
```

(Alice's INVITE message part not involved, but represented by the three dots as shown above)

The "a=crypto" line is actually one long line; it is shown as two lines due to page formatting. The crypto attribute above identifies the encryption and authentication algorithm and specifies the master key, salt and the lifetime of the master key (2^20). The master key and salt are concatenated and base 64 encoded .The sender of the “crypto” attribute uses the master key to derive the session key for encryption and the receiver uses it to derive the session key for decryption.

SDES is considered secure because the messages are carried on hop-by-hop TLS encryption. However, intermediaries on each hop-by-hop link obtain unencrypted message and consider that an adversary manage to inject itself as a next hop then boom man in the middle attack. This leaves SDES as the weakest though simple to implement.

ZRTP

ZRTP is a key agreement protocol that performs a Diffie-Hellman key exchange during call setup in the media path and is transported over the same port as the **Real-time Transport Protocol (RTP)** media stream which has been established using a signaling protocol such as **Session Initiation Protocol (SIP)**. This generates a shared secret, which is then used to generate keys and salt for a **Secure RTP (SRTP)** session. More simply ZRTP is an interaction between two parties, defined as an Initiator and a

Responder. Figure 16 obtained from the protocol spec RFC 6189 illustrate the flow of a typical ZRTP transaction:

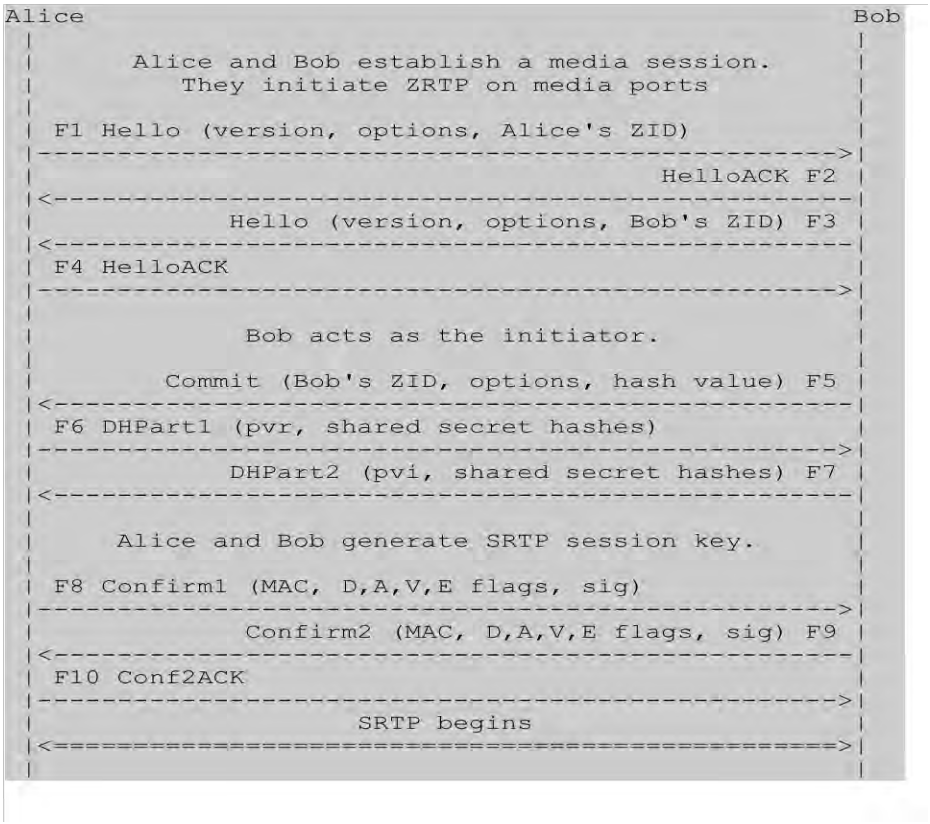


Figure 16. A typical ZRTP key exchange process/or call/ flow

(D, A, V and E are flags: Disclosure flag (D), the Allow Clear flag (A), the SAS Verified flag (V), and the PBX Enrollment flag (E).)

The protocol breaks down into roughly four phases:

1. **Discovery and protocol negotiation** (F1-F4): In this phase the parties start up a protocol transaction and agree on a supported ZRTP version and cipher suites.
2. **Diffie-Hellman key establishment** (F5-F7): Assuming that the two parties have successfully completed the negotiation, the next phase of the protocol requires the two parties to establish a shared key. This is done using Diffie-Hellman key exchange mechanism. This is the core phase of the protocol where the communicating parties generate a secret shared private key.
3. **Key confirmation** (F8-F10): The parties verify that they've agreed on the same key.

4. **Secure communication.** This is the last phase shown on the figure labeled as "SRTP begins" and it is the actual stage where the communicating peers started encrypted communication using the secret key they managed to share securely over insecure channel, internet. One can see the protocol spec for complete understanding of the protocol and its implementation from RFC 6189.

Generally one notable feature of the ZRTP protocol is that it does not require a Public Key Infrastructure (PKI). This feature is a clear advantage of ZRTP as the user levels of the Public Key Infrastructure are very hard to maintain and here the users are free from the certification authority. If the certification authority is in use, then there is a need for each user to hold his or her unique digital certificate, moreover the operational cost of maintaining such an authority is high and it requires high cost of issuing, revoking and renewing user level certificates. The other big advantage of the protocol as its creators say, P. Zimmermann, et al, is the capacity of the protocol giving double security feature to prevent an adversary from breaching the confidentiality and integrity of the secure communication. These two stage securities are *(1)* every time we establish a connection with some remote party, we can verbally compare a "short authentication string" (SAS) to ensure that we've both agreed on the same key. Assuming that our attacker *isn't* a voice actor, this should let us easily detect a typical MiTM. And just in case we're too lazy to bother with this **voice authentication** or identification, then completing one 'un-attacked' connection leaves us with *(2)* a long-term shared secret that we can use to validate future connection attempts, **key continuity**[20],[18]. One more point that is considered as an advantage of ZRTP is its independence of signaling protocols. One can use either SIP or H.32.3 or other proprietary protocol to establish session and once RTP starts the callers can start secure communication anytime he or she wants irrespective of the protocol used for signaling.

MIKEY

MIKEY is an another key management protocol which is used in VoIP .The Multimedia Internet Keying [14] was designed to meet the requirements of initiation of secure multimedia sessions. In MIKEY the parameters for the security protocol should be exchanged in one round trip. It was designed to make the means of key exchange simpler

and straight forward. The MIKEY protocol provides end-to end security for the keying material and moreover it is independent from any specific security functionality of the underlying transport. Another notable feature of MIKEY is that it consumes low bandwidth consumption and low computational workload.

MIKEY supports three types of key agreements

1. Pre-shared key (PSK) – In this method the key derivation for both the encryption and the integrity of the message purpose the pre-shared secret is used and. The randomly generated TGK of the MIKEY message is securely transported. This key agreement mode is considered to the most cost effective scheme but the drawback of this scheme is that the shared key distribution leads to scalability issues.
2. Public-Key Encryption (PKE) - This method is mostly similar to the above method, the only difference is that here the initiator makes use of a random key for encryption and integrity.
3. Diffie-Hellman (DH) - This method provides perfect forward secrecy. This method can be used in peer-to-peer keys, but this method proves to be resource consuming. Further for the purpose of a message signing the existence of PKI is a must.

Chapter Four

4. System Design

So far in different portion of the document, particularly at section 3.4, we have described that leaving the signaling layer in plain without any reliable security caution and focusing on securing the actual communicated voice either by encryption or any possible method is not a complete security or even not security at all as SIP, the signaling, is equally necessary to the service and not secured. SIP messages frequently contain sensitive information about their senders - not just what they have to say, but with whom they communicate, when they communicate and for how long, and from where they participate in sessions. Many applications and their users require that this sort of private information be hidden from any parties that do not need to know it. So if complete security is desired one must secure both phases. Thus in this chapter we describe the security mechanisms that we propose to secure VoIP's both phases. The proposed mechanism relays on the strength of prior strong and successfully implemented protocols described in section 3.5 by making them work together in order to yield greater security and increased performance.

4.1. System Design Diagram

The overall design of the system is as shown in figure 17. As seen on the figure the newly proposed key-establishment technique, MZRTP, is sandwiched between the signaling protocol SIP and the media layer protocol SRTP. This means part of MZRTP starts during the TLS protected TCP connection as part of the SIP invite message while the rest part or step will be completed over the UDP connection established for RTP packet transportation. As soon as the MZRTP part completed successfully, meaning both parties obtained a common shared session keys and salts, then the voices or audio payload of the communicators start to be encrypted/and decrypted using SRTP protocol specification. Since in prior sections or chapters of this document we have explained the implementation or detailed description of TLS and SRTP plus SIP and RTP, in this section we will describe the newly introduced term, MZRTP.

