Performance-cost Analysis of Secure Media-streams in VoIP Systems

Florian Berthelot

Submitted in partial fulfilment of the
requirements of Napier Edinburgh University
for the Degree of
MSc Advanced Networking

Work supervised by
Prof. Bill Buchanan

School of Computing
October 2010

AUTHORSHIP DECLARATION

I, Florian Berthelot, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date: 29 October 2010

Matriculation no: 07010875

DATA PROTECTION DECLARATION

Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

ABSTRACT

Voice communications emerged during the last century, and gained worldwide popularity. Telephony service was provided by the Public Service Telephony Network (PSTN), a dedicated and reliable network infrastructure. The Quality-of-Service (QoS) of dedicated network for voice provided 99.99% availability to user, with excellent quality of voice. The recent apparition of data networks (IP networks) gained large popularity during the past 30 years, with the development of Internet and broadband connections. Voice over IP (VoIP) became rapidly a cost-effective solution and gained popularity. However, IP networks uses shared infrastructure, were data and real-time applications are sharing the same medium. Mechanisms of priority allow VoIP to provide the best QoS as possible. Although VoIP presents good QoS performance, the security aspect is a key-issue in VoIP systems.

The standard secure solutions have thus been adapted to secure VoIP, but these affects dramatically the QoS performances. The security in VoIP systems against the performance cost is still a key issue. This dissertation has for aim to evaluate the recent techniques used to secure VoIP, and compare them to existing standardised solutions, like tunnelling. It evaluates the QoS cost and resource usage of both standardized protocols and recent protocols under development. Recent protocols, such as Inter-Asterisk and Secure Real-Time Protocol, are not defined as standard yet and only recent software have implemented them. It is believed that new protocols are more adapted to secure VoIP than standards, like IPSec.

A novel experimental test-bed was designed to allow the measurement of VoIP QoS and resource usage of VoIP servers. The framework allows the evaluation of different protocols, such as SIP, IAX2, SRTP over TLS, ZRTP and IPSec, in a real-scenario. It was designed to measure VoIP QoS when network condition is optimal. A wide range of concurrent calls allowed to perform measures and simulate different scenarios. The tests performed on the framework allowed to prove that securing real-time application has a performance cost, at the QoS and resource level. It was proven that recent protocols, developed to secure VoIP have a minimal impact on the QoS, but needs more resource requirements, although the overhead in less than 20% in CPU utilization.

Secure IAX2 presented significant improvement performance when securing VoIP calls, using the same security mechanisms with other protocols. This emerging protocols thus optimise some of the voice parameter, like jitter, to provide even better performance than standard voice trunk. However, SIAX requires more CPU usage and memory use than other protocols, for the same encryption mechanisms. Unfortunately, the SRTP and ZRTP scenarios could not be implemented due to their early development and the lack of documentation for their implementation.

TABLE OF CONTENTS

INDEX OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

The recent expansion of Internet technologies allows real-time applications to be carried on the same infrastructure as Internet applications. As packet networks are nowadays very reliable, they have been adapted to provide a good quality of service for real-time application. Transportation of voice using IP networks is an interesting alternative to traditional public telephony network (PSTN). The utilisation of a single converged infrastructure to transport both data and real-time applications is cost-effective (Carlos, 2009), and provides a better flexibility. New services, like voice-mail and address-book lookup, can be performed thanks to applications running in parallel of the telephony service.

Despite of numerous advantages to switch to VoIP, availability and security still remain an issue (Benini, 2008). The PSTN was build on a dedicated architecture (Min, 2004), providing 99.99% availability and excellent quality of service to users. In opposition, the transmission of voice on the data network is performed on a shared infrastructure, opened to anyone, independently of the geographical localisation. The VoIP systems,though, are vulnerable to many threats (Albers, 2005), like deny of service attacks or eavesdropping.

Although Internet applications can be secured, transmission of real-time data across the Internet requires more attention than standard Internet applications. The adaptation of secure techniques have been adapted (Wieser, 2003) to mitigate VoIP security issues. However, these solutions cannot be deployed on a long-term basis, and have a cost in terms of performance (Pérez, 2006). The majors drawbacks of existing solutions are the loss of quality of service (Diab, 2008), the lack of scalability and resource usage (Ferrante, 2007). To comply to VoIP requirements, new sets of protocols have been developed during the last past years (Gupta, 2007). These are designed for real-time applications, and should be optimised to provide good security for the lowest performance cost.

1.2 Initial objectives

Some effort has been put to reduce the performance cost of secure solutions to provide the best service to users. This dissertation evaluated the performance costs of secure implementations in VoIP systems. New protocols have been developed to overcome major drawbacks of standard solutions, like IPSec or TLS, to secure voice data.

This dissertation will evaluate the needs of voice transmission on packet networks, review the main requirements and possible threats. A critical review of secure solutions has been performed, and a review of a previous study shows the problems that secure VoIP faces.

The second objective is the design of a framework to allow the evaluation of the protocols which have been reviewed in the previous time. This framework will allow the evaluation of VoIP parameters described in the previous objective, and confirm findings of previous works.

The last objective is the implementation of the designed framework, with the use of open-source software and perform tests. The tests should confirm the findings of previous studies, and allow the evaluation of recent VoIP secure protocols, that have not been fully evaluated yet.

1.3 Report layout

Chapter 1: This presents a general background and initial objectives of this dissertation. This chapter present the methodology and layout of the dissertation.

Chapter 2: General background. This presents the essential information to introduce VoIP in its context. This chapter will present how voice can be carried on packet network, and explain why VoIP cannot be treated as any standard application. A brief presentation of VoIP quality assessment will also be carried. This chapter will also present the main protocols that allow voice transmission on an IP network.

Chapter 3: Literature review, This presents the security issues and secure techniques encountered in VoIP systems. Then a critical review will present previous studies carried in the VoIP area. Then, a review of studies about server performance will be undertaken, to present a methodology for the work performed in this work.

Chapter 4: Design and methodology. This will present a framework, based on the critical evaluation of previous work. Scenarios chosen for the tests will be presented, and choice will be justified in order to retrieve the most accurate results. Methodologies to perform measurement will be presented, with respect of the result obtained by previous studies.

Chapter 5: Implementation, This presents how the tests were performed, and gives details about the tools used to perform the measurements. Problems encountered and solutions are also described in this section.

Chapter 6: Evaluation; this chapter presents the results obtained during the tests and will present the findings of this research.

Chapter 7: Conclusion. This presents a summary of this study. The findings will present the results of the work undertaken during the tests, and a critical evaluation will discuss the project guidance.

CHAPTER 2: GENERAL BACKGROUND

2.1 Introduction

In order to introduce the work carried in this dissertation, this first chapter will present a general background of VoIP systems. The first part will present how voice is converted to allow its transportation onto packet networks. Then, a presentation of voice requirements will show that VoIP is not a standard Internet service, and needs special care to provide a good quality-of-service. The third part will introduce the main protocol used to carry voice over IP network.

2.2 Converting voice to digital signal

Voice, defined by an analog signal, must first be converted into a digital signal, to be able to be processed by the networking equipments. The first step consists of sampling the analog signal at a rate at least twice more than the frequency (Nyquist theorem). Sampling allow the analog signal to be converted into binaries values, coded with 8 bits. A sample rate which is too low will result in high information loss, while a high sampling value will result in good voice quality. VoIP system, after samples the voice at 8KHz, resulting in a bandwidth utilisation of 64kbps per voice stream. Figure 1 illustrates the process of sampling an analog signal.

The second step in processing voice in VoIP system, is the compression of the data. Uncompressed voice streams (PCM) requires 64Kbps of bandwidth per call, which does not optimize bandwidth utilisation on WAN links. Compression occurs after the sampling process, using to codecs. Different algorithms are used to compress voice: Pulse Code Modulation (PCM) codes each sample, while Adaptive Differential PCM (ADPCM) code transmit only the difference between the previous sample. Other algorithms, like Code Excited Linear Predictive (CELP), compress the voice with a vector quantizer, predict the waveform using a Vector Speech Book. Codecs are defined by the ITU-T standards, and are presented in Table 1.

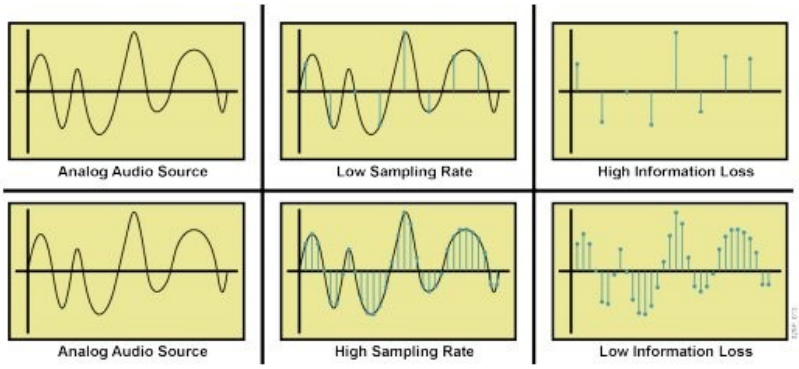


Figure 1: Sampling an analog signal (Cisco CCNP materials)

ITU-T Standard	Codec	Bit-rate (Kbps)
G.711	PCM	64
G.726	ADPCM	16, 24, 32
G.728	LDCELP (Low Delay CELP)	16
G.729	CS-ACELP	8
G.729A	CS-ACELP, but with less computation	8

Table 1: Codecs defined by the ITU-T and bandwidth requirement

2.3 Quality of Service requirements

Voice transmission over a packet network has often more requirements than any conventional Internet application, such HTTP traffic or file transfer protocol (FTP). Voice is a real-time application, where users are sensitive to loss of quality. To provide the best quality of service, VoIP defines special requirements, typically with three main factors: delay, jitter and packet loss. These key factors are presented respectively in the following sections.

2.3.1 Latency

The overall latency (or delay) is defined by the time that takes to the voice data to travel from the speaker's mouth to the callee's ear. Acceptability of time-delay for time sensitive applications is defined by the ITU-T G.114 specification (ITU-T[G.114], 2002). A delay between 0 and 150ms is acceptable for most applications, where users will not notice the difference with a traditional phone call using the PSTN. Between 150ms and 400ms, the delay is still acceptable but noticeable from users. A delay superior to 400ms is usually not acceptable under normal circumstances.

One way delay depends of five factors: encoding delay is the time taken by the en-user device to encode the voice using a codec. Packetisation delay will occur when the devices proceed the voice data into the IP packet. The serialization delay is the time taken to send the packet on the network, and depends of the network speed, usually the speed configured on a link (WAN links are slower that LAN links). Finally, the queuing delay occurs at the network level, when network equipments have to queue the packet when congestion occurred.

Delay type	*Type*	*Dependence*
Encoding	variable	codec dependant
Packetisation	variable	sample block-size
Serialization	fixed	link speed
Queuing delay	variable	traffic usage and QoS policy
Network delay	fixed	medium used

Table 2: Latency types

2.3.2 Jitter

The Jitter is defines the difference of latency between packets onto a data stream. In real-time applications, like VoIP, packets should be received as a continuous flow, to allow a fluid replay of the voice data. A high jitter value forces the network devices to store previous packets in a buffer, before playing the voice sample to the callee. This process is resource intensive and can be responsible of network congestion. The voice quality is dramatically degraded when jitter cannot be corrected, resulting in gaps in the speech.

2.3.3 Packet loss

Packet loss is also an important parameter in VoIP quality. Although local networks are reliable with very low packet loss rate, WAN and wireless links are subject to packet loss. VoIP requires less than 1% of packet loss to provide good quality of service to users.

2.3.4 Quality of Service models

On packet networks, any application will send packets across the network, in a reliable manner or not (TCP acknowledge packets while UDP does not provide error detection). With no Quality of Service applied (default behaviour of networks), packets from different applications are processing without any distinction: packets are proceeded on a first-in first-out basis. With the use of time-delay sensitive applications, like VoIP, new QoS models has to be developed. Three main QoS models exist:

- **Best-effort model:** this is the default behaviour of network equipments, there is no distinction between applications and packets are processed equally. It is not important when and how packet arrive, and there is no prioritisation of flows.

- **IntServ model:** Integrated Service was the first QoS model developed. Application uses signalling to reserve bandwidth (RSVP protocol). All network equipments in the path need to support this feature, and connection is not established if the bandwidth requirements are not met. This model provides end-to-end QoS but is complex to set-up.

- **DiffServ model:** the Differentiated Service model overcomes the limitations of the IntServ model: flows are classified into classes and processed on a per-hop behaviour. There is no signalling protocols used (no overhead) but configuration must be consistent on network devices.

Of the three models presented, DiffServ model is the most appropriate to converged networks, carrying voice and other time-sensitive applications. Best-effort model is the Internet model, there is no flow priority processing onto ISP public backbone. However, QoS can be used with certain ISP, but this service is often not free-of-charge. DiffServ model is a scalable solution, where most of networking equipments support this model. Configuration consist in identifying, classifying and set prioritisation to the traffic. Then different queuing systems are used to reduce time-delay and packet loss, depending of the class requirements.

2.3.5 Quality measurement

As presented before, voice quality is affected by the compression algorithm and the link quality. However, in ideal conditions, the network does not influence on the voice quality: delay should be really short (less than 150ms) and jitter value should be null. The measurement of voice quality can be performed using either by assessing the voice quality by a group of people, or by comparing the input and output signals.

The Mean Opinion Score (MOS), that gives a numerical value (from 1 to 5) of the voice quality after compression. The voice quality is assessed by a group of people, based on different sentences. Perceptual Speech Quality Measure (PSQM) and Perceptual Evaluation of Speech Quality (PESQ) allow to evaluate the voice quality by an automated model, comparing the received signal with the original signal.

2.4 VoIP Protocols

Transmission of voice on IP networks needed the creation of new protocols to comply to the requirements, presented earlier. The voice application is defined by two channels: the voice data channel and the signalisation channel. The voice channel is in charge to carry the voice from a user to the other, and the signalling channel create a session between users and VoIP devices, to allow the establishment and monitoring of calls. The main signalling protocols are H.323, SIP and IAX2. For the needs of the study, SIP and IAX will be presented more in details. A brief security aspect will be also presented for the signalling protocols, while the security solutions for voice data protocols will be presented separately.

2.4.1 H.323

H.323 is a recommendation from the ITU-T that specifies the transmission of media over IP network. It is an adaptation of the H.330 standard, originally developed for media transmission over ISDN. H.323 specifies the signalisation of the media transfer, allowing the end-points to exchange capabilities (codecs, bandwidth required), start the transmission, monitor the voice channel and terminate the communication. The H.323 specification relies on three main protocols:

- **H.225 (Call signalling):** establish the communication between two H.323 devices.

- **H.225 (Registration, Admission and Status, RAS):** exchanges the messages over the H.225 call signalling channel to register users, notify end-points of bandwidth changes, exchange devices capabilities set-up, monitor and terminate calls.

- **H.245 (Control signalling):** this channel allows the exchange of control messages related to media channel. Ports, codecs and media flows are controlled thanks to this protocol.

The H.323 recommendation specifies a network architecture to allow the good functionary of protocols defined above. This architecture is described by the following network elements:

- **Gatekeeper:** provides services to users, like registration, user authentication and address resolution. Call admission can flow through the gatekeeper (optional, not the default behaviour), where it relays the call admission control (H.225 protocol).