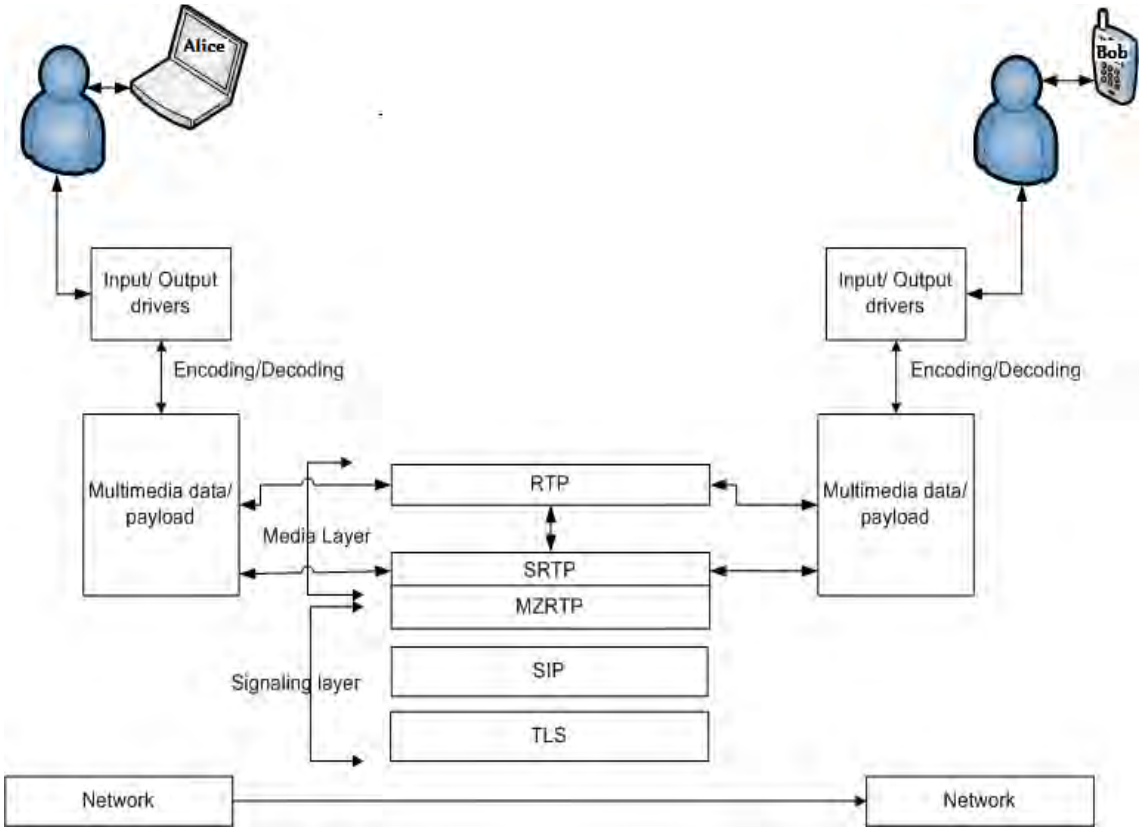


Figure 17. Over all Secure VoIP designed system diagram

4.2. Description of the proposed MZRTP

MZRTP is not an exclusively full fledge new key exchange mechanism addition to the existing lists of key exchange mechanisms of VoIP like SDES, MIKEY and ZRTP instead it is a slight modification of ZRTP so that ZRTP, the media layer protocol, will be used efficiently with SIP, the signaling layer protocol, in an integrity protected channel, TLS capable connection channel, to maintain confidentiality and integrity of both the signaling and media/conversation/ phases of VoIP service.

As clearly pointed out in the ZRTP spec, i.e. RFC 6189, ZRTP is independent of any signaling layer protocol and this is considered as one of the biggest advantages of ZRTP, even though the spec also describe how to associate ZRTP with signaling layer protocol if one found it necessary. Specially associating ZRTP with signaling layer protocol like SIP is found necessary in order to notify the other end peer on the call setup as this call is intended to be secure using ZRTP key exchange mechanism. Let's assume that Alice

wants to talk to Bob securely but Alice knows that her phone can support ZRTP,SDES and MIKEY and she doesn't know which of this three choices Bob's device capable of parsing. So in order to have a common understanding they must negotiate the security mechanism they both support during the session establishment which means ZRTP support capability **SHOULD** be indicated in the signaling phase otherwise Alice phone must have another button that sends ZRTP hello message in the middle of the talk to notify Bob's phone/device/ that Alice wants to make the established communication to be encrypted communication. The idea in this paragraph is written in section 8.1 of the ZRTP spec document, RFC 6189.

The RFC suggests that if reliable integrity protected channel is available, one can send ZRTP's hello message hash by including it in SDP part of signaling layer using an attribute "zrtp-hash" as follow:

a="zrtp-hash" zrtp-version hello-hash

Actually sending the hello message is good even if the channel is not secure but we may not depend on the sent message as a reliable. *According to the RFC sending this hello message hash in an integrity protected channel during the signaling phase is used to detect falsely injected hello message from an intruder by comparing this message to the one sent during the media layer when the actual ZRTP starts.* However given availability of integrity protected channel like TLS, one doesn't have to resend the Hello message again in the media layer because the commit message (F5[1]) does have the hash of the hello message(F1 and F3) and in order to check if there is no falsely injected message one can just compare the Hello hash in the commit to the one set during the media layer. This is what MZRTP removes from ZRTP. The advantages of not starting ZRTP with hello message resent on the media layer and instead sending the hello messages on the signaling phase with SIP(Or in short the advantages of MZRTP are) are the following:

[1] All numbered F's like F1, F2, etc are ZRTP message flows indicated by arrows as shown Figure 16, here and hereafter in the whole document

- F1 to F4 of the ZRTP will be no more needed hence reduces the number of message exchange made to establish secure media communication from 10 to 6. See figure 16 and section 3.5.2 to refresh the 10 stages of ZRTP.
- The reduced steps in the ZRTP contribute to fast completion of ZRTP and hence increased performance will be obtained, in a context of time required to setup a secure call.
- Since the devices negotiated to use ZRTP at the signaling layer and part of the ZRTP key exchange already started during the signaling there is no need to add a button like Go Secure as suggested on the ZRTP spec to start the ZRTP exchange instead the ZRTP key exchange will be automatically started at the end of signaling, i.e. SIP, hence reduces the complexity of implementation and improves the GUI of the phone too. Actually currently existing ZRTP can be made automatic, meaning without requiring pressing any Go Secure buttons, but this requires the devices to send repeatedly hello message at the end of SIP until the other end device recognize the hello message and initiate its own ZRTP portion by sending equivalent hello message. This is the technique used currently on most devices.

Now we will see detail of the proposed MZRTP implementation normatively.

MZRTP proposes addition of new attribute in the SDP part of SIP Invite message, irrespective of the currently existing zrtp-hash attribute, this is because of backward compatibility with the existing ZRTP implementation. We denoted the new attribute as **mzrtp-hash**.

Description of mzrtp-hash

mzrtp-hash has completely similar structure to zrtp-hash attribute except that its inclusion in SIP Invite message as an attribute indicates that the inviter needs a reply in the 200 OK message of that request if the other end devices support MZRTP, meaning if the responder is capable of making TLS connection and willing to make secure communication. The mzrtp-hash attribute definition is the following:

a="mzrtp-hash" zrtp-version zrtp-hello-hash

After composing ZRTP hello message-hash one must convert the hash into base 64 encoded format and include base 64 encoded hash hello message in the above form to each media attributes of SDP. If the devices are capable of establishing both audio and video call, then separate hello hash must be prepared for each and appended in the SDP part of the Invite request separately.

A sample example of a particular mzrtp-hash attribute looks like the following:

```
a=mzrtp-hash 1.00
FoAHEhlbGxvICAgMS4xMFBXYXZlSVByaXZhdGVHU01FyvyBH426w6RelxnEi6mVp7
sRAu4PDyqQrCrhXMLgSto1+TJH8DeDs8XuBQABIRFTMzg0QUVTMUFFUzNIUzMyRUM
zOEIyNTYcPS/QGl6SFQ
```

The newlines on the example are just for the formatting only which means the whole line is a single line on actual implementation.

The device receiving an Invite request(i.e. the callee) having an mzrtp-hash attribute in its message **should** understand that his caller peer wants to have an encrypted conversation(i.e. SRTP session rather than RTP session) with the aid of MZRTP key establishment mechanism and hence in its reply the callee device **must** include his own hello message in the 200 OK response of the invite if and only if the callee is willing to accept the call and his device is capable of supporting MZRTP, the device can simply ignore the mzrtp-hash message and reply accordingly. By accordingly we mean the device can either send 200 OK message in its normal way or reject the call itself.

The caller device once on receiving the 200 OK message with appropriate hello message included in it, then must reply an ACK message and **should** close the TLS connection. Here the TLS connection that we used to exchange the signaling messages is no more needed since we can use UDP to exchange the rest ZRTP messages (F5-F10) using the RTP ports and IP addresses we obtained during the signaling.

The callee device after receiving ACK then must open its RTP packet ports and finish the rest ZRTP protocol(F5-F10) on UDP transport protocol. In general after both devices exchanged their hello message(F1 &F3) on the signaling phase (that is during the Invite request and 200 Ok response) then the rest is completely implementation of ZRTP

protocol starting from commit(F5) to establishment of SRTP session keys to exchange an encrypted RTP packet(F10) as fully explained on the ZRTP spec.

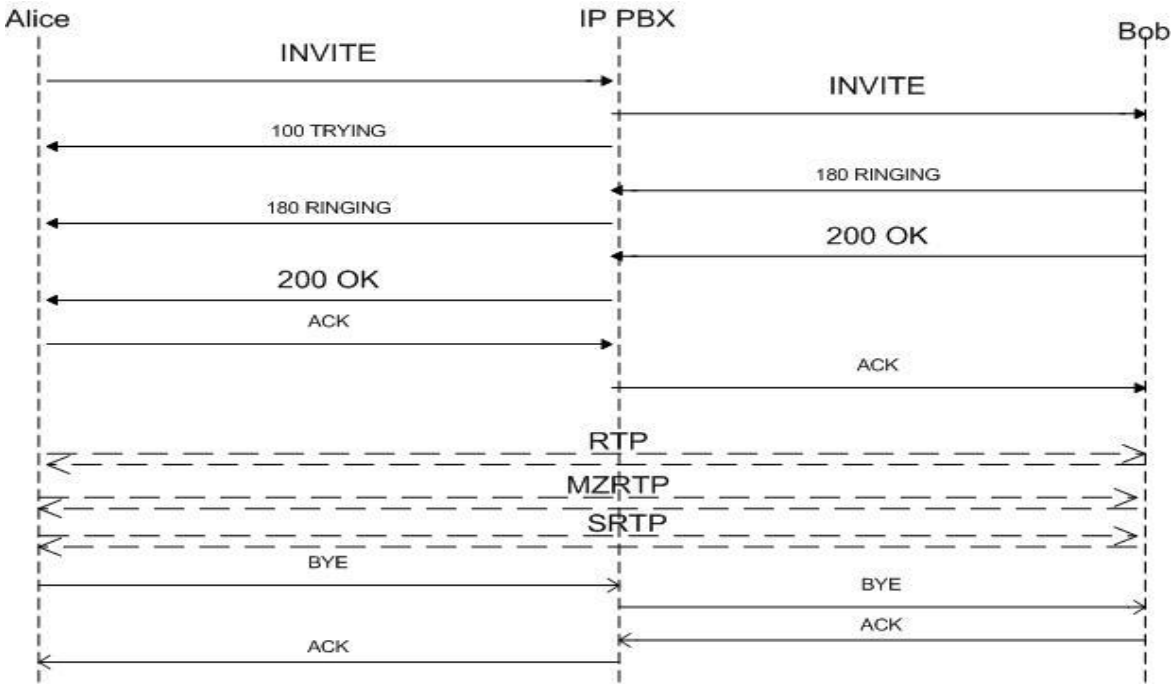


Figure 18. A Sample complete SIP based VoIP call Flow with one IP PBX

Figure 18 shows the complete procedures and call flows in one successful call setup between Alice and Bob. On the figure the hard (continues) lines are made continues for two purposes: one, they show the messages denoted by those lines are signaling messages and two, the messages are transported on reliable transport TCP with integrity and confidentiality protected channel using TLS. The dotted lines shows that the transport is UDP. One more point on the figure is that the INVITEs and the 200 OK are bolded intentionally to illustrate that the new attribute we proposed in this thesis work (i.e. mzrtp-hash) is carried by those messages. In the Invite message Alice, the caller, computes her own hello message(F1) as depicted in ZRTP spec and encode it to base 64 form and attach it as an SDP attribute mzrtp-hash definition given above. Then Bob after decoding and checking as the sent hello-message is a proper hello message then he also computes his own hello message(F3) and send it for Alice through the SDP part of 200 Ok response if he is willing to accept the call. By this procedure both Alice and Bob have exchanged their hello messages (F1 and F3) reliably and completed the signaling phase,

hence they can now close their TLS connection after exchanging ACK messages and start to exchange RTP packets using UDP.

As the RTP packets starts to flow, the devices will start to exchange the rest ZRTP messages to complete the creation of shared secret SRTP session keys and salts side by side. Actually the ZRTP specification suggest that, once the hello messages have been successfully exchanged and Commit messages exchanged the peers must halt sending plain RTP packet (unencrypted voice data) until the two devices establish shared secret keys. Once ZRTP got completed then the devices start the encrypted communication, SRTP session. ***Thus generally MZRTP is a ZRTP divided into the two phases of VoIP to reduce the rounds taken by ZRTP and hence reduce the time required to establish a shared secret keys used for making an encrypted VoIP session, SRTP while preserving all capabilities and advantage of ZRTP.***

Below is an example of messages exchanged by Alice (the caller registered at proxy or IP PBX address atlanta.example.com) and Bob(the callee registered at IP PBX or proxy server biloxi.example.com) taken from [1] and modified to include an mzrtp-hash attribute. The messages shown here are the messages that are input critical for MZRTP only, meaning only the invite request message and 200 OK response message is shown and as MZRTP spec depicts here in this thesis, the messages must be transported on reliable integrity and confidentiality protected channel, particularly TLS.

Alice's Invite message should look like the following:

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP
client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151
```

```
v=0
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=mzrtp-hash 1.00
UFoAHEhlbGxvICAgMS4xMFBXYXZlSVByaXZhdGVHU01FyvyBH426w6RelxnEi6mVp
7sRAu4PDyqQrCrhXMLgSto1+TJH8DeDs8XuBQABIRFTMzg0QUVTMUFFUzNIUzMyRU
MzOEIyNTYcPS/QGl6SFQ
```

Bobs 200 OK response message should look like the following:

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP
client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=192.0.2.101
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 147

v=0
o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com
c=IN IP4 192.0.2.201
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=mzrtp-hash 1.00
UFoAHEhlbGxvICAgMS4xMFBXYXZlSVByaXZhdGVHU03oduJq+mLPb8AHJXI7D0dLH
mANj49wgVHLetdeIsUmav4QvNlix+SdpfbsJQABIRFTMzg0QUVTMUFFUzNIUzMyRU
MzOEIyNTYDK+vY3Dy5YQ
```

In the next chapter we will describe the implementation and analysis of results for different test scenarios.

Chapter Five

5. Implementation and Result Analysis

In this chapter we will describe the demo-softphone developed to illustrate the implementation of the security features proposed to guaranty the complete confidentiality and integrity protection of both signaling and conversation (media) phases of VoIP and perform quality assessment of the effects of those security measures in the performance of the service.

It is obvious that a complete and commercial VoIP system requires a great deal of care and infrastructure however, one can easily develop a softphone (a software phone)of his own and deploy it on his or her own computer or simply download other freely available softphone software like G-talk and Skype for voice chat. For our purpose however, since most of freely available softphones are either difficult to customize their codes when compared to developing one from scratch due to their complexity or no open source complete VoIP phone found, and hence we developed complete and functional softphone from scratch although some free API's are used to implement our proposed key establishment mechanism, MZRTP, successfully.

The softphone is only one half of working VoIP system since one does need other infrastructures, internet infrastructures, like proxy server, IP PBX, databases to accumulate profiles and location address of users. We solved this half of the requirement by using open source PBX software called Asterisk PBX.

Asterisk is a well-known and open source IP PBX system which provides all the features offered by a classic PBX system such as making simple telephone calls, call transferring, voice mail, conference calling and voice menu etc. In addition to these features, more advance features can be created and integrated by defining new Asterisk dial plan scripts. Asterisk is black-box VoIP solution which bundles all the necessary VoIP protocols including SIP, MGCP, H.323, IAX etc., and it is supported on a variety of operating systems including OpenBSD, MAC OS, Free BSD and Microsoft Windows. Inter-Asterisk Exchange(IAX) protocol provides trunking of calls among multiple Asterisk

PBX systems. Asterisk can be used as soft PBX system in homes, offices, call centers, and by VoIP service providers.

5.1.The Softphone Application

The developed application has two basic functional modules in which each module has their own lists of sub modules. These are Signaling module where protocols needed for signaling, SIP and SDP, as well as TLS transport layer security protocol used for securing the signaling phases are implemented and the other module is a Media module where the protocols needed at the conversation phase ,RTP and SRTP, are basically implemented. However, a third module sandwiched between the other two module as a bridge or connecter is a module where we implemented the proposed key exchange mechanism MZRTP.

The softphone is developed using java software programming language. An open source API called JAIN SIP and JAIN SDP , particularly jain-sip-1.2 and jain-sdp-1.2 is used to implement the signaling protocols, SIP and SDP. One good reason for using JAIN API's is that the API supports all the transport protocol UDP, TCP and TLS hence avoids the complex implementation of TLS from scratch.

It is obvious that TLS requires a digital certificate signed by a well known and trusted certificate authority(CA). Each UA, VoIP phone device or the softphone, must have chain of certificate signed by the trusted CA. The TLS support in our demo softphone is made possible by creating a self signed certificate chain using OpenSSL software and Java built in tool called **keytool**. The procedure for generating the TLS certificate is presented in appendix B. In a commercial environment where there could be huge number of users, self signed certificates shouldn't be used since authenticating the certificates chain or distributing a certificate for each device is a difficult task or may be impossible instead certificates provided by trusted third body certificates authority or government certificate provider should be used. For our case this is not a problem since we are testing the application with limited users (actually two users, Alice and Bob plus

the Asterisk server is involved in the certificate chain creation) or softphones only, in which creating chain of trust of the certificate is simple issue.