- **Gateway:** allow the communications between H.323 network and other networks (like PSTN, ISDN). To establish communication outside the H.323 network, all traffic flows through the gateway.

- **MCU (Multi Control Unit):** manages conferences, where more than two users are communicating. The MCU allows the transmission of audio and video to all conference participants.

A H.323 call between two end-points will require the use of all protocols and network devices described above. The major drawback of H.323 is its complexity : a simple audio call requires five connections in total (Figure 2), and four of them have to be handled by the user. This is problematic in secure environments, where firewalls or NAT (Network Address Translation) are in operation to prevent outsider to penetrate the private network. Ports are randomly negotiated, making the firewall configuration more complex, or the need of H.323-ready firewalls, that will perform deep packet inspection and have an understanding of the protocol. However, this solution has a cost in terms of price and network performance.

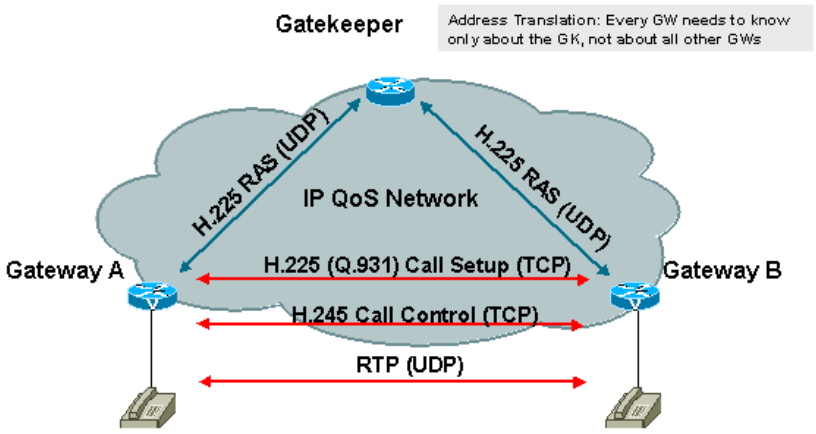


Figure 2: H.323 call (Cisco-5244, 2006)

2.4.2 SIP

Session Initiation Protocol (SIP) is an application-layer control for signalling media channels over an IP network. It is defined by the IETF as a standard (Rosenberg, 2002). SIP describes the establishment, modification and termination of media, like voice or video calls. SIP is a simple text-based protocol, like HTML, sending messages between peers in clear and human-readable text. SIP relies onto other standardised protocols, like DNS, and uses email-like URL to identify the users. The SIP standard defines the following network equipments:

- **Proxy server**: it acts as an intermediate between users to set-up and terminate calls. It routes the call, and allow call policy enforcement (for example, is the caller allowed to call abroad?).

- **Registrar server**: allows the users to register onto a domain, using user-names and passwords. Only authorized users are allowed to register, and then be able to place / receive calls.

- **Redirect server**: this server is in charge to redirect requests in function of the location of users. It acts like a DNS server, where the proxy server will ask the registrar server to resolve domain names. This server is useful when calls are placed in different domains.

The establishment of a VoIP call using SIP is illustrated in Figure 3. The different servers defined by SIP must be logically distinct but can be physically hosted within the same machine. A single application can provide many services (i.e. Asterisk PBX provides a SIP proxy and registrar service). However, the media (voice or video) is usually carried directly from the caller to the callee. The case where voice is processed by any other device will be presented later in the study.

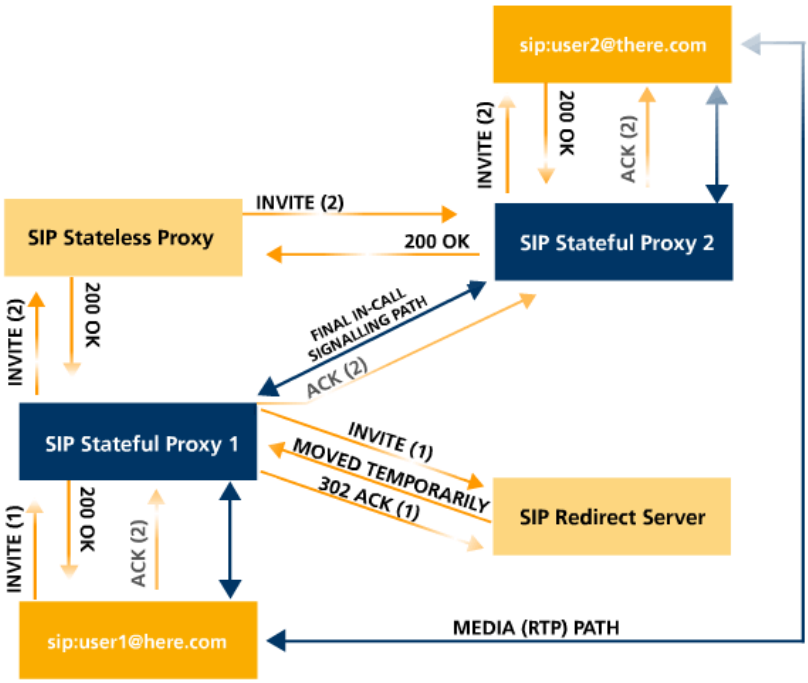


Figure 3: SIP signalling (unknown reference)

2.4.3 RTP

Real-Time Protocol is the standard that defines the packets to carry voice data over IP network (Schulzrinne, 2003). RTP is an application-layer protocol, carried unreliably and connectionless by UDP (transport layer). The RTP packet is optimised for real-time packet transmission: packets are not acknowledged, they are timestamped and have sequence numbers.

Time-stamping allow the receiver to play back the packets at appropriate intervals. Unless in any other Internet application, it is important in VoIP to play back the packet with respect of the original timestamps. Sequences numbers helps the receiver to play back the packet in the correct order. None action is taken when a packet is lost, as retransmission systems require connection-oriented protocol (like TCP) and will produce a too high delay and data overhead.

The use of UDP as transport method results in a header of 8 bytes, reducing the overhead (in opposition to the 20 bytes long TCP header). The data payload size is variable, depending of the packetisation rate and the codec used. Figure 4 present a RTP packet using two different codecs, with a packet rate of 20ms of voice per packet.

By default, 20ms of voice is packetised in a IP packet. In this case, the total frame overhead is 20% when using G.711 codec, and reaches 200% when using G.729. The packetisation rate can be changed to optimise bandwidth usage: encapsulating 30ms of voice into a single IP packet will reduce the overhead, as the RTP payload will be bigger in comparison to the packet header.

To reduce headers overhead, a standard has been proposed: cRTP (Koren, 2003) allows the compression of IP, UDP and RTP headers, reducing the total overhead from 40 to 2 bytes. This compression is performed on a link-by-link basis, depending of the router capabilities and usually across slow WAN links. Although it seems to be a good solution to optimise bandwidth utilisation, this implementation requires configuration on every router on the media path, and is not scalable.

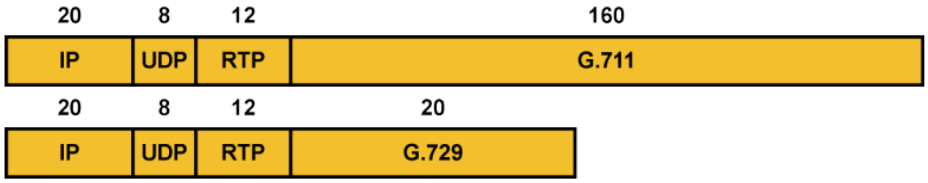


Figure 4: RTP frame

2.4.4 IAX2

The Inter-Asterisk eXchange protocol, is the native Asterisk protocol to exchange voice and signalisation between servers. IAX was developed to interconnect Asterisk PBX, acting like a trunk link. IAX carries into a single channel both signalling and voice data. Signalling is carried using *Long Frames* (12 bytes header) in a reliable manner, and voice data is carried by *Mini Frames*, with only a 4 bytes long header, sent in an unreliable manner.

IAX overcomes the complexity of H.323 by using a single channel to carry both signalling and voice data: a single connection is established between the users. IAX is also able to perform the multiplexing of many voice flows into a single channel. IAX protocol also supports encryption and authentication. These features will be developed later in the study.

2.5 Conclusion

This first chapter presented a general background of VoIP. The different steps to convert the original analog signal into data have been introduced, where the importance of the codec used have been brought in light. The requirements for voice over a packet networks were also presented: the three main parameters,

delay, jitter and packet-loss must have values as low as possible to provide the best quality of service to users. The QoS models in a packet network were also introduced: despite its needs of end-to-end QoS agreement, the DiffServ model is dominant for its flexibility and performance. VoIP quality is measured using MOS rating, from 1 to 5 where 5 presents the best sound quality.

The main protocols solicited in VoIP systems were presented: signalling protocols allow to establish and terminate calls, while transport protocols carry the voice data. SIP and IAX are more recent than H.323, but H.323 is still used for its maturity. The security aspect of VoIP system will be presented in the next chapter, with the help of previous researches related to this study.

CHAPTER 3: LITERATURE REVIEW

3.1 Introduction

A general background of voice over IP functionality and main protocols have been presented in the previous part of the study. It has been shown that voice is a complex application to be used on standard packet networks. A good quality of service is important to users, used to use traditional telephony service, which did not suffer of constrains regarding delays, jitters and packet loss. The mains signalling protocols, H.323, SIP and IAX were presented. This study will focus on SIP and IAX protocols, as they are more recent and more likely to be used in recent VoIP systems.

The use of standard packet network as support for voice introduces new threats to voice data (Benini, 2008): these will be presented and some examples will be provided. In a second time, some secure techniques will be presented, with the help of previous studies. The solutions presented to secure voice contains advantages and drawbacks, that will be also presented in this part.

3.2 VoIP threats

Like any other important application, voice needs to be used in a secure manner, especially when it is carried over Internet. Voice system were reliable and secure when using the dedicated PSTN network. With VoIP, voice is carried over Internet: it is as vulnerable as any other application (Benini, 2008), and attackers do not need to be on the physical path (Collier, 2005) to be a threat to this service. Voice is considered as a critical service: communication (Thermos, 2007) and business relies a lot on telephony. This part will present the main existing threats to VoIP, and the consequences if an attack is successful.

3.2.1 Denial of service

Denial of Service (DoS), also called Distributed DoS (DDoS), is an attack that intent to disable a target in order to prevent legitimate users to access to its service. These attack are performed by sending large amount of requests to the server to exhaust resources, and force it to crash or be unresponsive to other users. DoS attacks are very common (Labovitz, 2009) in any system open to the Internet.

The flooding technique consist in sending large amount of requests to a machine. This DoS attack represents 45% (Labovitz, 2009) of the DoS attacks worldwide. Regarding the same annual report, the exploitation of protocol weaknesses represents 23% of the attacks.

The main DoS attack in VoIP targets the SIP servers (Ormazabal, 2008), particularly against the SIP registrar. Sending a large amount of REGISTRAR or INVITE messages the proxy server will keep the session in buffer until the time-out expire (Rosenberg, 2002), like specified in the SIP standard. This will result in a resource exhaustion, and a denial for the users registered within the domain.

Some vulnerabilities were found in the SIP protocol, where sending malformed packets (Wieser, 2003) makes some SIP devices crash. Although these vulnerabilities are known from the manufacturers, flooding malformed packets is still a threat (Albers, 2005) for SIP servers, that have to handle a large amount of requests. The best countermeasure so far consist in SIP honey-pots (Nassar, 2007) to retrieve maximum of informations about the attacker in order to establish a blacklist (Jackson, 2010).

3.2.2 Eavesdropping

Eavesdropping is by definition the act of listening secretly to a conversation. In classic telephony system, an eavesdropper can listen to a conversation physically (i.e. listen from the next room) or place themselves in the path (plug another telephone in parallel). Placing a device in the PSTN network is a difficult task: connection uses physical, and equipments are not accessible to anyone.

In opposition to PSTN, VoIP uses public architecture: anyone with malicious or not intention can be placed in the communication path. For example, the use of hubs instead of switches as network equipments will permit any users within this network to monitor all traffic. In an intentionally way, IP addresses can be spoofed, making the attacker in the communication path (hijacking, described below). Once the eavesdropper can retrieve traffic, packets can be captured with any traffic analyser or packet capture tools[1]. This process is even easier when using insecure wireless network.

Default behaviour of VoIP protocols send the voice data in clear, making eavesdropping easier. Encrypted traffic can be captured as well, and the attacker can break the key to disclose the data. This process requires brute-force attack, where attacker will try all the possible keys. This process is processor intensive and is directly dependant of the key-size used to encrypt the data. For example, using a 256 bits key will give one chance of 2^{256} ($1{,}15 \times 10^{77}$) to find the correct key.

1 A non-exhaustive list can be found at http://voipsa.org/Resources/tools.php

3.2.3 Hijacking

Hijacking consist in stealing somebody's identity and pretend to be this person. In VoIP, hijacking allow an intruder to pretend to be legitimate user (Collier, 2005), and gain access to the resources (placing calls, voice-mail access). SIP implements basic authentication, using user-name and password. Unfortunately, the security parameters are sent in clear.

Common hijack attack in VoIP systems is the brute-force attack of SIP registrar servers to retrieve users credentials. This process requires previous traffic analysis, to retrieve user-names. Once the attacker has successfully registered, they are able to place calls and inflict money loss, (Pauli, 2010) for the hijacked company. Social engineering can also occur, where a hijacker can impersonate a user or a service. For example, pretending to be from the IT department and asking for user's password can help the attacker to get into the system.

3.3 Evaluating VoIP performance

The most interest given to the VoIP security is the performance impact for the users. Presented in earlier in the VoIP background, the transportation of media over IP network is very sensitive to environment changes. Latency, jitter and packet loss are the most important parameters that need to be monitored, in order to provide the best quality of service. This part will present some work undertaken to assess VoIP performance in general, and then when security mechanisms are applied.

3.3.1 Scenarios

When measuring VoIP Quality of Service, the three main VoIP parameters defined earlier are measured: delay, jitter and packet loss. These three parameters, allow to define if the communication comply to users expecting: a reliable and good quality telephony service.

In most cases, scenario established to measure QoS are not emulated, by using network simulators. Results are more accurate and closer when real network is used (this will be referred as real-scenario in this study). SIP is commonly used as the signalling protocol (Guillen, 2009 ; Clayton, 2007 ; Detken, 2008 ; Mohiuddin, 2008), and different of the presented secure solutions are tested to secure the voice data. A common scenario will implement a SIP registrar and Proxy server.

The scenario usually evaluate the VoIP quality under many network circumstances: the network scenario should be able to measure calls made

within a private LAN (hi-speed links) and when using Internet links (WAN links, slower). Figure 5 illustrates the two different approaches.

In the LAN scenario, the links are Fast Ethernet (100 MB) and allow internal users to call each other. This implementation of VoIP replaces the internal PBX. The second scenario simulates the interconnection of many private branches, across Internet. The WAN link simulates the Internet cloud and communication across this link must be secured. It is considered that the link load cannot be predictable, and usually no network QoS model is supported. The link is slower than within the private LAN.

Simulated scenarios are an alternative to real-scenarios: they allow to make tests without the need of specific network equipments and servers (Salama, 2006). However, simulating networks requires powerful machine to emulate many computers and servers. Simulating network behaviour requires complex software, like OPNET (Salama, 2006). Values can be inaccurate in opposition to real-scenarios, as a single machine is in charge of simulating the network and different nodes. The study will later present that results using simulated scenarios are different from real-scenarios.

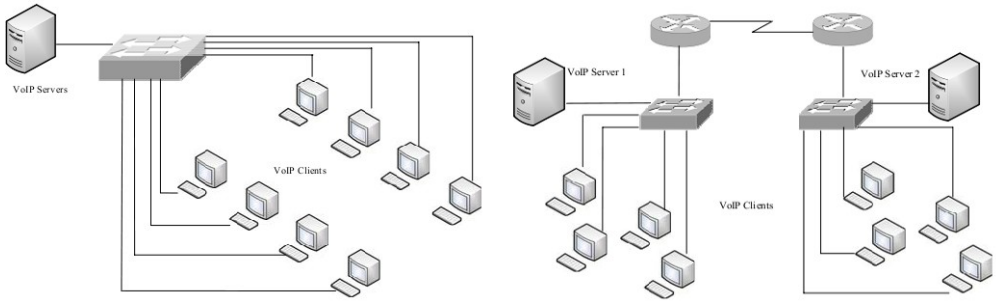


Figure 5: LAN and WAN scenarios (Guillen, 2009)

3.3.2 Measuring VoIP QoS

VoIP quality is based on three parameters, presented in the background of the study. However, only delay and jitter can be measured on a network point of view. The jitter can be easily measured as close as the receiver as possible: the packet arrival time suffers from the process across the network.

The delay can be measured in two manners: ideally, the end-to-end delay, measure the time taken for a packet from the sender to the receiver. This measurement is easy to set-up in simulated networks but very complex onto real-scenario.

In real-scenarios, sender's and receiver's clocks must be perfectly synchronised and packets need to contain time information (RTP header does not implement this feature). However, some time stamp can be added at the application level

to compare the time between the departure and the arrival of the packet (Wowra, 2007). Unfortunately, no more information were found about the methodology and tools used. It is also possible to measure end-to-end delay using Internet Control Protocol (ICMP). This solution is not suitable for VoIP traffic, as the ICMP traffic has not the same QoS parameters across the network, and traffic may use a different path (Li, 2006).

When using real-scenarios, delay is measured by capturing packet as close as possible from the destination (Guillen, 2009 ; Alexander, 2009) and inter-arrival delay is measured. Measuring the arrival time of a packet in comparison to the previous one is an indication of delay. Ideally, packets arrive every 20ms (time of voice sampling). This delay can vary (jitter) depending of the charge of network equipments and the protocol used. For example, the receiver can receive the packets at t+20 ms (ideal, delay=0ms), t+25ms (delay=5ms), t=16ms (delay=4ms).

In Figure 6 presented above, delay measured has a value of 20ms in most cases. The actual delay measured is relative with the previous packet. 20ms delay, in this study, correspond to the amount of voice carried in packets after sampling (the *SERV → CLI (WAN)* delay is equal to 15ms, due to a sampling of 15ms). The packets are not delayed by any encryption algorithms in this case, as the arrive with a perfect timed interval of 20ms. However, these results are normal in this scenario, where only 4 simultaneous calls are measured in a non-stressed LAN scenario.

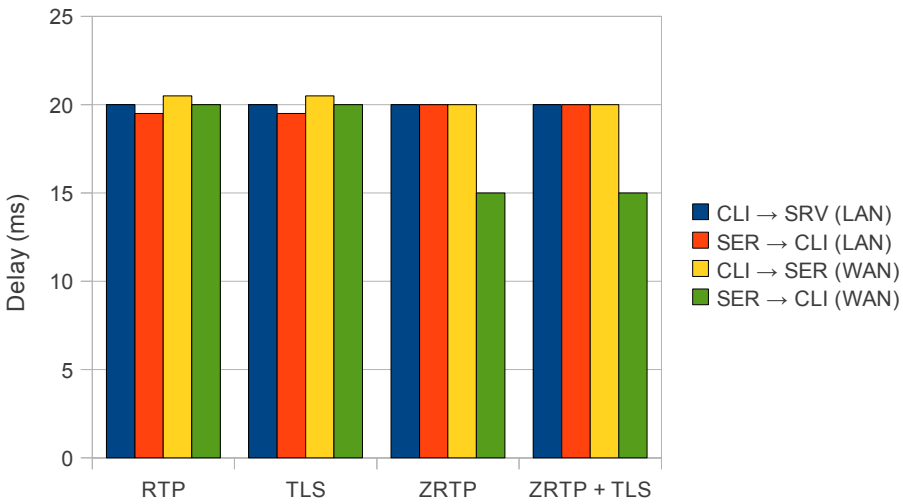


Figure 6: Delay measurement (Guillen, 2009)

3.4 Securing VoIP

The main VoIP protocol presented above allow the transmission of voice over a IP network. However, these protocols, except IAX2, do not define any security mechanisms to preserve confidentiality and integrity. Both signalisation and voice are carried over the network without any security applied. Without any security applied, the transmission of voice is subject to treats, like eavesdropping or hijacking, presented in the previous section. DoS mitigation techniques will not be presented in this study, as the threat is very specific and hard to mitigate. Besides, DoS attacks can occur to any service open to Internet.

Secure solutions used to preserve confidentiality (to prevent eavesdropping) and integrity (prevent from hijacking) will be now presented. In a first time, both most developed tunnel technology will be presented: IPSec, a tunnel at the transport layer and TLS, at the application layer. Then, Secure RTP, a secure adaptation of RTP, will be described in details. The key exchange protocol that apply to SRTP will also be presented. The last section will present briefly the security mechanisms that IAX protocol implement.

3.4.1 IPSec

IPSec is a framework using a set of protocols to create a secured and authenticated tunnel between two parties. IPSec provides Confidentiality, Integrity and Authentication of data across the network. Confidentiality is performed thanks to symmetric (DES, 3DES or AES) or asymmetric (RSA) encryption algorithms. The key exchange is ensured by the Internet Key Exchange (IKE), that relies on Diffie-Hellman method to agree with a shared secret. Integrity is ensured by calculating on both side on-way hash functions, like MD5 or SHA-1 algorithms. Authentication is performed thanks to pre-shared keys, configured on the devices. IPSec can operate in two different modes:

- The transport mode of IPSec encrypts only the data payload and IP headers are not modified. The whole packet (data and header) is authenticated with a HASH value. This mode is usually used for host-to-host communication, where routing values are not changed.

- The tunnel mode encapsulates the original IP packet within a new packet header. The original IP header and data are encrypted and authenticated. The original routing information are encrypted, and authenticated by a hash function. This mode is usually used when interconnecting sites.

IPSec is has been evaluated (Barbieri, 2002) to secure VoIP in real-scenario.

The mains parameters, delay and jitter have been measured (Perez, 2006) and confronted to other solutions, that will be presented later in this study. As a result of numerous tests (Salama, 2009), the QoS parameters are dramatically affected by the encryption process (Guillen, 2009). The latency measured using IPSec increased the value of 132% (Perez, 2006) in simulated scenarios and reached 200% (Guillen, 2009) in real-scenario. Results are presented in Figure 7 when both voice and video traffic were assessed with and without the use of IPSec as secure solution.

The increased latency when using IPSec is sometimes the result of loss of QoS parameters when IPSec is used in tunnel mode. To prevent the loss of header loss, some techniques (Völker, 2007) were developed to preserve QoS parameters and reduce latency delays. The additional overhead is situated between 20% and 50% (Guillen, 2009), depending of the codec used. An experimental modification of IPSec mechanisms, called cIPSec (Barbieri, 2002), allowed to save up to 6% of bandwidth usage and 10ms latency on a 64kps link. However, the process is resource intensive.

IPSec, mostly used in tunnel mode (Doraswamy, 2003) is more resource processing (Ferrante, 2005) than the transport mode, although less secure. The processing of encryption with IPSec results in an important additional delay (Ahmed, 2008) and an important non-negligible impact on the CPU performance regarding the encryptions mechanisms used (Figure 8). The CPU effort increments up to 5 times more than when encryption is not used (Ferrante, 2005) and usage remains at 100% usage, due to the large throughput configured in this research.

Securing VoIP service using IPSec has a large influence on the quality of service (Perez; 2006) and CPU usage (Ferrante, 2005). These factors are related (Ahmed, 2008) as a stressed machine will be less likely to process packets in short time.

Both scenarios implemented the WAN scenario. The same number of concurrent audio and video calls are measured. In the second scenario, the WAN link was stressed using a traffic generator. Whereas the delay is similar in the first scenario (no stress is applied to the WAN link), the latency on a stressed network is dramatically increased.

When implementing IPSec, network QoS parameters are responsible for high latency values. When encrypting the packets, the QoS parameters are not conserved, as they are encrypted with the original IP header. This result in the same treatments for the voice packet and other applications. Packets are processed with the same priority (Best-effort model). In opposition to the delay, jitter and packet loss are reasonably low and still acceptable (Perez, 2006) under normal circumstances when IPSec is used to secure VoIP.

In a simulated environment, the processing of encryption and authentication also had a large impact on VoIP performance. Measurements during network simulation (Salama, 2009), using OPNET, prove that the voice delay is increased by 170% from and the jitter increased by 200%.

Apart form the performance impact, IPSec suffers of other drawbacks. Configuration is complex and requires manual intervention of the system administrator on each tunnel-end network equipment. Deploying IPSec solution to a large topology requires complex configuration and limits the scalability (Berthelot, 2009).

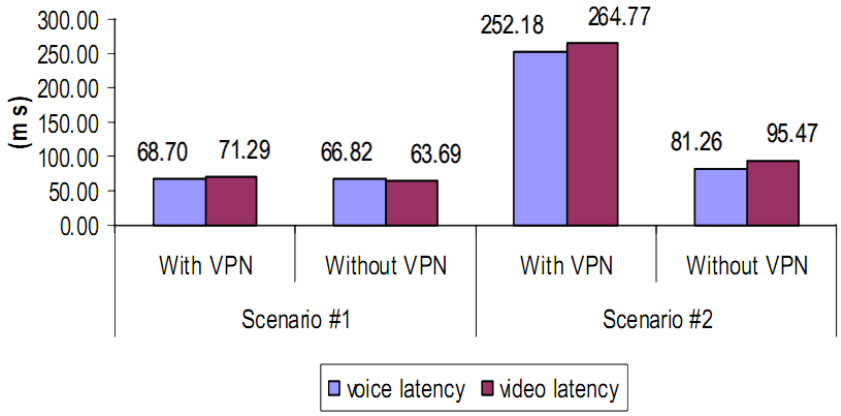


Figure 7: IPSec delay measurement (Perez, 2006)

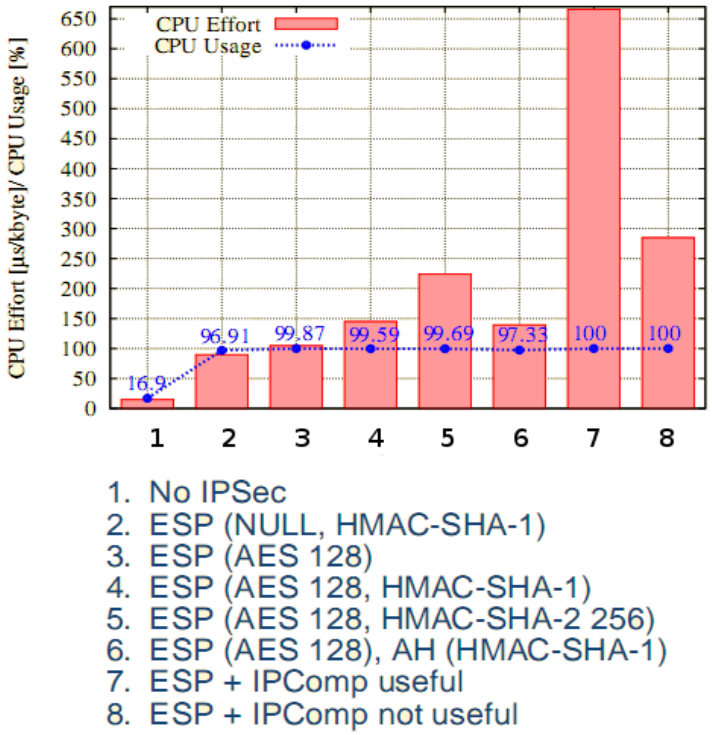


Figure 8: IPSec CPU usage (Ferrante, 2005)

3.4.2 Transport Layer Security

Commonly called SSL (Secure Socket Layer), TLS (Transport Layer Security) is a protocol using cryptography to provide secure communication between entities. TLS works at the application layer of OSI model. Originally developed for HTTP traffic to secure commercial transaction, TLS is now deployed for many applications, like emailing, instant messaging and voice over IP. TLS relies on a client/server architecture: encryption provides data protection against eavesdropping while authentication preserves data integrity. Public Key Infrastructure (PKI) is used to provide both encryption and authentication. Encryption performed with extremely large keys (1024 or 2048 bits long) to prevent form brute-force attacks. Authentication relies on digital certificates, managed by Certificate of Authority (CA). TLS authentication can be unilateral, where only the server is authenticated but the client remains anonymous, of bilateral (both client and server are authenticated).

Securing VoIP using standard TLS implementation is not a scalable solution, (Berthelot, 2009) because of its lack of flexibility, the complexity of configuration and the certificate management. Standard TLS also introduces a large overhead (Voznak, 2009), adding 117% overhead in comparison to standard RTP packet.

TLS has been adapted to secure real-time applications, like voice, in order to reduce packet overhead. Datagram TLS (DTLS) allows the transmission of RTP traffic using same security features than TLS (Tschofenig, 2006). The RTP packet is encapsulated by a DTLS header and a trailer. Figure 9 presents a RTP packets secured by DTLS.

The main drawback of DTLS is the addition of new headers and authentication field: DTLS ads 23 bytes to the original packet, incrementing the serialization delay by the network equipments. A study demonstrated that for a single communication, DTLS encryption increments the latency of 20ms (Wowra, 2007) in an end-to-end link, without any PBX or network equipment. Under more complex and realist circumstances, results seem to be similar. A study implementing a LAN and WAN scenario (Guillen, 2009) shown that delay and packet loss are very low. Furthermore, the same study proven that jitter is considerably increased when DTLS is used to secure RTP in the WAN scenario, increasing of 200% (from 6ms to 12ms).

In opposition to IPSec tunnelling, DTLS allow the QoS parameters to be kept when sending packets across the network. Moreover, the CPU performance have not been investigated yet. A study on web-servers performance using TLS shown that CPU resources used for TLS sessions is situated between 13 and 58% of the total usage (Coarfa, 2006). This has not been investigated yet for DTLS applied to RTP traffic but results should be similar, as encryption algorithms remain the same.