In order to implement the RTP protocol a library called **jlibrtp** [10] is used. The library jlibrtp was started as a term project in VoIP Security, a class taught by Prof. Henning Schulzrinne at Columbia University was written by Vaishnav Janardhan and Arne Kepp. The version used by this thesis work is the one rewritten by Arne Kepp, as a student project under the supervision of Prof. Henning Schulzrinne, Columbia University.

The rest is implementation of MZRTP. As we have explained thoroughly in previous sections MZRTP is another flavor of ZRTP that tightly cooperate with SIP and TLS in order to reduce the time taken to establish secure call setup to guaranty confidentiality and integrity of both the signaling and media layer or phase of VoIP by using TLS as confidentiality and integrity guarantor of signaling and SRTP as media phase guarantor of confidentiality and integrity of the service. So implementing ZRTP and splitting into two half between SIP (F1-F4 of ZRTP i.e. discovery and protocol negotiation part of ZRTP got completed during the signaling protected by TLS)and RTP(F5-F10 of ZRTP i.e. the actual key establishment steps of ZRTP got completed as soon as the SIP ends and RTP begins) is what it takes to implement MZRTP. To do this easily we searched for pre-implemented open source code for ZRTP. We have found the one implemented in java and freely available for modification or usage as it is. The ZRTP implementation is called **Zorg.**

Zorg is an implementation of the ZRTP protocol. Coupled with an SRTP implementation, Zorg provides VoIP security with Diffie-Hellman (up to 3072 bits) or elliptic curve Diffie-Hellman (up to 384 bits) for key exchange, AES (up to 256 bits) for confidentiality and HMAC-SHA1 for authentication. Zorg implements all mandatory features of ZRTP, plus key continuity and all optional Diffie-Hellman key agreement types. The project is sponsored by PrivateWave, an European (Italy) company that uses ZRTP security in its PrivateGSM voice encryption product for Blackberry, Nokia and iPhone following an open source and transparent security approach[11],[13].

Thus after obtaining the source code for ZRTP we made a slight modification to turn it into MZRTP. Particularly, we made the discovery and protocol negotiation part of ZRTP (i.e. F1-F4, see figure 16 to understand the F's)to begin and end with SIP invite request than its pervious way of starting after SIP concluded with RTP as we have explained the procedure of MZRTP in chapter 4. Doing this doesn't reduce the security strength of ZRTP according to our belief because: one, the part that got moved to signaling layer out of the ZRTP is transported over other strong security protocol, TLS, two, even if TLS got breached and the integrity protection of TLS fails ,hence the signaling also transported insecurely, ZRTP still remains secure by its key continuity and SAS authentication feature. These are the features that made ZRTP protocol strong and fairly reliable as key exchange mechanism than the other key exchange methods. Generally the softphone we developed has the interface illustrated in figure 18.

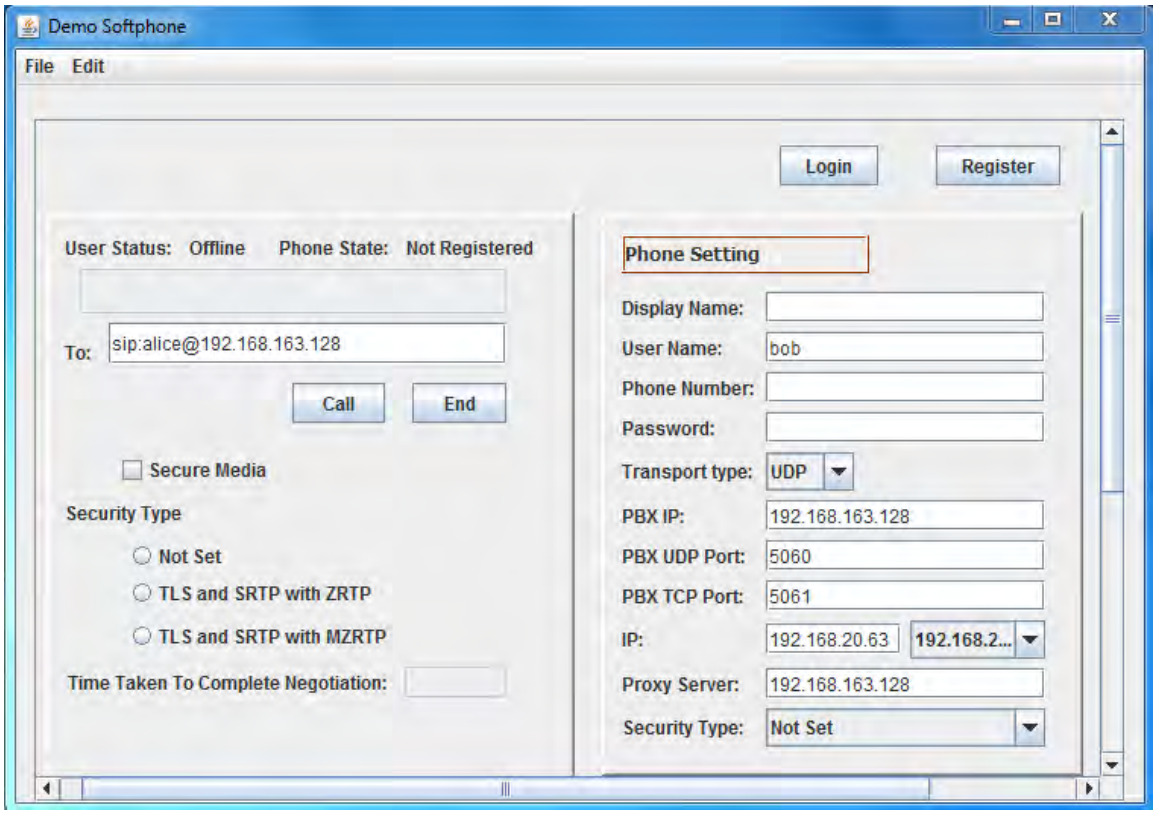


Figure 19. Snapshot of the Developed demo Softphone

As seen from figure 19 the softphone can be used in three ways regarding its security features.

- ✓ A scenario used without any security features to guaranty confidentiality and Integrity of both phases.
- ✓ A scenario where the softphone is used using TLS as confidentiality and integrity protector of the signaling phase and SRTP protocol used to guaranty confidentiality and integrity of media phase using ZRTP as a key exchange mechanism.
- ✓ A scenario where the softphone is used using TLS as confidentiality and integrity protector of the signaling phase and SRTP protocol used to guaranty confidentiality and integrity of media phase using MZRTP as a key exchange mechanism. This is the scenario proposed by this thesis work.

These are the three test benchmarks or scenarios we used to test the reliability and dependability of our proposed system for VoIP technology security. We will explain the reason for this in detail in the 5.4.2 section.

5.2.The Asterisk PBX server

Asterisk is an open source framework for building communications applications. Asterisk turns an ordinary computer into a communications server. Asterisk powers IP PBX systems, VoIP gateways, conference servers and more. It is used by small businesses, large businesses, call centers, carriers and governments worldwide. It is free and open source, sponsored by Digium, the Asterisk Company. Asterisk is “under the hood” in countless voice communications applications and is capable of interfacing with many traditional telecom protocols, VoIP protocols, and codecs. Asterisk provides a staggering list of capabilities an features including: IVR, ACD, Audio and Video Conferencing, Voicemail, Call Recording, Fax termination and CDR.

In order to establish a SIP call setup one does need to edit at least two files of Asterisk, sip.conf and extension.conf, usually located in the directory /etc/asterisk/. In the sip.conf we configure the settings of devices of our customers, meaning the UA's like the softphone or IP hardware phones, where as in the extension.conf we configure the dial plans of the phones configured in the sip.conf. After all we reload the newly configured sip files by asterisk command line command "sip reload" so that when incoming calls

arrive from devices registered in the phone and currently available online asterisk looks if the call is made according to the settings in sip.conf and if okay it looks for the called peer in extension.conf when peer got found asterisk forward the call to the peer according to the description of properties of the device found in its own sip.conf context.

For our asterisk setup for the test demo softphone we installed latest and stable Asterisk PBX 11.1.2, on virtual machine with Ubuntu desktop 12.04. Thus accordingly our demonstration has two phones in its asterisk sip.conf and extension.conf, the minimum number phones required to make a test. Below we have presented the minimum basic asterisk configuration for our two phones, Alice and Bob's phone.

Table 6: Asterisk Configuration for SIP call setup

Alice's device Asterisk configuration	Bob's device Asterisk configuration
sip.conf	
[general] tlsenable=yes tlsbindaddr=0.0.0.0 tlscertfile=/etc/asterisk/keys/asterisk.pem tlscafile=/etc/asterisk/keys/ca.crt tlscipher=ALL tlsclientmethod=tlsv1 ; Note that asterisk.pem and ca.crt are TLS files we generated for TLS support of our demo softphone as shown in Appendix C.	