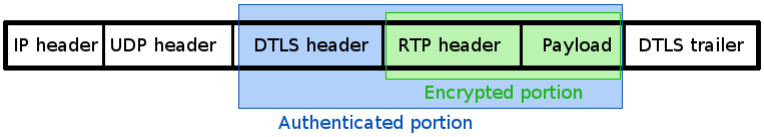


Figure 9: DTLS encryption to RTP packet

3.4.3 Secure RTP

Secure Real-Time Protocol is a standard that provide encryption (confidentiality) and authentication to the RTP traffic. Mechanisms to prevent message replay are also implemented (Baugher, 2004) in this secure adaptation of RTP. SRTP framework defines a set of encryption mechanisms to secure the voice payload from eavesdropping. Authentication mechanism helps to preserve packet integrity are also defined: hash function authenticate the packet header and the encrypted payload. The authenticated portion helps to prevent against packet replay. Encryption is performed using AES (Advanced Encryption Standard) with 128, 192, or 256 bit master key. AES algorithm is a two way encryption system, using a single secret key to encrypt and decrypt data. Authentication of the packet is assured by HMAC-SHA1 hash function, that add 8 bits authentication field at the end of the packet. SRTP relies on

external key management protocols. The three main key management protocols, MIKEY, SDES and ZRTP are presented above.

Multimedia Internet KEYing (MIKEY) is a key exchange protocol (Arkko, 2004) to allow the exchange of encryption parameters using either Pre-shared keys, PKI infrastructure or Diffie-Hellman methods. MIKEY parameters are carried by the SDP channel: SIP supports the MIKEY parameters (Arkko, 2006), but any other signalling protocol will need to be adapted. SDES, (Session Description Protocol Security Descriptions), allow to pass the key using the signalling protocol (e.g. SIP). A new field (Andreasen, 2006) specifies the type of encryption used and the secret key. However, the signalling channel has to be secured in another way, not defined by SDES.

The major advantage of SRTP is the reduced overhead in IP headers (Table 3) in opposition to out-of-band protocols, like IPSec. The SRTP packet has a lower overhead than when using IPSec tunnels: IPSec overhead is 4,4% longer than SRTP in any codec used. When GSM compression is used, SRTP overhead stays reasonably around 10% while IPSec overhead reaches 43%. This significant difference should allow SRTP traffic to be processed more rapidly by network equipments and reduce total delay.

Voice performance has been evaluated (Alexander, 2009) when SRTP is used to secure media. Using a real-LAN scenario, jitter and latency have been measured, in order to compare quality of service using RTP and SRTP. The latency was measured (Figure 10) using inter-arrival packet delays. In both conditions (with or without security applied) the mean delay are very short, where values are very close to 20ms (Guillen, 2009). This is ideal in a LAN scenario where packets do not cross WAN links and only one call is evaluated (the network is not stressed). The study demonstrated that delay also depends of softphones used, and some optimization can be made on software.

The Jitter has also been measured (Figure 11) during this study: the values are slightly different, and should not affect the voice quality. These differences of values are possibly due to the difference in timing between the softphones when processing voice packets (Alexander, 2009), but further investigation need to confirm the results. As the study implemented a LAN scenario, without stressing tools, the pack loss has not been investigated, and should be negligible, as no other traffic interferes with the voice traffic.

METHOD	ALGORITHM	OVERLOAD BW ETHERNET	
		ULAW	*GSM*
Signalling	TLS	0%	0%
Media	SRTP	4,42%	10,10%
	IPSEC	19,47%	43,43%

Table 3: Ethernet overhead (Guillen, 2009)

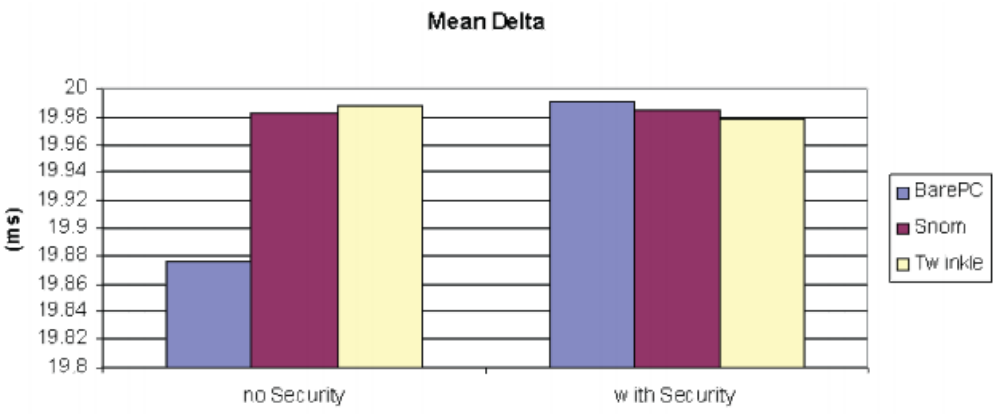


Figure 10: Latency measurement using SRTP (Alexander, 2009)

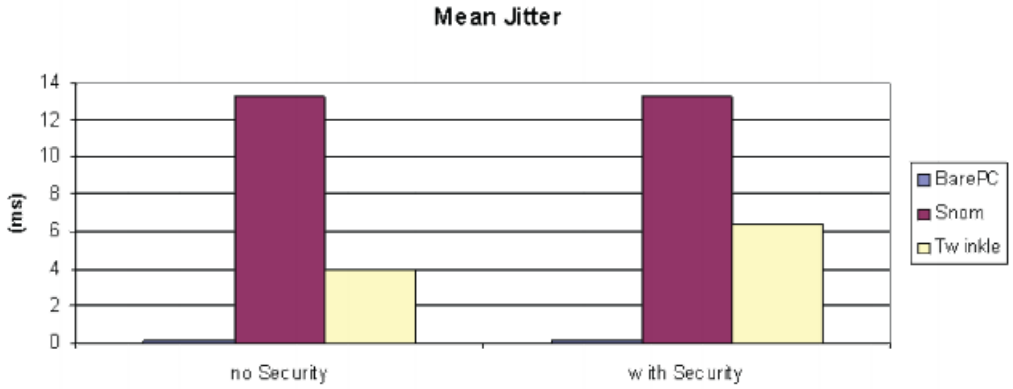


Figure 11: Jitter measurement using SRTP (Alexander, 2009)

3.4.4 ZRTP

Z Real-Time Protocol is a media path keying designed exchange the encryption key, to secure RTP traffic. It has been developed by Phil Zimmermann, the creator of PGP, and is at the moment still a IETF draft (draft-zimmermann-avt-zrtp-21, May 2010). ZRTP, currently in its version 1.10, defines the key exchange mechanism to secure RTP traffic between two peers. ZRTP relies on Diffie-Hellman key exchange protocol, to agree on a shared key to encrypt RTP packets.

Key exchange mechanism is independent of signalling protocol: messages are sent through the RTP channel. This approach present two main advantages: firstly, the signalling protocol does not transfer any secret and voice confidentiality cannot be compromised by an insecure use of signalling protocol. Secondly, passing the key in the media channel is simpler: there is no need to establish a new connection (use another port) that could be blocked by ant NAT or firewall device.

To avoid man-in-the-middle attack, a Short Authentication String (SAS) is generated from the Diffie-Hellman parameters. This SAS can be compared verbally by users to verify the integrity of voice data. ZRTP has been evaluated and approved as a secure protocol (Bresciani, 2009) but VoIP performances have not been evaluated successfully yet (Guillen, 2009).

3.4.5 IAX2 encryption

The Inter-Asterisk protocols implement security mechanisms. Authentication and encryption can be performed only between Asterisk peers, end-to-end encryption is not supported yet. When used in trunk mode, IAX2 performs encryption on a call-by-call basis: encryption keys are different for each call. Authentication can be performed by MD5 hashed password or RSA. Encryption is ensured by AES128, using a symmetric key. Only the payload is encrypted: source and destination call numbers are sent in clear.

3.5 PBX performance measurements

Studies presenting the measurement of VoIP quality of service were described in the previous part. It has been demonstrated that some secure solutions highly affect the quality of voice. However, another factor can influence the VoIP QoS parameter. The resource usage of VoIP system has not been presented yet in this study. The performance cost of in term of resource will be presented in the first section, and then the encryption mechanisms performance will be presented, with the help of previous studies.

In both testing scenarios, LAN and WAN topology, a virtual PBX was used to allow signalisation of calls. This device can be a single point of failure for the domain if it fails in registering users, establish and monitor calls. The VoIP quality of service can also be affected if the PBX has difficulties to handle requests from clients. Its availability is more important than any other Internet service, as any important delay or fail or service will be immediately noticeable from users.

The response time of the PBX will affect the maximum delta (maximum latency) when measuring delays in VoIP. The call will take time to be established, forcing the users to wait until signalling protocol allow the establishment of voice data. This delay is not noticeable from users when the PBX is well-dimensioned.

A study evaluated the behaviour of a SIP server when large amount of concurrent calls are made (Nahum 2007). The PBX evaluated, in a LAN real-scenario, was openSER (now openSIPS[2]). The server was able to handle a large amount of requests (3,06 GHz) and Gigabit links were used to reduce packetisation delay. Using SIPp (which will be described later in this study) to generate requests, the server was stressed with SIP requests. Results from the study are presented in Figure 12.

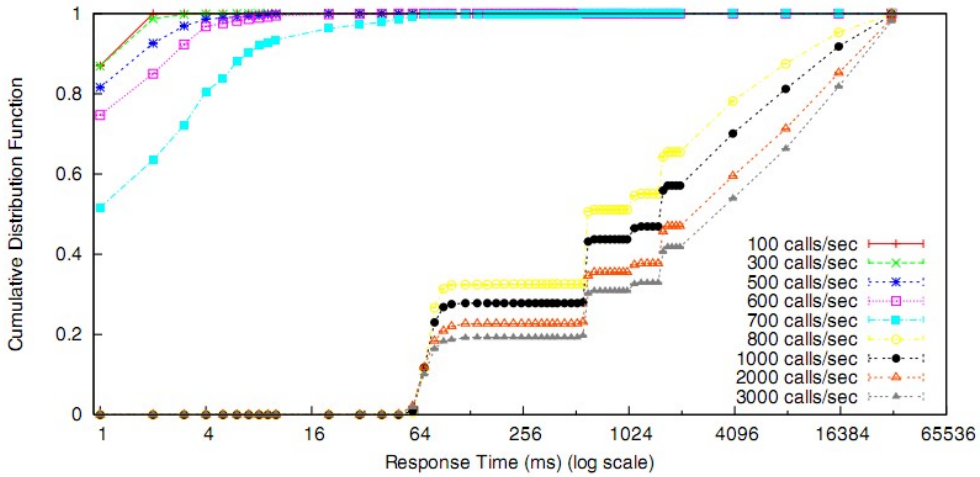


Figure 12: Response time of SIP server (Nahum, 2007)

The results of the tests shown that the server does not respond in a linear manner to requests. There is a large gap between 700 calls and 800 calls made per second. This is due to the overload of the server, that is unable to respond to requests in time. According the graph, the server is considered overloaded for all requests above 700 calls per second (Nahum, 2007). The response time becomes considerable, 64ms when the server is overloaded.

2 OpenSIPS is a free SIP server available from www.opensisps.org

Another study demonstrated that the number of possible concurrent calls is directly dependent of the CPU power (Ahmed, 2008). The study evaluated an Asterisk PBX performance during many concurrent calls. The Asterisk server was stressed using SIPp software, making numerous numbers of calls until the server fails or CPU reaches 100% usage.

As it is in the role of SIP proxy to perform transcoding (when client cannot agree on codecs), the study investigated the two cases where transcoding had to be performed, and when Asterisk did not have to perform any codec transcoding. Figure 13 presents the results obtained with a AMD Sempron 1,5GHz CPU, running Asterisk PBX version 1.4. It is shown that transcoding has a large impact on Asterisk server: the server can handle 80 concurrent calls before being overloaded (CPU usage reaches 100%) while almost 130 concurrent calls can be performed when there is a codec compatibility between clients.

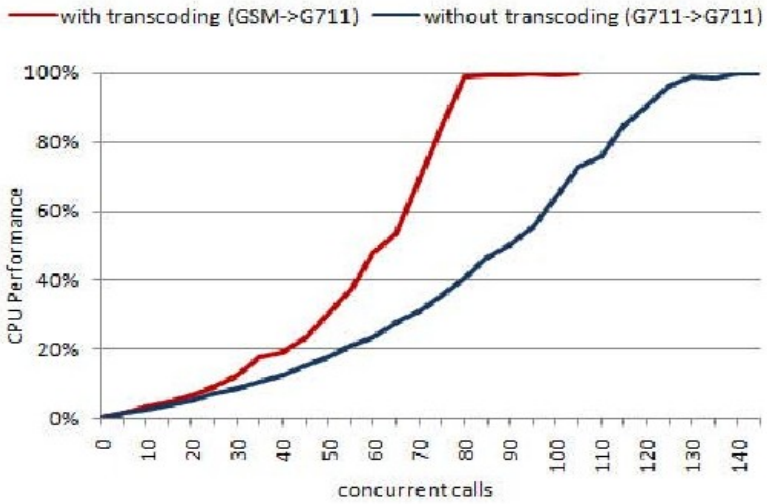


Figure 13: Asterisk CPU usage (Ahmed, 2008)

3.6 Conclusion

The three main VoIP security issues have been presented in this chapter. The major vulnerabilities that voice encounter are similar to any other Internet service, but secure solutions must respect the VoIP requirements. VoIP servers, like WEB servers, are likely vulnerable to DoS, while hijacking (also called scam) is a threat at the user level (similar to email spammers attempting to retrieve bank details). Eavesdropping remains the main threat to VoIP. Eavesdropping conversation is a threat specific to VoIP: usually, eavesdroppers try to retrieve data information (logins and passwords), but in the case of commercial transaction attackers are gaining interest in wire-tapping telephone conversation.

Security techniques against VoIP threats have been presented in this part of the study. Standardised solutions, like IPSec and TLS tunnelling, can be used to secure voice transmission. IPSec is more likely used to secure voice payload for its simplicity and reduced overhead in comparison to TLS. However, TLS is most commonly used to secure signalling. An evaluation of all presented technologies has been performed, in term of quality of service and resource usage. It has been proven that tunnelling solutions, like IPSec, degrade voice quality by affecting the delay and jitter. SRTP presents better results in terms of quality of service than tunnelling protocols. However, SIAX2 has not been investigated deeply, and researches about resource usage of VoIP PBX could not be found. Despite the numerous number of studies presented, none have presented results with variable number of concurrent calls. The next chapter of this study will present a methodology to perform such evaluation, in order to complete the set of studies presented in this part.

CHAPTER 4: DESIGN AND METHODOLOGY

4.1 Introduction

The literature review, presented earlier in this study, described previous work about VoIP quality of service when security mechanisms are applied. Voice quality and server performance have been evaluated in most common scenarios. However, the IAX2 protocol has not been evaluated yet, and not compared to other secure solutions. It also seems that the importance of the relation between resource usage and quality of service was missed in previous studies. This study will complete previous researches, by measuring both quality of service and resource usage, for a variable number of calls. The experiment framework is presented in Figure 14.

This part will present the design of tests that will be performed. Choices made for the tests will be presented and justified. In a first time, the network scenario will be presented and justified. Then, a short introduction to Asterisk, the software used as a PBX will be presented, as well as the role that it will have in the tests. The two last parts will describe how the VoIP quality and CPU performance will be measured and evaluated.

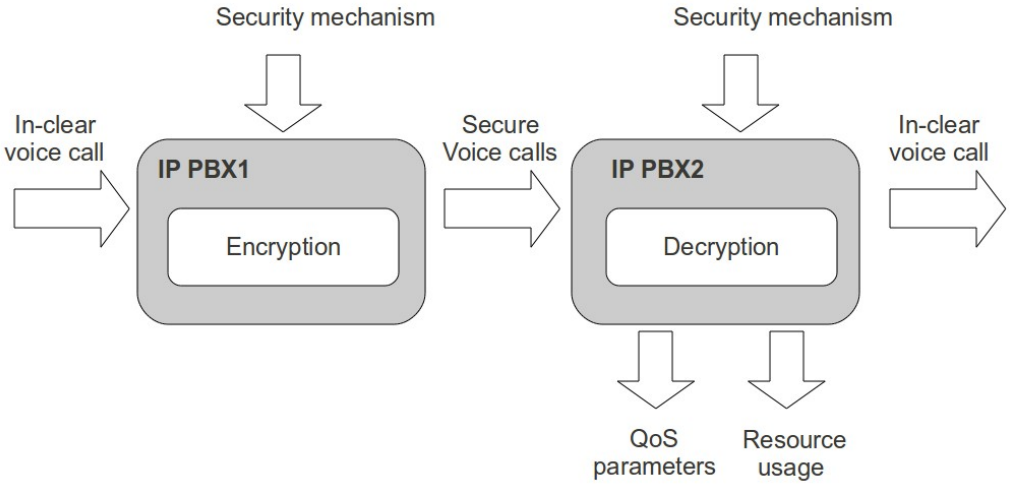


Figure 14: Framework diagram

4.2 Network scenario

The literature review allowed to prove than results are more realistic and accurate in real-scenario, using network equipments instead of virtual machines. The scenario chosen in this study will represent two distant sites, connected via Internet. Both sites have a PBX to manage their local SIP domain. Figure 15 presents the topology implemented in this study.

For the study, the users in both sites are using softphones onto their computers. A dedicated server act as a PBX to manage the local SIP domain: it acts as a registrar server. In this scenario, it is considered that network policy disallow the encrypted traffic in the local area: this can be justified by, for example, by the need to record communications or prevent cover channel. All calls within a SIP domain, considered as safe, will remain unencrypted.

However, calls passed across Internet should be secured to prevent from external threats, presented before in this study. The encryption will be performed from PBX to PBX, to reduce complexity of routers configuration. Another advantage of this method consist in identifying the traffic: VoIP uses random ports for each call, if the encryption is performed by the routing devices, they must be VoIP-protocols-aware. In opposition the voice traffic, once encrypted by the PBX, is sent through a single port to the other end. Firewalls and routers will need to open one port, facilitating the NAT configuration.

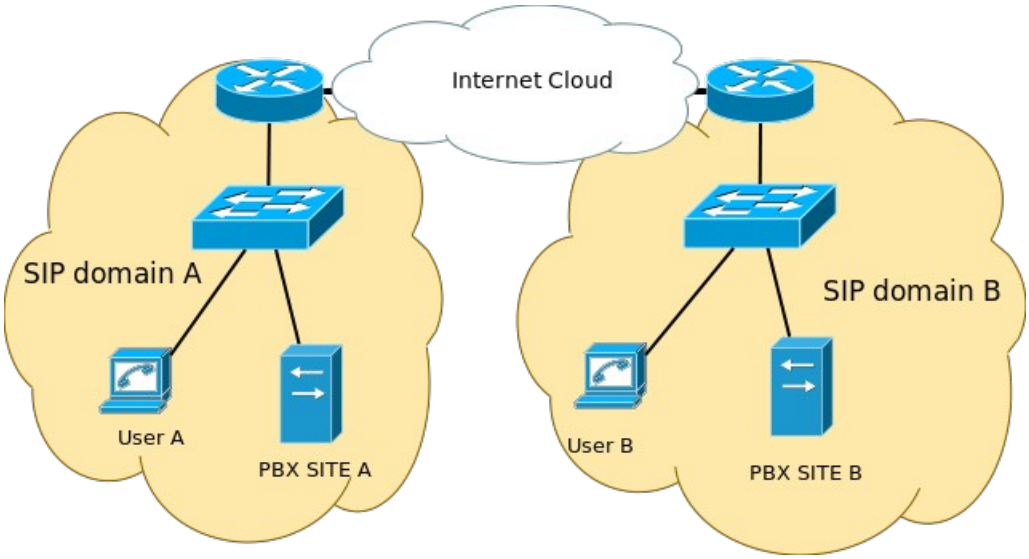


Figure 15: Implementation

4.3 Asterisk PBX

Asterisk is an open source PBX software developed by Digium (www.digium.com). In opposition to standard PBX, Asterisk is hardware independent and runs onto any ordinary computer running a Linux distribution. Asterisk, originally developed by Mark Spencer, is now one of the most popular widely deployed telephony project. The software supports most of the VoIP protocols (SIP, H.323, MGCP, IAX, RTP and SRTP) and can act as a gateway (call routing, codec and protocols transcoding, connection to PSTN). Asterisk includes as well a complete set of applications, like voice-mail, voice recorder, dial plan, LDAP and database access for large companies.

Asterisk is a good alternative to expensive and proprietary solutions for small and medium size companies. However, the major drawback of this application is the configuration management: there is no graphic interface[3], and all configuration is made by editing configuration files. The main configurations that will need to be performed are presented below.

4.3.1 SIP trunk

The first tests will present a common SIP trunk, solution existing and widely deployed to interconnect distant sites. Both the signalisation and media are managed by the PBX, acting like a proxy. Calls are established by SIP protocol between Asterisk PBX. There will be no security mechanisms in this scenario. Each call has its own signalling and voice channel, unlike IAX which allow the multiplexing of calls.

4.3.2 SRTP trunk

Initially developed to secure voice on an end-to-end basis, SRTP provide encryption to RTP traffic. A SIP trunk can be secured using SRTP between two PBX. The configuration is the same as a simple SIP trunk, apart from the attributes in the signalling channel. Encryption keys are carried by SIP. Asterisk has native support for SDES only, but a patch can be applied for ZRTP (this aspect is developed in next chapter). The key will be passed by SIP, that will need to be secured to pass the encryption key in a secure manner. SIP channel will be protected by TLS: however, TLS uses UDP to carry packets, and SIP must send packets in a reliable method. TLS has been adapted to operate using TCP, providing reliable connection. Figure 16 Resumes the SIP over TLS scenario.

3 Asterisk Now and FreePBX have a web interface but configuration is limited to basic features.

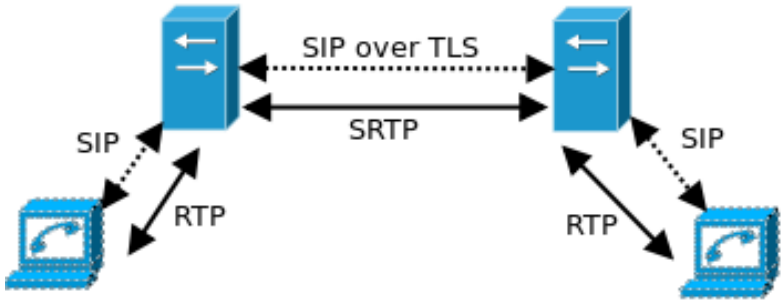


Figure 16: SIP over TLS and SRTP

4.3.3 IAX2 channel

Initially developed to interconnect Asterisk servers, the IAX2 protocol will be used to interconnect both sites. A single channel will be created between the two Asterisk servers to exchange control and voice data, using a single connection to carry all traffic. The IAX2 channel will be in a first time tested without encryption, and then security features will be enabled. The clients, within the domains, are not aware of IAX2 protocol, and are still connected to their local PBX using SIP.

4.3.4 Dial plan and VoIP topology

In any telephony system, it is important to use a consistent dial plan. When using PSTN, the dial plan is hierarchical: every country has a prefix number (i.e. 44 for U.K.). Different geographical area within the country have a defined prefix (i.e. 0131 for Edinburgh). This hierarchical structure allow a more efficient distribution of numbers, and better routing of calls.

The scenario in this study will consist in the interconnection of two sites, where Asterisk servers will have to route calls, whether within the local domain or transfer the calls to the other domain. Even if the routing in this study will not be complex (only two sites), the dial plan should follow a logical and consistent structure.

4.4 IPSec tunnelling

To implement security between two end-points, IPSec tunnelling is a secure and reliable solution. A part of the scenario presented in this study will use IPSec to secure the link between the two domains. The tunnel will secure only the traffic flowing from PBX to PBX. The Asterisk servers will perform the encryption and authentication of the traffic.

Because the traffic will be encrypted from machine to machine, IPSec will work in transport mode, where only the payload is encrypted and authenticated. Tunnel mode cannot be used in this case, as the PBX will not act as a gateway between two networks. Transport mode has not been evaluated in previous studies, where IPSec VPN were configured in tunnel mode, directly on the networking equipments.

4.5 Voice traffic

The review of previous studies showed that most of the researches made measurements using a fixed number of calls. The network, in some case, has been stressed, to measure the impact on the VoIP performances (Guillen, 2009). This study will measure VoIP performance in a wider way: the number of concurrent calls will be variable, to allow a more realistic implementation.

A study about Asterisk performance showed that 130 calls can be handled by the server before reaching 100% CPU, and have a significant impact on the VoIP quality of service (Ahmed, 2008). In this study, the network will not be stressed apart from the voice traffic generated by the calls. Simulating large number of calls should stimulate enough the network to have an impact on the CPU and quality of VoIP.

The implementation will simulate from one to the maximum number of calls possible. A SIP simulator will be presented in the next part, to allow the establishment of large number of calls, with voice traffic flowing across the server.

4.6 QoS measurements

Voice quality will be measured in all scenarios presented above. The quality of voice is affected by three main factors, presented earlier in the study: echo, jitter and delay. The echo cannot be measured in a network point of view, as the echo will be carried by the media flux in the same channel as the speaker's voice. The jitter is measured by capturing the packets as close as possible from the destination: the variation of delay between packets arrival is ideally equal to 0.

Ideally, delay is measured on an end-to-end basis, where the time between the speaker's mouth and the callee ear is measured. However, as presented in the related work section, this measurement requires complex mechanisms, where no documentation and sufficient solutions were found.

For this study, delay measurement will consist in measuring the time between packets arrival. Ideally of 20ms, the extra time taken will be an indication of delay. WAN (slow) links are not implemented: network equipments processing time is very short (no QoS queuing system or congestion). Any delay can only be the result of servers, protocols and encryption methods processing the data flow. Measuring delay and jitter should occur as near as possible from the destination, when all packets are processed by network equipments and transport protocols.

4.7 Server performance analysis

The PBX servers will have to be monitored to retrieve their CPU usage. However, to retrieve more accurate results, the normal system usage must not affect the measure. The choice of Ubuntu server edition, with no graphical interface, limits to the strict minimum the system usage (no graphical processes and limited services running in the background). Asterisk PBX and encryption processes (TLS and IPSec) must be measured apart from the OS processes.

A UNIX command allow to measure in real-time CPU usage and other useful information about the machine. The *vmstat* command[4] displays into the terminal the CPU performance in two distinct fields: the *sys* column shows the system usage and the *us* column displays the percentage of CPU used by the user. The last column, us, will permit to measure the percentage of CPU used by Asterisk and other encryption processes.

4.8 Conclusion

This part presented the scenarios that will be tested in the study. The network scenario implements two PBX to allow the interconnection of two distant sites. Standard security algorithms (IPSec) and recent protocols (SRTP and IAX) will be tested in comparison to non secured protocols, currently widely used. Number of calls will vary, to measure the impact on the VoIP quality of service and the server resource requirements.

The methodology presented to measure VoIP quality of service is similar to the studies presented during the literature review. However, results may be different, as the previous researches did not implement a scenario using two PBX, and number of calls did not vary. The CPU measurement will take in consideration the encryption mechanisms, as the PBX will be in charge of authentication and encryption of voice.

4 *vmstat* command is available in the package *sysstat* (*apt-get install sysstat*)

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

The previous chapter of the study presented the scenarios and methodology for measuring the performance cost when using encryption on VoIP. This part will now present in details the tools and configuration that permitted to realise the designed tests. More technical information will be provided, and parts of configurations file used during the tests will be presented.

The same plan than the design section will be used: first the network set-up will be presented, and then the PBX software and configuration to create the different types of trunk links will be described. IPSec security, which is not part of the PBX software configuration, will be presented in details. Then, SIPp, a stressing tool used to generate VoIP calls, will be presented as well as its configuration, to simulate calls across the test network.

Finally, tools used to measure VoIP quality of service and CPU performance will be presented, and method of collection and processing of results will be described.

5.2 Network scenario

The scenario described in the design part consist in two local domains interconnected by Internet. The two domains will use private netmasks: 172.16.10.0/24 for domain1 and 172.16.20.0/24 for domains2. The domains are connected directly within a subnet of 192.168.10.0/24. Both domains are routed using EIGRP routing protocol.

Network QoS is not implemented in the scenario, as the study focuses only on performances. All links are FastEthernet (100MB/s connections). This unusual solution to simulate an ISP connection has been chosen to limit at maximum packet loss, delay and jitter. The networking equipments used for this study are Cisco devices. Two routers allow the interconnection and routing between the two sites, while access switches Catalyst 2950 allow the communication within the broadcast domain.

The switches have a port configured using SPAN (Switched Port Analyser) to allow a computer to monitor all the traffic within the domain without interfere on network performances. The network measures will be taken from this port. Further details will be presented later in the study.

5.3 Asterisk PBX

Asterisk has been chosen as the PBX software. Both Asterisk PBX are in both side a dedicated machine: the SIP clients and PBX could have been hosted on the same machine, but as CPU performance will be measured, this configuration could result in inaccurate results. Machines hosting Asterisk are desktop computers, 3,2 GHz CPU and 512 MB of memory. Using Ubuntu server on the host machine present two main advantages: there is no graphical interface that could consume CPU resources, and the server edition is optimised for performances (minimum number of services are installed by default).

The Asterisk version used for tests was the latest stable, version 1.6, compiled from the source. For SRTP scenario, both Asterisk version 1.8beta2 and SVN trunk version were tested. The problems encountered will be described later. Once Asterisk was successfully compiled and installed, a script had to be used to allow large number of connection. By default, the OS limits the number of files simultaneously open by a process. Above seventy simultaneous calls, Asterisk started to display errors due to number of opened file limit exceed. To overcome this limitation, a start-up script has been used to launch Asterisk process:

```
#!/bin/sh
ulimit -n 65536
asterisk -vvv &
```

This simple script overwrite the limit of allowed opened files from 1024 (by default) to 65536. Then Asterisk process is calls with the *-vvv* argument, to set the verbosity to 3 (for debugging information).

5.3.1 SIP Users

The SIP users are defined in the Asterisk SIP configuration file: */etc/asterisk/sip.conf* contains the list of SIP users and optional information, like host location and preferred codecs. For the study, a user called *sipp* is declared in the *domain 2* and another user called *sippuas* is declared in *domain 1*. These users do not need to register (in our testing scenario) and are allowed to receive and place calls. Because the media sent by the users (*sipp* and *sippuas*) will have to flow through the PBX, the optional canreinvite option should be set. By default RTP traffic is sent from users to users, Asterisk sends a re-INVITE message to modify IP and ports used to carry RTP flow. This concept is presented in Figure 17.