[alice] type=peer secret=alicepsw ;note that this is NOT a secure password host=dynamic context=local dtmfmode=rfc2833 disallow=all allow=g722 transport=tls context=local	[bob] type=peer secret=bobpsw ;note that this is NOT a secure password host=dynamic context=local dtmfmode=rfc2833 disallow=all allow=g722 transport=tls context=local

extension.conf
[peer] exten=>alice,1,Dial(SIP/alice,20) exten=>bob,1,Dial(SIP/bob,20)

Extension is simply the phone number of device and it can be a number only or text only or combination of number and text. Here we used texts to name extensions like 'alice' and 'bob'. One more thing that someone has to bear is that the value of the transport attribute in the sip.conf file. The value of transport attribute is 'tls' which in this case means alice and bob both support TLS, which is basically an encrypted TCP. If one wants to support the other two protocols too, udp and tcp, all he/she has to do is add them in by comma separation. Generally the above setting is a very basic working setting of asterisk for SIP devices and more sophisticated tasks has to be done when the service is commercial or out for public use.

5.3.System Flow Chart

The developed softphone is a complete working Java desktop application developed to meet basic SIP headers. The general flow chart of the system is shown in Appendix A. Mainly the flow chart simply demonstrates the registration phase and Invite phase caller and receiver phone.

5.4.System Test

5.4.1.Test Environment Setup

The environment we used to implement the softphone and the VoIP service with the newly proposed key establishment mechanism, MZRTP, is composed of three computers connected with a LAN network on virtual machines. The first two computers are used for deploying the softphones developed as presented above while the other computer is used as a PBX server by deploying asterisk on it. The virtual machine used for the networking purpose is VMware version 9.1 and the asterisk server 11.1.2 is installed up on Ubuntu 12.04 machine on VMware virtual machine. The softphones are deployed on: host

computer with operating system of windows 7, 64 bit 6GB RAM and on virtual machine installed on the host machine with an OS of Windows XP, 32 bit. The architecture of the interconnection of the machines is shown on figure 20 below.

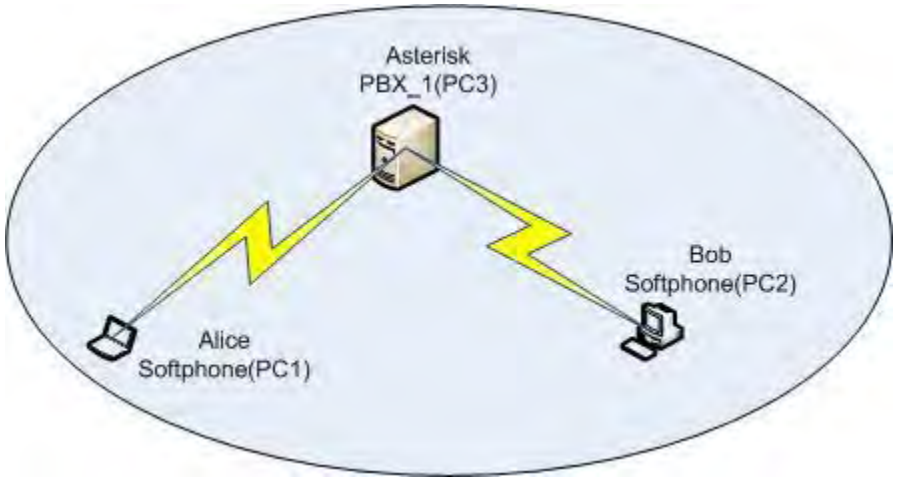


Figure 20. LAN network composed three computers used for the test of the implemented proposed for complete confidentiality and integrity protection of VoIP in performance reliable manner.

5.4.2.Test Scenarios

As explained sorely in the previous chapters the intention of this thesis is guarantying confidentiality and integrity of VoIP service completely (meaning securing both signaling and media phase) by using the widely implemented network security protocol TLS to carry SIP messages and SRTP protocol for security of actual voice packet with an aid of a newly proposed key exchange mechanism MZRTP short for Modified ZRTP protocol. In the description of the design of implementation of MZRTP in chapter four we have noted that as *MZRTP is a linking protocol rather than a full key exchange or establishment protocol like SDES or ZRTP, and* it is used for associating the separately dealt phases of VoIP telephone service so that the two best security protocols TLS and SRTP suggested by many scholars, as explained in chapter 3, cooperate to yield a secure and reliable service in short secure session establishment period.

The confidentiality and integrity of the session is guaranteed by the two protocols TLS and SRTP operating in the two phases and this is not required to be tested since there are many researches done to demonstrate the strength of these protocols prior to this thesis as

seen in chapter 3 and this thesis is done by considering those research as cornerstone. Hence what we test in our work is primarily measuring the performance we manage to increase by employing MZRTP as a key exchange mechanism. To do this analysis we have made three comparative scenarios listed in section 5.2.1. For simplicity we shorten the representation of the scenarios as SC1, SC2, and SC3 based on their sequence of list in the section 5.2.1.

The scenarios we picked are intended for comparing the time required to establish a secure session using our proposed system or mechanism against the insecure (i.e. plain exchange of data, or default VoIP setup) and session secured by TLS and ZRTP in combination. The formula to calculate the time taken in each scenario can be simply the addition of time taken to complete a session negotiation (SIP part, let it be represented by SIPt) and time taken to start SRTP encrypted media or conversation phase(shortened as SRTPt). Let us shorten the times as SC_1, SC_2 and SC_3 to denote the time taken by scenario 0ne, two and three.

SIPt is the time measured by calculating the time taken to generate Invite request by the caller to the time the caller received 200 OK response from the callee. In our proposed system the invite request is not the ordinary invite only since it is also contains an MZRTP hello message included in it as an SDP attribute with an attribute labeled by mzrtp-hash. So SIPt time in our proposed system includes the time taken to compose the MZRTP hello message. We simplified the SIP call setup transactions by avoiding the digest authentication which is a highly recommended security feature by SIP RFC 3261. In SIP there is two digest authentication one used for authenticating the proxy that we are directly communicating with and this authentication is labeled in SIP header message by Proxy-Authentication while the second kind of authentication is an authentication made to authenticate the client itself by the server, labeled by WWW-Authenticate header in SIP. The restrictions we made here on the test are not involving proxy authentication or PBX server authentication (i.e. user authentication by the service provider or the PBX server) to both the caller and callee device. By avoiding this two authentication types we used two-way or mutual authentication feature of TLS, but still it is recommended to implement digest authentication on a commercial device since the TLS simply identifies

the devices not the users but by digest authentication we can authenticate the proxy that we are talking to is the proxy that we subscribed for by using Proxy-Authentication and the server or the proxy can also authenticate us by checking if we are who we said on the invite message.

SRTPt is the time taken to complete the startup of SRTP protocol on each devices. Well for the first scenario this should be almost automatic since no security protocol is executed at the media layer except just opening the ports of the RTP protocol for exchange of RTP packet whereas in the other two cases the time taken to complete ZRTP and the proposed MZRTP is a calculable difference. We implemented the plain RTP with no security to compare the amount difference in time it takes for the other two secure call setups. Just by observation only, one can say that MZRTP takes a reduced time since MZRTP remove the first 4 message exchange of ZRTP by completing them during the signaling phase. However, this might not be the case since the message size encrypted by TLS during the ZRTP and MZRTP are different then the time taken to complete those encryption might also vary hence bringing change on the total secure call setup time. That is what we are going to test in our test scenarios.

5.4.3. Test Results

As explained the complete session negotiation (the time between the calling and the picking or hearing of the first hello sound) is calculated by the formula:

SC = SIPt + SRPTt

We believe the computers spec we used for the test will not have an effect on the outcome of the result even though one might guess that a different result might be calculated by using computers with a different spec. Our reason to falsify this guess is that:

> *We calculated the time on the same computer with the same spec for each scenario which means that the result that might be obtained if the test was made on computer with a different spec than that we used, one will also obtain similar result at the end since we compare* ***the average of the differences between completion of each scenario****. For example assume that one calculated the test on a computer with a 6GB RAM and the other on*

4GB RAM. The session might end quickly on the 6GB RAM say 0.5 second and slower on the 4GB RAM say 1 second using SC_3 mechanism. And the same call was made using SC2 and assume that one obtained the following results for the two computer spec: 0.8 second on 6GB RAM and 1.3 second on 4GB RAM computer however. So in above example the average difference on both computer spec is 0.8-0.5= 0.3 second for the 6GB RAM and 1.3-1= 0.3 second for 4GB RAM which shows that finally one will start his conversation 0.3 second earlier than the other if he uses SC_3 on any computer type. On the above example we didn't show any calculation of average because the test was made ones and dividing by one yields the same results.

Based on the above assumption we have run our demo softphone ten times and obtained the results shown in table 7. We have no reason for deciding the test to be performed ten times except that we believe that it is enough since the results we are obtaining throughout the test does not deviate from each other in considerable magnitude.

Table 7. Result of Test of each scenarios after testing them ten times.