For the needs of this study, the RTP traffic must flow through Asterisk PBX to use the trunk link between the domains. Both users, *sipp* and *sippuas* are configured with the option *canreinvite=no*. A sample of the sip.conf file from *domain 2* is shown in Figure 18. By defining the local users in the configuration, Asterisk acts as a registrar server. Asterisk PBX knows what users are connected and online. To place local and inter-domain calls, Asterisk must define a dial-plan. This feature is described in the next section.

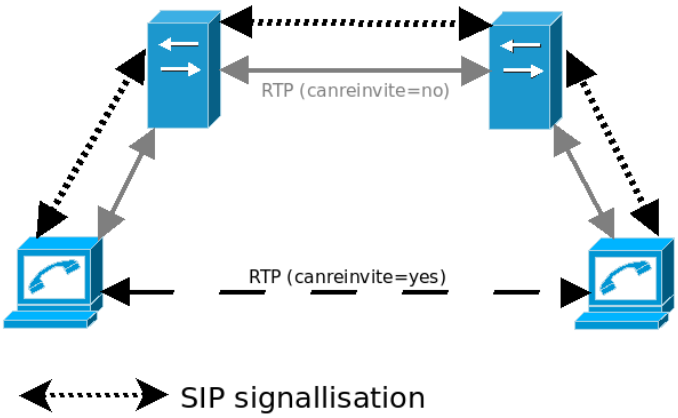


Figure 17: RTP path when using canreinvite option

```
[sipp] ;definition of sipp as a local user
type=friend             ;can place & receive calls
host=172.16.20.1        ;location of host (do not nee
d to register)
user=sipp               ;username
canreinvite=no          ;this PBX will stay in the
media path
```

Figure 18: SIPp user configuration

5.3.2 Dial-plan

The dial-plan allow Asterisk to manage calls: this defines PBX behaviour when clients try to place a call. This allow, for example, to retrieve voice-mail when calling a defined extension, or notify user of eventual congestion and the impossibility to place a call.

In this study, very simple dial-plan is needed: phones numbers consist in four digits, on two domains only. The *domain 1* uses extensions beginning by 1 (from 1000 to 1999) and *domain 2* uses numbers from 2000 to 2999. The dial-plan must be configured in Asterisk to allow call routing. This is done using the */etc/asterisk/extensions.conf*. Figure 19 represents a part of the dial-plan used in this study. The complete configuration can be retrieved in the appendices.

```
[default]
;the following lines defines the behaviour when
calling ext. 1500
exten => 1500,1,Answer              ;Answer the line
exten => 1500,2,Playback(welcome)   ;Play a welcome
message
exten => 1500,3,Hangup()            ;hangup the line
;to allow calling the local extensions
exten => _1XXX,1,Dial(SIP/${EXTEN},20,r)
exten => _1XXX,2,Congestion
```

Figure 19: Dial-plan configuration

This part of configuration allow the PBX server *asterisk1* (domain A) to transfer calls to extensions 2000 to 2999 to the peer *asterisk2*. These calls will be transferred to the domain B (*asterisk2*) using the IAX channel. Then if will be in charge of asterisk2 to route the call. Note that the configuration is different when using SIP trunk (described below).

It is important to have many actions when defining the dial-plan: if the call cannot be established, the caller will be notified by the function *Congestion*. Some other actions can be defined, like playing a custom song or redirect on the callee's voice-mail.

5.4 SIP trunk

The connection between the two domains is first realised thanks to a standard SIP trunk: the signalling and RTP traffic will flow throughout the PBX. This scenario is the easiest to implement, as no security mechanism are set-up. Both PBX will register with each other to provide allow the signalling to use the trunk link. The configuration occurs in the configuration file *sip.conf*. To reload configuration from Asterisk, the command *sip reload* must be issued in the terminal administering the PBX.

```
;this pbx will register with asterisk1 to establish a
trunk
register => asterisk2:mypassword@172.16.10.51

;asterisk1 is defined here to allow its registration
onto this PBX
[asterisk1]
type=friend                ;can place and receive
calls
canreinvite=no             ;RTP will flow through
this PBX
username=asterisk1
secret=mypassword
host=172.16.10.51          ;IP address of remote
host
```

Figure 20: SIP configuration (sample)

This configuration allows the PBX from *domain 1* to register with the PBX located in *domain 2*. To establish the connection, no other configuration is needed. However, the dial-plan should specify the route to redirect inter-domain calls (Figure 21).

```
;the following line define the behaviour when calling
ext. 2XXX
;use SIP to transfert the call to the other PBX
exten => _2XXX,1,Dial(SIP/${EXTEN}@asterisk2,30,r)
;If the call fails, send a message to the caller:
exten => _2XXX,2,Congestion
```

Figure 21: SIP dialplan (sample)

The calls issue from *domain 1* (extensions from 2000 to 2999) will be redirected to 2XXX@asterisk2. The PBX *asterisk2* is reachable thanks to the connection established in the *sip.conf* file. The calls will be handled by asterisk2, using its dial-plan, that should specify how to route the call (in this case, *2XXX* is a local user, so it should call it directly). Note: for more scalability, regular expressions are used (*_2XXX* represent all extensions from 2000 to 2999 and the variable *${EXTEN}* contains the dialled extension).

5.5 SRTP trunk

Although interconnecting two Asterisk PBX using SRTP is in theory possible, Asterisk does not fully support SRTP yet. The latest version of Asterisk, at this time version 1.8beta2 and the developer version (SVN) have SRTP features

(SIP over TLS + SRTP). Unfortunately the lack of documentation and the experimental versions of Asterisk did not allow to successfully set-up this scenario. Despite of numerous tries with different versions of Asterisk (1.8beta2, SVN Trunk and SRTP group[5]) and many emails exchange with developers, it was impossible to successfully set-up STRP between Asterisk PBX. The investigation of this solution had to be abandoned in this study.

5.6 IAX trunk

The evaluation of IAX performance will be performed by creating a trunk link between both Asterisk servers. The configuration of IAX is set thanks to the */etc/asterisk/iax2.conf* file. Both Asterisk servers register with each other, using user-names and passwords set manually in the configuration file. A sample of the configuration is provided in Figure 22.

The trunk option allow to multiplex all voice channel into the single IAX channel. Signalling will be sent using long frames (sent reliably) and voice payload will be carried by short frames.

For the study, the authentication method used is MD5. Although this is not the most secure (RSA is a more secure option), this setting will not affect the performance when using encryption in trunk links. Authentication uses only a few packets when PBX are registering. Then the trunk is set up and channel is encrypted. Authentication mechanisms is no longer used once the secure channel is established (Spencer, 2010).

The dial-plan has to be modified to route inter-domain calls through the IAX trunk link. The following sample of extensions.conf shows how calls from *domain 1* to *domain 2* are routed:

```
;use IAX to transfert the call to the other PBX
exten => _2XXX,1,Dial(IAX/asterisk2/${EXTEN},30,r)
exten => _2XXX,2,Congestion ;if the call fails
```

5 SVN SRTP version of Asterisk can be retrieved from http://svn.digium.com/svn/asterisk/team/group/srtp and http://svn.digium.com/svn/asterisk/team/oej/securertp-trunk (last accessed on 2nd of June, 2010)

```
;register this PBX (asterisk1) @ asterisk2
(user:pass@IP)
register => asterisk1:mysecret@172.16.20.51

;definition of the user asterisk2 to allow its
registration
[asterisk2]
type=friend             ;user can place & receive
calls
host=dynamic            ;host IP address
trunk=yes               ;allow multiplexing voice data
auth=md5                ;authentication method
user=asterisk2
secret=mysecret
encryption=yes          ;force or not encryption
```

Figure 22: IAX trunk configuration (sample)

5.7 IPSec tunnel

A popular software to implement IPSec under Linux system can be found in the *ipsec-tools* package. The security associations, including Diffie-Hellman protocol need to be retrieved from the *raccoon* package. The IPSec encrypts only the traffic between the two PBX (traffic between 172.16.10.51 and 172.16.20.51). The traffic is defined in the */etc/ipsec.conf*, thanks to the following configuration lines:

```
spdadd 172.16.10.51/32 172.16.20.51/32 any -P out
ipsec esp/transport;
spdadd 172.16.20.51/32 172.16.10.51/32 any -P in
ipsec esp/transport;
```

The security association used is Diffie-Hellman, and payload encryption is configured to use the same algorithm as IAX2 and SRTP: AES128. The operating mode chosen is transport. Configuration is located in the file */etc/raccoon/racoon.conf*.

5.8 SIPp: generating calls

SIPp is a free and open-source tool developed to emulate and test SIP. It can generate SIP packets using different scenarios: an instance of SIPp can simulate a user-agent placing a call, while another instance on another computer can simulate the callee. SIPp allow the emulation of large number of calls, and the tuning of SIP packets.

SIPp can simulate scenarios, based on customisable XML files, allowing the simulation of simple to complex scenarios. In the scope of the study, SIPp will be used to simulate calls from the *domain 1* to the *domain 2*. A custom scenario has been set-up to provide such behaviour. The set-up consist in an instance of SIPp in the *domain 1* calling another instance of SIPp in the *domain 2*. The scenario of the caller, located in *domain 1* is described by Figure 23.

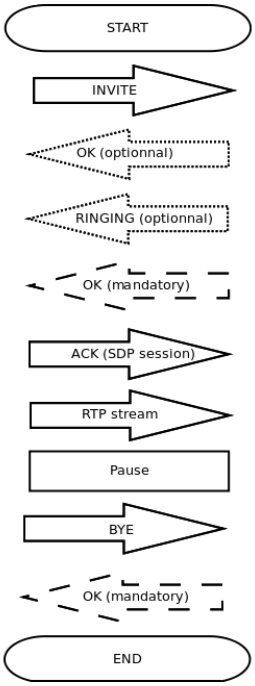


Figure 23: SIPp client scenario

To set-up successfully a call, the callee instance of SIPp, located in the *domain 2*, should be able to respond to SIP messages and so allow the voice data to flow through between the domains. The scenario is simpler than the client, and consist uniquely in sending OK messages ACKnowledge the packets to establish the communication and terminate the session.

For the study, a pcap file containing RTP traffic has been recorded and is played by SIPp to generate the RTP traffic. This file contains 13 seconds of music, using g.711U codec. Both Asterisk, caller and callee are configured to support this codec, to prevent the need of transcoding that would increase the CPU usage of the servers. The scenarios are saved as an XML file (available in appendices). SIPp is started with the following command:

```
sipp -sf client-pcap.xml -s 1111 -p 5060 -l 10
```

The *-s* parameter specify the extension that will be called, *-p* set the listening port for incoming packets to 5060 (default port for SIP) and *-l* parameter sets the maximum number of simultaneous calls. To produce two way

communication, the callee is configured to echo reply the received RTP traffic. This action is performed thanks to the *-rtp_echo* argument.

5.9 Voice QoS measurements

Quality measurement occurs as close as possible to the destination, when packets have been fully processed. Ideally, the measurement should be performed on the callee's computer. However, as the system will be stressed by a large amount of calls, using SIPp, the CPU usage of the machine hosting the user can affect the capture results. Capturing large amount of packets will require lot of resources from the capture device.

To retrieve more accurate results, a dedicated machine has been set-up to capture all traffic within a domain. This machine has been connected on the SPAN port of Cisco switches, and can only receive traffic, in order to not perturb the traffic during the tests.

The software used to capture packet is Wireshark, a popular traffic sniffer and analyser. Packet capture is running during all the test period, where all the packets are saved into a *pcap* file. Statistics of all streams are then exported into a csv[6] file, and then statistics are calculated thanks to a spreadsheet software[7]. Figure 24 shows a graphical representation of a call made from *domain 2* to *domain 1* in the IAX2 encrypted trunk scenario. The traffic between PBX cannot be decoded by Wireshark, due to encryption mechanism turned on. However, traffic within the SIP domain is sent in clear, RTP traffic can be wire-tapped and analysed, for the need of the study.

To retrieve more accurate results, all the RTP streams have been recorded, regardless of the number of calls simulated (packet capture files reached 600MB when 100 calls were simulated). The QoS parameters (max delta, max jitter and mean jitter) of all streams have been retained for the study. Results will be presented in the next part.

6 Comma-Separated Values is a simple text format for a database table (wikipedia)
7 OpenOffice Calc was used in this study

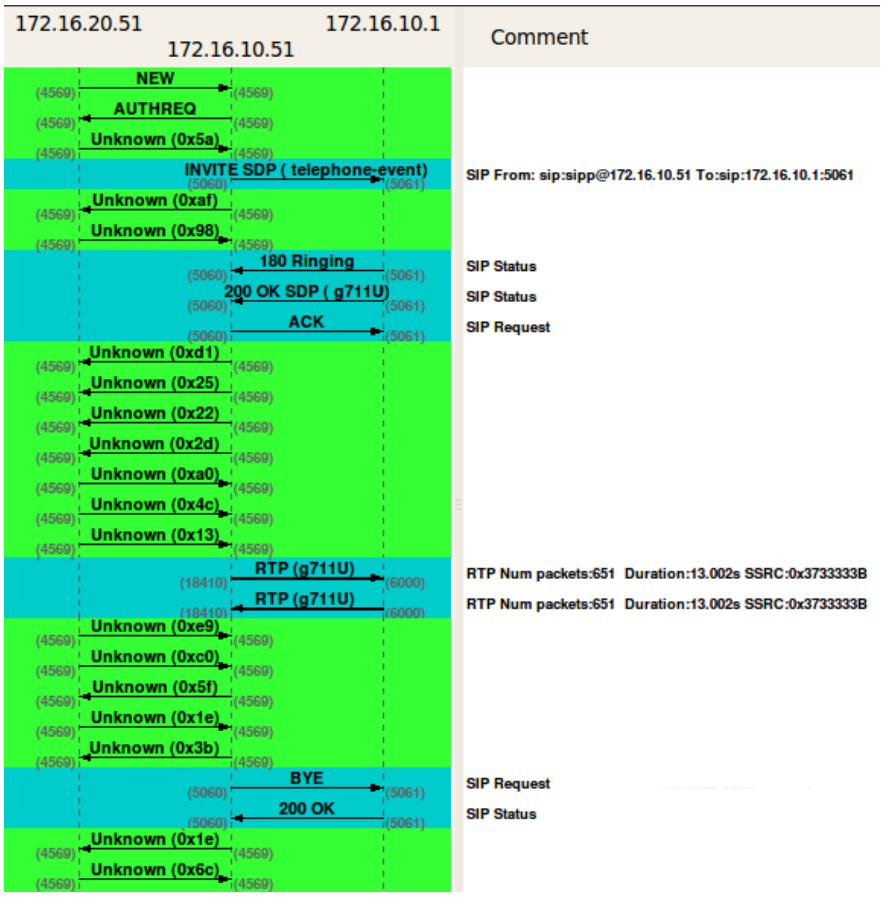


Figure 24: IAX2 encrypted trunk

5.10 CPU performance

Measuring the performance was achieved using the *vmstats* command, presented in the design section. For the needs of the study, informations displayed by the command *vmstats* needed to be computed into a more readable format. The command *vmstats -n 1 > file.csv* allows to save output display into a csv file, with a poll interval of one second. This command is running the time of the test (approximatively 30 seconds).