Test Numbers	**Time taken by Scenario one (SC_1) in millisecond**	**Time taken by Scenario Two (SC_2) in millisecond**	**Time taken by Scenario three (SC_3) in millisecond**
1	2466	10330	6960
2	2622	10953	7887
3	2501	10646	8743
4	2514	10417	8547
5	2488	10213	8643
6	2521	10482	7582
7	2464	10037	7537
8	2457	10182	7812
9	2415	11086	8216
10	2424	10816	7898
Average	2487.2(2.5sec)	10516.2(10.5sec)	7982.5(8.0sec)

As expected the result of the test shows that it takes at an average 8.00 seconds for VoIP call to establish a secure communication session between two peers using a combination of TLS and SRTP by using MZRTP as a key exchange mechanism and 10.5 second at an average by the same security mechanism but ZRTP as a key exchange mechanism which means MZRTP will let us complete the complete the call setup in 2.5 second earlier than its equivalent ZRTP. With this if lots of devices are to establish a voice communication at the same time on one asterisk PBX server using MZRTP as a key exchange the traffic congestion that might occur at the start of the call will be reduced and more phones will establish secure session (in context of confidentiality and integrity) successfully than the one established using ordinary ZRTP.

Chapter Six

6. Conclusion And Future Work

6.1. Conclusion

Voice over Internet Protocol is a rapidly growing Internet service due to the demand of cutting costs of international telephone connections by transporting voice over public IP network. Actually cost reduction was not the only demand as there are also additional demands for consolidating multimedia services like voice call, text chat, video conferences, automated voice replays, extra in just one application and hence creating a unified communication system. These demands were driving forces for fast growing Voice over IP services.

Obviously telephony is an essential technology for betterment of human's daily life. This service needs to be secured to be relayed on for which the customers discuss their confidential issues or private matter in privacy assured way without letting access by an intended third body. This is the biggest challenge in VoIP where as relatively this was not a problem for the traditional telephone networks since physical security is tight on the infrastructures. Now a days as there are many researches done and being done on how to secure the service, however there is a wide understanding by the service providers that the security features to be incorporated in their service will highly reduce the quality of their service. Hence they tend to sacrifice security as a pay for quality of service which we considered as inappropriate. Or some vendors or VoIP applications in the market or freely available once simply encrypt the conversation phase while leaving the signaling phase vulnerable which by itself can lead to serious threat to the service. Thus we proposed a technique of securing VoIP completely , the signaling and the media phases all at once, using current best internet security protocols and recommended features in the individual protocol documents so that we can provide a secure and performance enhanced service.

The proposed system employs one of well known IETF standard and strong security protocol called transport layer security(TLS) for securing the signaling messages or data hence providing confidentiality and integrity protection alongside the device

authentication capability of TLS. Upon completion of signaling successfully the next is the conversation phase that needs to be mainly secured to attain privacy, confidentiality and integrity protection, that is encryption of the voice data. We used SRTP again the famous protocol for securing real time data. SRTP is an encryption protocol that does not generate an encryption key by itself but need a key negotiation phase to take prior to its start so that it can use the negotiated keys for encryption and hence yielding confidentiality, integrity, authentication and replay attack protection. That is where we proposed a new technique of key negotiation mechanism by modifying one of well acknowledged and strong shared key establishment method of VoIP.

The modified and newly proposed key establishment mechanism we suggested in this thesis work is called MZRTP short for Modified ZRTP. ZRTP is strong protocol that can thwart many attacks all by itself, like man in the middle attack, to successfully establish a shared key between two entities. Thus we didn't modify the basics of the protocol so as to increase its security strength instead we modified its working principles so that it can be linked to signaling layer protocols, specifically to SIP, and hence the media layer (conversation phase) protocol ZRTP will actually start during the signaling and ends at the beginning of media layer before any unencrypted conversation begins. This reduced complexity of ZRTP and reduced the time needed to start SRTP, the encrypted conversation session.

We have implemented and tested the proposed system for establishing secure call setup that can successfully thwart confidentiality and integrity attacks of VoIP service in QoS reliable fashion. By the test we investigated the contribution of the proposed technique, MZRTP, over the old ZRTP technique at improving the time needed to establish secure conversation session. Accordingly we obtained at an average 2.5 second saves per calls when modified ZRTP is used than the well known ZRTP is used. In percentage that means MZRTP takes 24% less seconds than ZRTP takes to establish the encrypted conversation session.

6.2.Future Work

Our aim in this thesis was to guaranty confidentiality and integrity protection throughout the call setup to the end of conversation while maintaining overhead and call setup time to the minimum possible. And for that we used TLS to secure the signaling and SRTP to encrypt and secure the actual conversation with an aid of newly crafted shared key establishment technique, MZRTP. That worked successfully and improved the call setup time as well.

Here we relayed on the security strength of TLS to secure the signaling layer of call setup. However, TLS is a hop by hop security protocol and doesn't guaranty end to end encryption or security. If one, unintended intruder, manage to be one of the intermediate hop then it is obvious that he can manipulate the signaling messages and hence next TLS hops doesn't know whether message were manipulated since they don't have anything to compare with. To overcome this challenge SIP RFC 3261 suggests that S/MIME encryption or signature be used with TLS. The RFC says that at least S/MIME protecting the To header of the SIP Invite message will be enough to prevent call redirection or hijacking and downgrade attacks like converting the SIPS (i.e. SIP message protected by TLS)request to SIP request and thus avoiding TLS protection. This is done by repeating the To header of the request in the SDP part of the SIPS invite request and then protecting the SDP part by S/MIME. When we do this if someone changes the SIPS Request-URI to SIP Request-URI then it can be easily detected by the recipient of the message from the SDP part. However for privacy purpose developers tend to make the Request-URI and the To header different in the first place and hence their difference necessarily doesn't mean security breach. Hence adding both the Request-URI and To header repeatedly in the SDP part and protecting that SDP with S/MIME is a good choice. Even though S/MIME has great advantage in this way however it doesn't come without challenge like, need to establish pre-shared keys between peers and overhead due to increased message size.

Here our future work is thus adding an extra security future S/MIME to have a real end-to-end integrity protection rather than relying on TLS only. By doing this we can be sure

that even if TLS got breached still we have integrity protection to our MZRTP hash hence we are sure that the media is secure though the signaling got breached. As a precaution however, continuing a call setup even after knowing security breach in the signaling is not safe, so the device must notify the user about the possible security breach before going on the signaling. Figure 21 shows sample of invite request with S/MIME security being used for extra integrity request as suggested by SIP RFC with our mzrtp-hash.