The output file can be opened by any spreadsheet software (OpenOffice was used in this study) to import the data and create graphs.

5.11 Conclusion

The implementation has been achieved, with respect of the specifications presented in the design chapter. More details were provided in this chapter, regarding the measurement methodology and software configurations.

The tests have been performed successfully, except for the SRTP trunk, where, unfortunately, the very recent development of this solution did not come with any documentation. Further the point made of, no help could have been found from the developers of Asterisk.

In every scenarios, the tests consisted in thirty seconds of calls emulation for each number of calls assessed. The simulation of more than hundred calls generated some errors on the PBX, and results were inaccurate and erroneous. However, CPU measurement could be performed until 130 calls, regardless of the errors displayed by the Asterisk PBX software. The results retrieved during the test will be presented and analysed in the next chapter.

CHAPTER 6: EVALUATION

6.1 Introduction

This part will present the results of the tests performed, following the guidelines described in the Design and methodology chapter. All scenarios presented earlier have been successfully set-up and measurements will allow to evaluate both the VoIP quality of service and the CPU usage of the PBX. The results, originally retrieved in csv files, have been computed to make them available as graphics.

The first section of this chapter will present the quality of service measurements, to demonstrate how the voice quality is affected by encryption mechanisms. The second part will present the performance cost of secure applications on PBX, by presenting the CPU usage of different protocols in the same conditions.

6.2 Latency measurements

Initially, delay was measured by calculating the mean value of all RTP packets for a fixed number of calls. As mean values are meant to hide differences, this method did not show any accurate results, where the delay values were all around 20ms, at 0.01% precision. This aspect has been developed earlier, in the Literature review chapter. The delays presented in the study will be the maximum latency values of sixty randomly chosen calls. The cloud formed by the lines presents more accurate results than a calculation of average latency values.

Figure 25 presents the latency obtained when SIP trunk and SIAX2 were in use. As expected, using encryption (SIAX2) seems to inflict slightly more delay than SIP. The processing of encryption algorithms adds a time overhead to the RTP traffic. However, the difference does not seem to be significant (SIAX2 latency is on average 6ms longer than SIP for 50 calls). For large amount of concurrent calls (100 simultaneous calls), SIAX2, despite of the encryption delays, offers lower latency than SIP. This result is probably explained by the optimisation of IAX2 protocol to handle large number of calls.

The delay measured with IPSec scenario is compared with SIAX2 in Figure 26. It was expected that IPSec inflicts more delays than SIAX2, as IAX2 has been developed for real-time applications. However, it seems that IPSec offers better latency values than SIAX2, for 20, 50 and 70 simultaneous calls. Or very large

number of calls, IPSec latency increases dramatically to reach 138ms, while SIAX2 seems to be able to keep delays under 93ms for maximum values, and under 60ms in average, against 80ms for IPSec.

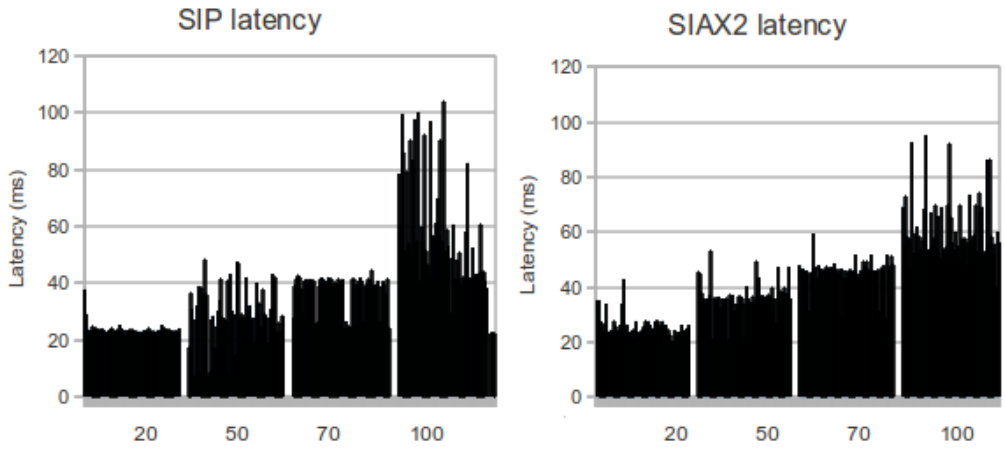


Figure 25: SIP vs SIAX2 latency measurements

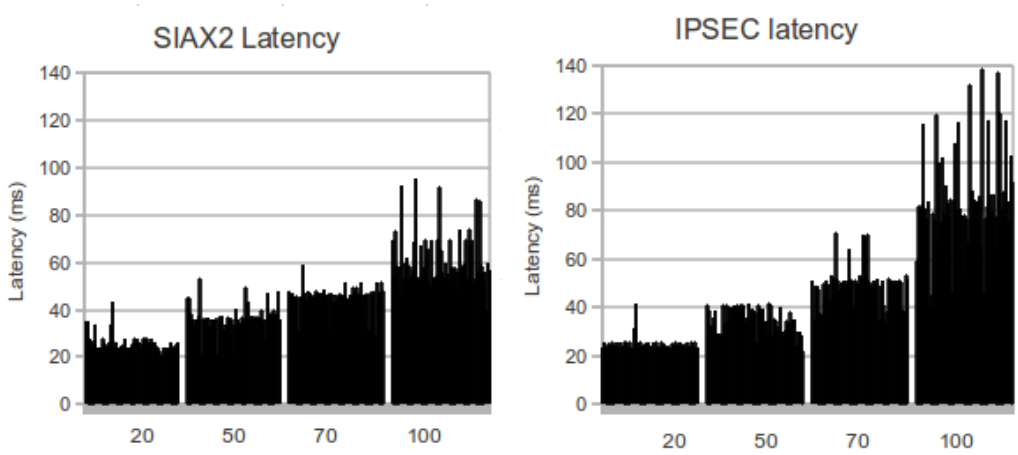


Figure 26: SIAX2 and IPSec latency measurements

6.3 Jitter measurement

The jitter values seems to be the most affected by the different protocols used. Its was expected that encryption systems affects the jitter and degrade voice quality. Figure 27 Shows the results obtained during the tests, for both SIP and SIAX2 protocols. The graph shows more important jitters when SIP is used than for the secure IAX2. However, for low number of calls, jitters values for SIP calls are more regular, where values barely reach 2ms. It can be noted that independently of the number of calls, SIAX2 jitter values increase in a regular basis, while the gap between 70 and 100 calls for SIP is important (increasing of 400%).

The difference of jitter values can be explained by the use of SIAX2 as a trunk mode, where all calls are multiplexed into a single channel. This implementation allows a better management of different media flows, and IAX2 implement mechanisms to reduce jitters (Spencer, 2010). However, the use of encryption as an impact when encryption is used: Figure 28 presents the jitter of IAX2 used in non-secure mode. The jitter reaches 7ms in some cases when encryption is used, while values are never higher than 4ms when no encryption algorithms are in use. This difference does not seem to be significant, as average values are less than 2ms in any cases. The recommendation specify that jitter must be shorter than 30ms to provide good voice QoS.

The results obtained with IPSec are presented in Figure 29. For a low number of concurrent calls (20 calls), jitter values are near from results obtains in other scenarios. For 50 and 70 simultaneous calls, the jitter values present a significant difference than the results obtained with SIP and (S)IAX2. Although values seems more regular, they are significantly higher than results obtained in the SIAX2 scenario: 200% difference is reached when 70 calls were performed during the tests. Despite numerous tests, the values for 100 calls do not seems to be accurate. With a maximum of 500 000 000ms jitter (5.59 days), these values cannot be taken in consideration in this study. Many different calls have been tested, resulting of similar results. The tests have been performed many times, but for every attempt same range of values were retrieved for 100 calls. This could be due to the limitation of Wireshark to capture packets and gather correct information about flows. Unfortunately, the results for 100 calls will not be considered for the rest of the study.

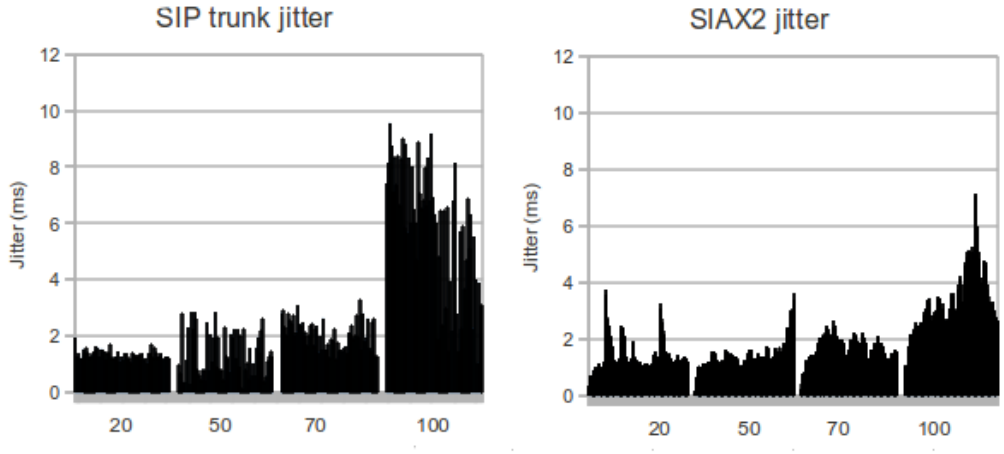


Figure 27: Jitter values for SIP and SIAX2

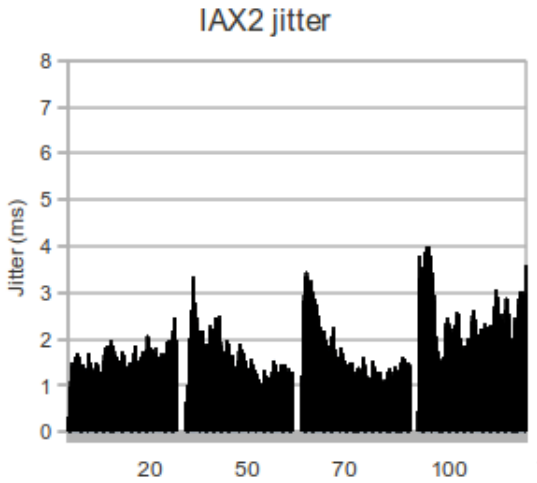


Figure 28: IAX2 jitter (without encryption)

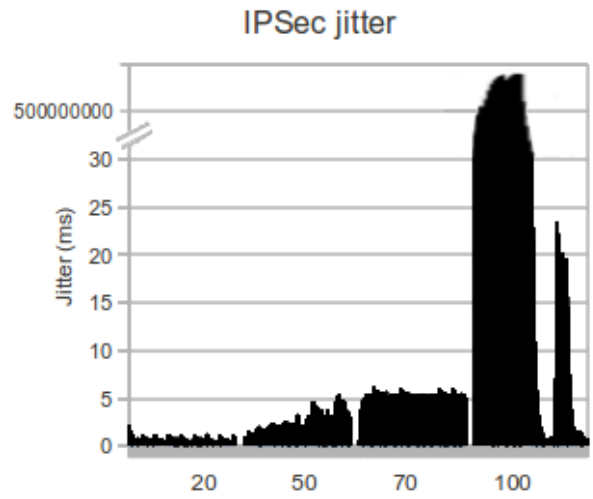


Figure 29: IPSec jitter values

6.4 Packet loss

Packet loss was the third parameter that defined VoIP quality of service. Packet loss was measured in every scenario, as close as possible from the destination. It was expected to retrieve very low values, as less than 1% packet loss is needed to provide good VoIP QoS. The results are presented in Figure 30. It was expected that securing voice traffic had an influence on the voice QoS, but it was not expected that SIAX inflicts more than 1% packet loss. Overall, SIP trunk inflicts less percentage of packet loss than SIAX2, apart for very large number of calls, were 12% packet loss was reached in some calls.

Figure 31 shows percentage of packet loss in opposition to SIAX. As unexpected, packet-loss is around 0.5% in for any number of calls, presenting even better results than SIP scenario. These results confirm some tests (Salama, 2009) presented in the Literature review. The global percentage of packet loss is more important than other studies, where 1% was rarely reached (Salama, 2009 ; Guillen, 2009). The high percentage of packet loss in this study can be explained by the test-bed set-up: network was not stressed by other traffic flows (Perez, 2006), but very large number of VoIP calls were transiting on the network. All packets were concurrent and when queues are full, packets are dropped. The use of QoS model would not have affect the packet loss, as all packets would have the same priority (highest).

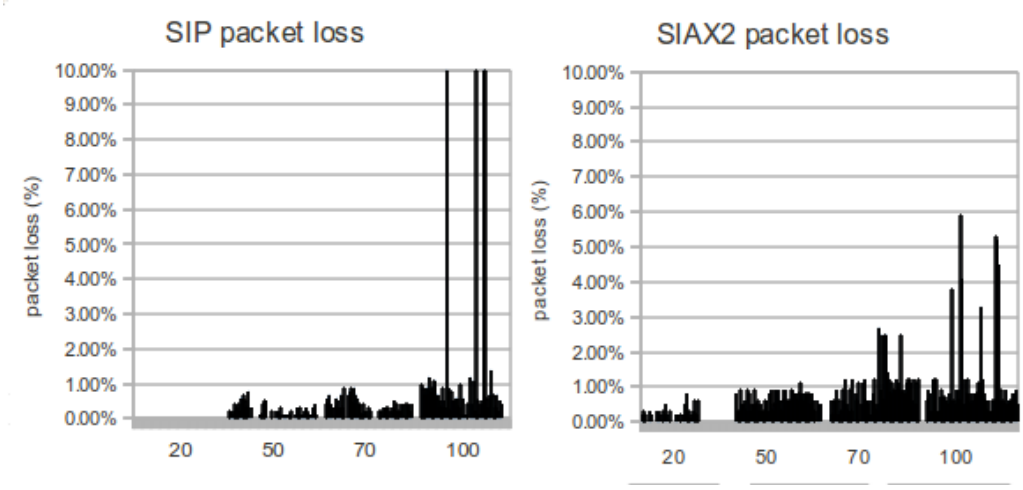


Figure 30: SIP and SIAX2 packet loss

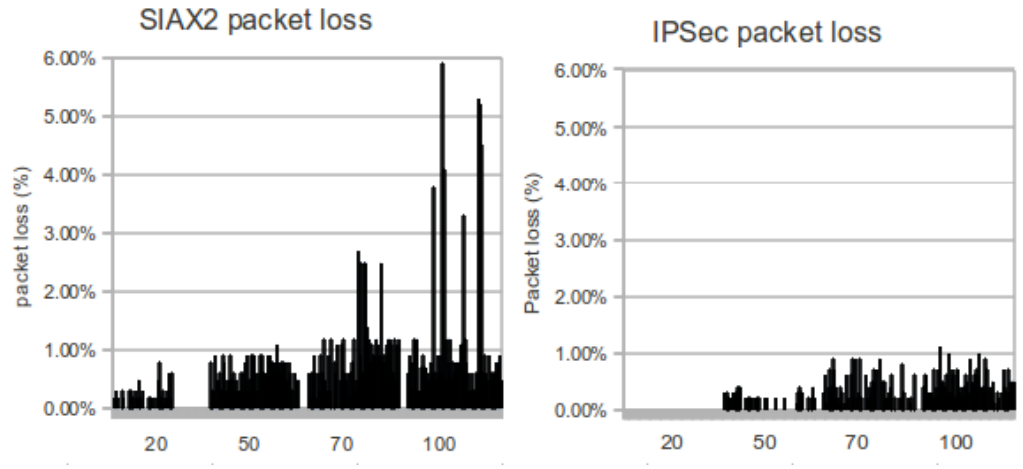


Figure 31: Packet loss in SIAX2 and IPSec scenarios

6.5 Resource usage

The resource usage of PBX servers were measured according the methodology described in Chapter 5. CPU and memory usage were measured, and results are presented in. It was expected that secure protocols (SIAX and IPSec) require more resource than SIP and IAX. IAX, designed to encapsulate many concurrent calls, should require less CPU effort than SIP. SIAX2 should be more optimised than IPSec to encrypt voice data, so should also needs less CPU power.