SIP invite Request headers

```
INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIPS/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sips:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sips:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sips:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
```

S/MIME protected SDP part of SIP invite request

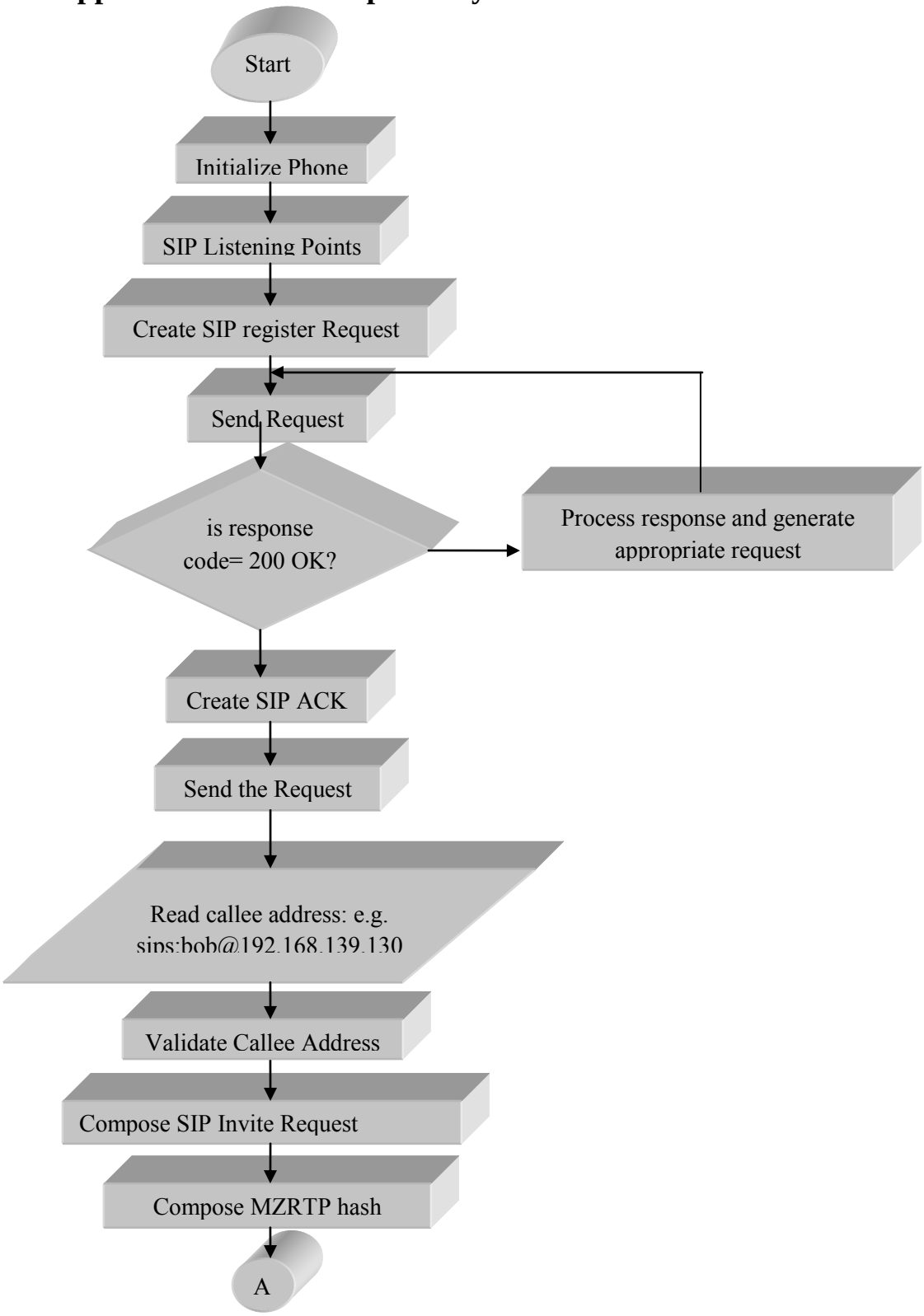
```
INVITE sips:bob@biloxi.example.com SIP/2.0
To: Bob <sips:bob@biloxi.example.com>
Content-Disposition: attachment; filename=smime.p7m
handling=required
Content-Type: application/sdp
v=0
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=mzrtp-hash 1.00 zrtp-hello-hash
```

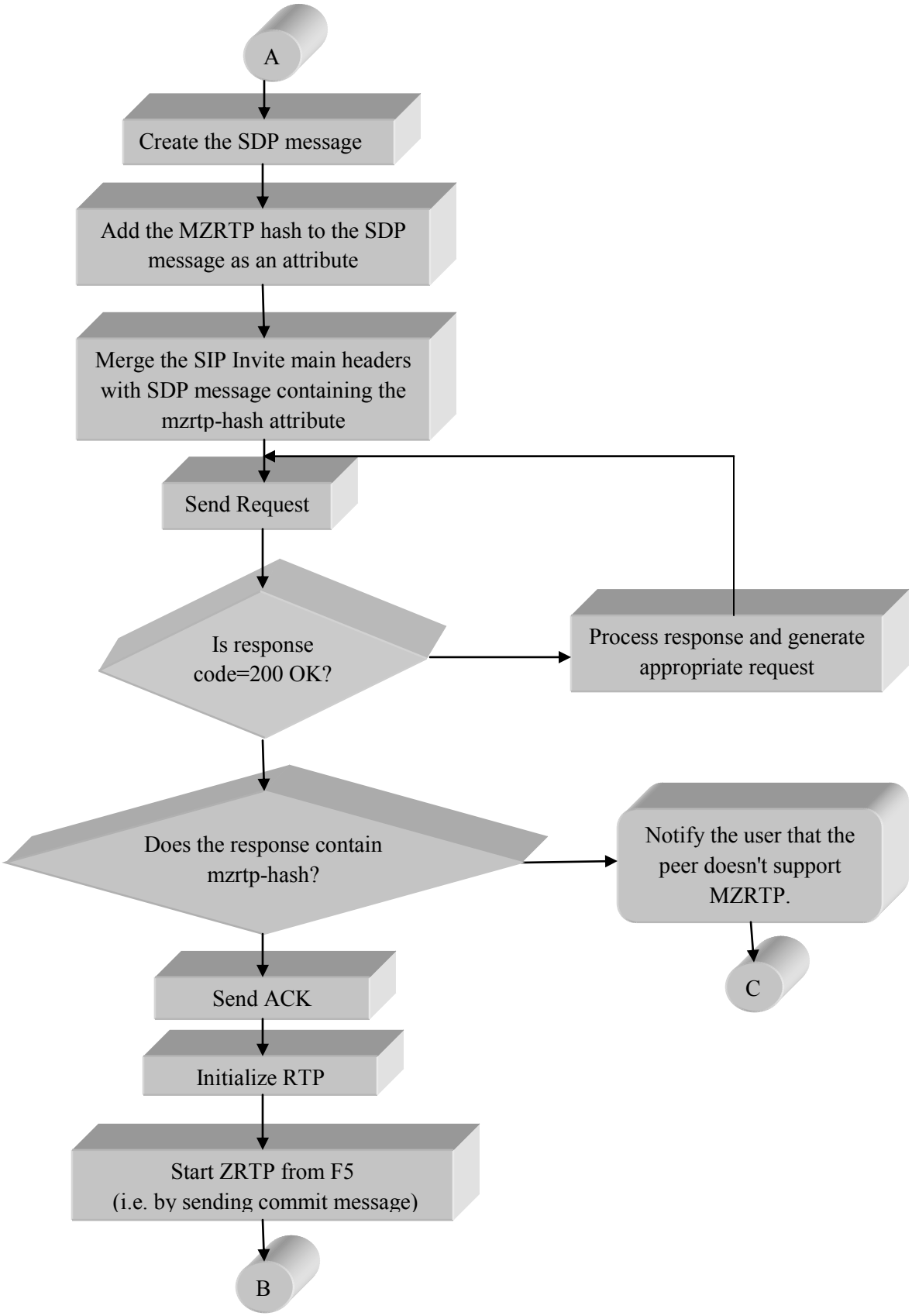
Figure 21: S/MIME protected SIP invite request.(Actually it is the SDP part that is protected by S/MIME. Look that the Request-URI and the To header of the request is also repeated in the SDP to detect if the unprotected part is edited on its path to the recipient for downgrading attack.)

Our other future work is to research on how to guaranty or at least increase availability protection to VoIP service so that the three basic information securities confidentiality, integrity and availability be assured and migration from traditional telephone network to VoIP is facilitated for the sake of satisfying the demands we mentioned in chapter one.

Appendix

Appendix A. Demo Softphone System Flow Chart




A
Create the SDP message
Add the MZRTP hash to the SDP message as an attribute
Merge the SIP Invite main headers with SDP message containing the mzrtp-hash attribute
Send Request
Is response code=200 OK?
Process response and generate appropriate request
Does the response contain mzrtp-hash?
Notify the user that the peer doesn't support MZRTP.
C
Send ACK
Initialize RTP
Start ZRTP from F5 (i.e. by sending commit message)
B

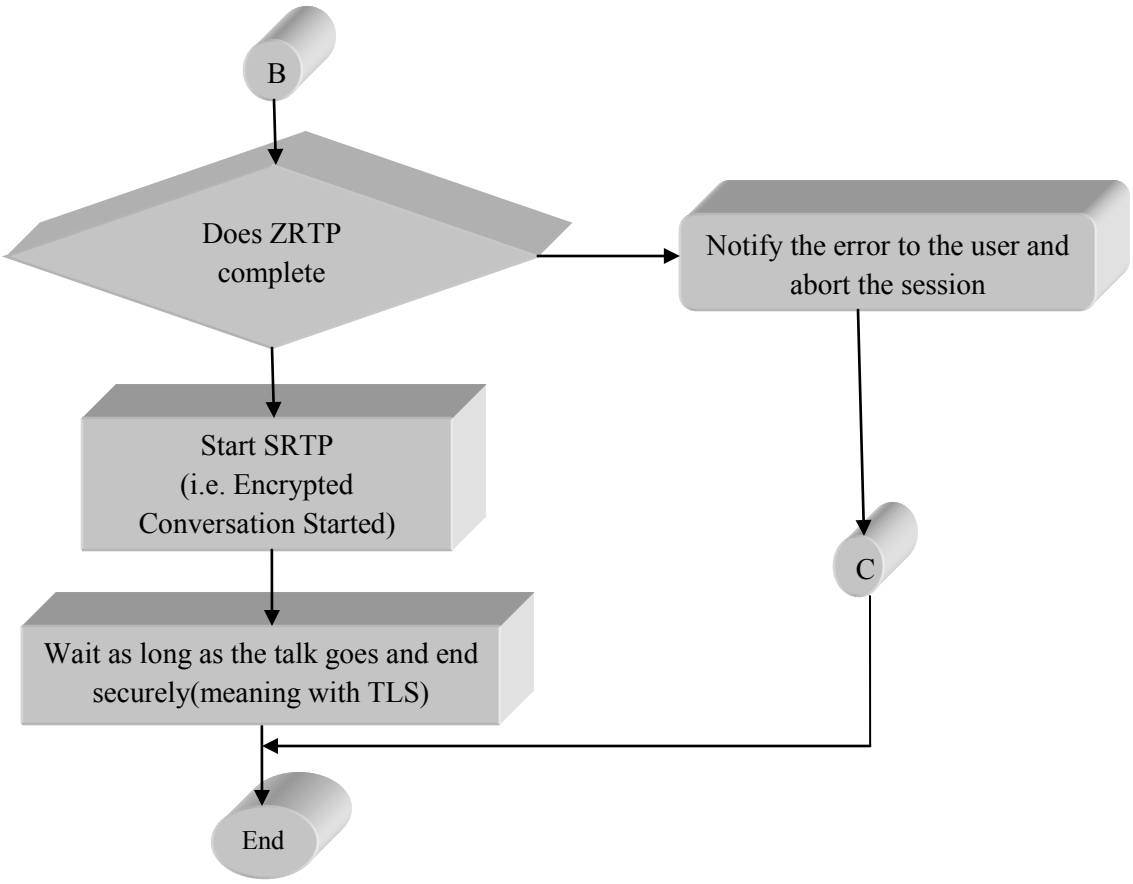


Figure 22: System Flow Chart

Appendix B. TLS certificate generation and binding to Asterisk PBX server

Transport Layer Security (TLS) provides encryption for call signaling. It's a practical way to prevent people who aren't Asterisk from knowing who you're calling. Setting up TLS between Asterisk and a SIP client involves creating key files, modifying Asterisk's SIP configuration to enable TLS, creating a SIP peer that's capable of TLS, and enabling the SIP client to connect to Asterisk over TLS. There are a lots of ways of doing this of which we used the script found in the Asterisk server software itself to generate the keys for us. The steps are shown below:

Keys

First, let's make a place for our keys.

```
mkdir /etc/asterisk/keys
```

Next, use the "ast_tls_cert" script in the "contrib/scripts" Asterisk source directory to make a self-signed certificate authority and an Asterisk certificate.

```
./ast_tls_cert -C 192.168.139.130 -O "My Company" -d /etc/asterisk/keys
```

- The "-C" option is used to define our host DNS name or our IP address. Here we used the IP address of our Asterisk server.
- The "-O" option defines our organizational name.
- The "-d" option is the output directory of the keys.

1. You'll be asked to enter a pass phrase for /etc/asterisk/keys/ca.key, put in something that you'll remember for later.
2. This will create the /etc/asterisk/keys/ca.crt file.
3. You'll be asked to enter the pass phrase again, and then the /etc/asterisk/keys/asterisk.key file will be created.

4. The /etc/asterisk/keys/asterisk.crt file will be automatically generated.
5. You'll be asked to enter the pass phrase a third time, and the /etc/asterisk/keys/asterisk.pem will be created, a combination of the asterisk.key and asterisk.crt files.

Next, we generate a client certificate for our SIP device, let it be for Alice and IP address of Alice's softphone is 192.168.39.129. The following step will be repeated for all clients similarly, in our case for Bob.

./ast_tls_cert -m client -c /etc/asterisk/keys/ca.crt -k /etc/asterisk/keys/ca.key -C 192.168.139.129 -O "My Company" -d /etc/asterisk/keys/ -o alice.pem

- The "-m client" option tells the script that we want a client certificate, not a server certificate.
- The "-c /etc/asterisk/keys/ca.crt" option specifies which Certificate Authority (ourselves) that we're using.
- The "-k /etc/asterisk/keys/ca.key" provides the key for the above-defined Certificate Authority.
- The "-C" option, since we're defining a client this time, is used to define the hostname or IP address of our SIP phone .
- The "-O" option defines our organizational name.
- The "-d" option is the output directory of the keys.
- The "-o" option is the name of the key we're outputting.

1. You'll be asked to enter the pass phrase from before to unlock /etc/asterisk/keys/ca.key.

Now, let's check the keys directory to see if all of the files we've built are there. We should have:

asterisk.crt	alice.key
asterisk.csr	alice.pem
asterisk.key	ca.cfg
asterisk.pem	ca.crt
alice.crt	ca.key
alice.csr	tmp.cf

Next, we copy the alice.pem and ca.crt files to the computer running the demo softphone.

The Asterisk SIP configuration

Now, let's configure Asterisk to use TLS.

In the **sip.conf** configuration file, set the following:

```
tlsenable=yes
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/keys/asterisk.pem
tlscafile=/etc/asterisk/keys/ca.crt
tlscipher=ALL
tlsclientmethod=tlsv1
```

Here, we're enabling TLS support. We're binding it to our local IPv4 wildcard (the port defaults to 5061 for TLS). We've set the TLS certificate file to the one we created above. We've set the Certificate Authority to the one we created above. TLS Ciphers have been set to ALL, since it's the most permissive. And we've set the TLS client method to TLSv1, since that's the preferred one for RFCs and for most clients.

Configuring a TLS-enabled SIP peer within Asterisk

Next, we'll need to configure a SIP peer within Asterisk to use TLS as a transport type. Here's an example:

```
[alice]
type=peer
secret=alicePassword ;note that this is NOT a secure password
host=dynamic
context=local
dtmfmode=rfc2833
disallow=all
allow=g722
transport=tls
context=local
```

Notice the **transport** option. The Asterisk SIP channel driver supports three types: udp, tcp and tls. Since we're configuring for TLS, we'll set that. It's also possible to list several supported transport types for the peer by separating them with commas.

Reference

[1.] A. Johnston, S. Donovan, C. Cunningham, K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", RFC 3665, IETF, December 2003.

[2.] Angelos D. Keromytis, Voice over IP: Risks, Threats and Vulnerabilities, Symantec Research Labs Europe, Sophia-Antipolis, France.

[3.] Charles Shen,Erich Nahum, Henning Schulzrinne, Charles Wright, The Impact of TLS on SIP Server Performance, Columbia University.

[4.] D. Richard Kuhn, Thomas J. Walsh, Steffen Fries, Security Considerations for Voice Over IP Systems, 2005, National Institute of Standards and Technology.

[5.] Dimitris Geneiatakis, Georgios Kambourakis, Tasos Dagiuklas, Costas Lambrinoudakis and Stefanos Gritzalis, SIP Security Mechanisms: A state-of-the-art review, University of the Aegean, Karlovassi, Greece.

[6.] F. Andreasen, M. Baugher, D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, IETF, July 2006.

[7.] G.Aghila, D.Chandirasekaran, An Analysis of VoIP Secure Key Exchange Protocols against Man-in-the-Middle Attack, 2011, Pondicherry University, Puducherry,India.

[8.] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications" ,RFC 3550,IETF,July 2003.

[9.] Henning Schulzrinne, Jonathan Rosenberg, A Comparison of SIP and H.323 for Internet Telephony.

[10.] http://jlibrtp.org,(accessed February, 2013).

[11.] http://www.privatewave.com, (accessed August, 2013).

[12.] http://www.voipsa.org, (accessed August, 2013).

[13.] http://www.zorg.org, (accessed August, 2013).

[14.] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, MIKEY: Multimedia Internet KEYing, RFC 3830,IETF, August 2004.

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

[16.] Jianqiang Xin, "Security Issues and Countermeasure for VoIP", SANS institute InfoSec Reading Room, 2007.

[17.] M. Arango, A. Dugan, C. Huitema, S. Pickett, Media Gateway Control Protocol (MGCP),RFC 2705, IETF, 1999.

[18.] Mathew Green, A few thoughts on cryptographic engineering, http://blog.cryptographyengineering.com/2012/11/lets-talk-about-zrtp.html.

[19.] Michael Hall, Cisco Patches VoIP Phone Vulnerability, http://www.enterprisenetworkingplanet.com/netsecur/article.php/3507801.

[20.] P. Zimmermann, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, IETF, 2011.

[21.] Pawel Lawecki, VoIP Security in Public Networks, 2007, Alcatel-Lucent, Stuttgart Germany.

[22.] Piero Fontanini, "VoIP Security" , 2008, Gjovik University College

[23.] Provos, N., VOMIT - Voice Over Misconfigured Internet Telephones, http://vomit.xtdnet.nl/,(accessed January 2013).

[24.] PSTN vs. VoIP: Feature-by-feature comparison ; http://searchunifiedcommunications.techtarget.com/feature/PSTN-vs-VoIP-Whats-best-for-your-business, (accessed December, 2013).

[25.] Rahul Singhai, Prof. Anuridha Sahoo, VoIP Security.

[26.] Rakesh Arora, Voice over IP : Protocols and Standards, http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols.pdf.

[27.] Roberts, C., Voice over IP security, 2005, Center for critical Infrastructure Protection.

[28.] Shawn McGann, Douglas C. Sicker, An Analysis of Security Threats and tools in SIP-based VoIP systems, University of Colorado at Boulder.

[29.] Shawn Merdinger, ACT P202S VoIP wireless phone multiple undocumented ports/services, http://www.security.nnov.ru/Ldocument66.html.

[30.] SSL: Intercepted today, decrypted tomorrow, http://news.netcraft.com/archives/2013/06/25/ssl-intercepted-today-decrypted-tomorrow.html, Netcraft, (accessed June, 2013).

[31.] Swapna Maguluri, Performance Evaluation Of Sip Authentication And Tls, 2012, California State University, Long Beach.

[32.] Transport Layer Security, en.wikipedia.org/wiki/Transport_Layer_Security/.

[33.] Voice Over IP, http://en.wikipedia.org/wiki/Voice_over_IP.