Figure 32 and Figure 33 show that for any number of call, SIP seems to require less CPU and memory than other protocols. This result was not expected, as SIP trunk treats each voice channel independently, unlike IAX2 that supports trunking. The difference of CPU usage between IAX and SIP can be explained by the processing of encapsulating voice packet into the IAX trunk, and the mechanism to reduce jitter between voice packets. SIP would only forward packets to the interface without any further processing. The memory requirements for IAX2 are higher than SIP probably for the same reason.

When secure algorithms are in operation, CPU power is more required to perform encryption (AES 128 in both cases). Surprisingly, IPSec requires less CPU than SIAX2 for less than 50 calls. In other cases, IPSec needs between 7 and 17% more CPU usage than SIAX2, although same encryption algorithm are used. This difference may not be significant, as 100% CPU usage is not reached. The memory usage present unexpected results: IAX2 and SIAX2 both require more memory than SIP and IPSec. The reason could be the same as for the CPU usage, as IAX2 performs more complex operations to reduce the voice quality loss.

It can be noted that only 30% CPU is reached, although previous studies (Ahmed, 2008) with similar number of calls reached the 100% CPU usage. This difference is due to the framework design: there was no need to use transcoding (codec translation) as all VoIP devices supported the same codec (G.711). The computers used in this study were also more powerful, with 3.2GHz CPU (against 1.5Ghz in the previous study).

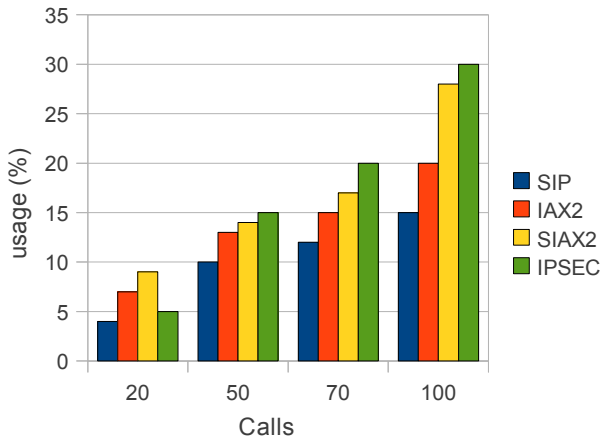


Figure 32: CPU usage

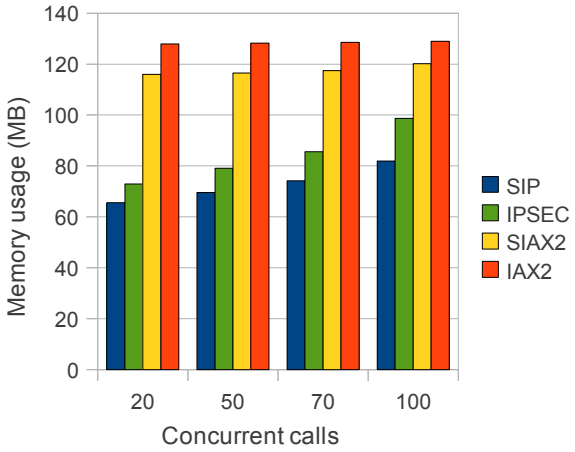


Figure 33: Memory use

6.6 Conclusion

This chapter presented the results obtained during the test phase of the study. A detailed description helped to demonstrate the impact of encryption mechanisms on VoIP performance. In the tests realised, major QoS parameters are not dramatically affected by the encryption: the latency measured was slightly affected by SIAX2 (less than 5% overhead), although out-of-band IPSec added a maximum of 20% overhead. The stronger authentication could be a reason of such difference. The second most important QoS parameter, jitter, was improved with the use of IAX2 protocol. This is probably the result of optimisation mechanisms implemented in the protocols itself, designed to support large number of calls. However, securing the voice streams with IPSec inflicted noticeable latency variation, presenting higher jitter values. This difference should be taken into consideration when deploying VoIP system, as the spotted difference reached 200% in some cases. Unfortunately, results for very high number of calls could not be exploited. Packet loss was also measured, with some surprising results, as IPSec seems to have the least packet loss value. With the help of jitter values, a loss percentage of packet loss for IPSec confirms that out-of-band protocols have more influence on VoIP QoS than in-band protocols, like IAX2.

CHAPTER 7: CONCLUSION

7.1 Objectives

This study had three main objectives, defined in Chapter 1:

- Conduct a literature review that critically present and review the previous studies about VoIP, the security issues and secure solutions. The review of work undertaken by previous research papers, assessing VoIP parameters and different techniques to retrieve results.

- The design of a test-bed framework to allow flexible and accurate tests on a VoIP system. This framework should be able to perform tests and retrieve results to verify findings described by the previous objective. This framework had to be flexible enough to adapt many different secure mechanisms and present accurate results of measurements.

- The implementation of the defined framework, using software and tools available. The tests defined in the previous objective should allow the study to retrieve and analyse them, to confront them to related studies and present results of new tests never presented in the literature review.

The objectives defined above have been met during the conduct of this study. The first objective was successfully conducted, as many papers (Guillen, 2009) and documentation was found on the topic. Main VoIP parameters were identified, and a presentation of the main threat was done. The secure solutions, like IPSec, TLS and SRTP were presented, along with their associated drawbacks. Indeed, the VoIP parameters, particularly latency and jitter, are affected by the use of encryption mechanisms. IAX2 protocol review was not complete, as only a few papers discuss about this emerging protocol. The resource usage of servers was also presented (Ahmed, 2008), when using either IPSec, TLS or Asterisk as a IP PBX. When many researches on the same topic were available, the results were confronted, to allow a review of different measurement techniques used.

The second objective consisted in designing a framework to allow tests. This objective was met thanks to the research conducted for the first objective. Indeed, the literature review allowed to gather informations about test procedures, useful parameters and tests undertaken. Some tests presented in the literature review did not seem complete, where, for example the voice QoS was

measured for a fixed number of calls. The designed framework allowed to create tests in real-scenario, using open-source software and basic network equipment. It also allowed to measure both VoIP QoS parameters and server performances. It was flexible enough to simulate a variable number of calls, using all secure protocols presented in the literature review.

The third objective was the implementation of the framework designed in the second objective. It was successfully implemented using basic computers, as IP PBX and VoIP clients, and basic network equipments. The software used in this study were standardised and open-source: Asterisk, the well-known and established PBX acted as the IP PBX, SIPp, the open-source call emulator was useful to generate from 10 to 100 concurrent calls. The packet sniffer, Wireshark, was helpful to retrieve QoS parameters, while simple bash script running on the IP PBX were retrieving resource usage. The use of Asterisk as IP PBX on a dedicated machine allowed the use of SIP, IAX and SRTP as VoIP protocols. However, the lack of maturity of SRTP presented to be an issue, at it was impossible to set-up this scenario. This aspect will be developed later in this chapter.

7.2 Findings

The main findings of this study were briefly presented in the evaluation chapter, when evaluating the test results. It has been confirmed that adding security to voice inflict an additional overhead to main parameters. One way latency was affected, mostly when IPSec is used (18% overhead for large number of calls). However, when SIAX2 shows better performances, where latency is insignificantly increased in comparison to the scenario with no security. Jitter results were even lower when SIAX2 was used to secure the trunk. Indeed, jitter values were similar (under 2ms) in most cases, but SIAX2 allowed jitter to be reduced of 33% (for 100 concurrent calls) in comparison to the basic SIP trunk. IPSec did not present as good results as SIAX2.

The good performances of SIAX2 have however a cost: packet-loss was more important than the IPSec scenario (average of 1% in some case, which is the limit of acceptable value). The resource usage, also measured, proved that IPSec, despite lower results than SIAX2, required more CPU usage than other protocols. Followed by SIAX2 (17% CPU usage difference), IPSec requires less memory than SIAX2. This difference of resource usage can be significant in some cases, where the server CPU usage is close 100%.

The exceptional performances of IAX2 protocol has a cost in term of resource usage. The new features that perform IAX2 (encryption, trunk mode and jitter buffer) increase considerably the quality of service, for the same security strength as IPSec. The performance cost is not consequent and the additional overhead does not affect dramatically the performance of the PBX.

IAX2 seems to be a good solution to secure voice calls, either for trunk links (large number of calls) than for end-to-end stream. However, IAX2 protocol does not support encryption features on an end-to-end basis (Spencer, 2010). As the protocol is not standardised yet, this feature could be developed in the next years, making IAX2 the first VoIP protocol supporting embedded security features without affecting voice QoS.

7.3 Critical analysis

This study was conducted to evaluate the performance cost of securing voice calls over IP networks. The background presented the most important points that needs to be developed for a good understanding of this research. The literature review was relevant to the topic, where many studies were presented. However, due to the recent development of IAX2, the lack of studies about this protocol could not help to present related work to it. This gap did not allow the study to compare the framework structure and results obtained with other works.

The designed framework was flexible and allowed to test many scenarios, with different protocols and for adjustable number of calls. It was also flexible enough to perform tests using different networking conditions, like implementing slow-links, different routing protocols or congested network, with the help of a traffic generator. QoS models and different routing protocols could be easily implemented in this framework, but these parameters are out of scope of the study.

The implementation allowed to test most of the scenarios described in the Design and methodology chapter. However, the current development of SRTP and ZRTP on the Asterisk platform did not allow to perform the tests with SRTP to secure voice calls. Despite numerous tries, emails and rare help by email, this scenario could not be implemented. The IPSec implementation was not fully functional, as for some unknown reasons, the jitter values for 100 simultaneous calls were inaccurate. At this time, no explanation was found, as all test parameters seemed correct.

The evaluation of results was presented as graphics, where a sample of 60 calls was represented for the latency and packet loss. This approach allowed a better overview of results, as mean values did not present accurate results. This type of graphics allow the audience to seem all values as a cloud, more representative of the results. The jitter values were also displayed a graphic, but from a randomly chosen call during the phase test. Also, the bandwidth measurement was a parameter that has not been investigated in this study. Measuring this parameter could have help in the evaluation of packet-loss results.

7.4 Further work

The presentation of the critical analysis have brought in light some scenarios that could not be implemented in this study. Further investigation, with more recent versions of both Asterisk, SRTP and ZRTP library could help in implementing successfully those scenarios. The implementation of SRTP scenario would complete this study, as SRTP could not be evaluated and compared to other protocols. Also, bandwidth measurement would complete the resource usage part of this study.

Security in VoIP is a vast topic in full expansion, and IAX2 has presented excellent results in comparison to assessed protocols. IAX2 should gain more attention in the VoIP domain, where SIP seems to be the most deployed protocol. It seems interesting to gain more interest in securing voice on an end-to-end basis, where encryption costs would be in charge of the end-device. Although SRTP was developed to secure on an end-to-end basis, using IAX2 to secure end-to-end connection should be considering, regarding the excellent performances presented in this study.

REFERENCES

Ahmed M. & Mansor A. M. *CPU dimensioning on performance of Asterisk VoIP PBX*. Proceedings of the 11th communications and networking simulation symposium, pp. 139-146, (2008).

Albers J., Hahn B., McGann S., Park S. & Zhu R. *An Analysis of Security Threats and Tools in SIP-Based VoIP Systems*. 2nd workshop on security voice over IP, Cyber Security Alliance, Washington, DC, (Jun. 2005).

Andreasen F., Baugher M. & Wing D. (July 2006). *Session Description Protocol (SDP) Security Descriptions for Media Streams*. Retrieved July 2010 from the IETF database at http://tools.ietf.org/html/rfc4568

Arkko J., Lindholm F., Naslund M., Norrman K. & Carrara E. (Aug. 2004). *RFC3830: MIKEY: Multimedia Internet KEYing.* Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc3830

Arkko J., Carrara E., Lindholm F., Naslund M. & Norrman K., (Jul. 2006). *RFC4567: Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc4567

Baugher M., McGrew D., Naslund M., E. Carrara E. & Norrman K. (Mar. 2004). *RFC3711: The Secure Real-time Transport Protocol (SRTP)*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc3711

Benini M. & Sicari S. *Assessing the risk of intercepting VoIP calls*. Proceedings of the V-IPSI 2007 Venice Conference (Mar. 2008).

Bresciani R. & Butterfield A. *A formal security proof for the ZRTP Protocol* . International Conference for Internet Technology and Secured Transactions, 2009, London, (9-12 Nov. 2009), pp. 1-6.

Berthelot F. *Analysis of security issues with respect to Voice over IP technologies*. Beng Honours project, Napier Edinburgh University, (May 2009).

Carlos T. & Vieira S.. *VoIP as a tool for an effective voice communication cost reduction.* Universitade de Porto, (Feb. 2009).

Chong H. M. & Matthews H. S. *Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems* . In IEEE International Symposium on Electronics and the Environment, (2004), pp. 106-111

Cisco-5125 (2 Feb. 2006). *Document ID5125: Understanding Delay in Packet Voice Networks*. Retrieved June 2010 from www.cisco.com/en/US/tech/tk652/tk698/technologies_whitepaper09186a00800a8993.shtml

Cisco-5244 (20 Jul. 2006). *Understanding H.323 Gatekeepers*. Retrieved July 2010 from http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00800c5e0d.shtml

Coarfa C. & Wallach D. (2006). *Performance Analysis of TLSWeb Servers*. , Network and Distributed Systems Security Symposium ’02, San Diego, California, (Feb. 2002).

Collier M. (2005). *VoIP Vulnerabilities – Registration Hijacking*. SecureLogix Corporation, VoIP magasine, (Jan. 2005).

Detken K. O. & Eren E.*VoIP Security regarding the Open Source Software Asterisk*. Customer Workshop "Flexible open-source solutions" to the DECOIT, Bremen, (2008).

Diab W. B., Tohme S. & Bassil C. *VPN Analysis and New Perspective for Securing Voice over VPN Networks* . In Fourth International Conference on Networking and Services (ICNS 2008), pp. 73–78.

Doraswamy, N. & Harkins D. *IPSec: the new security standard for the Internet, intranets, and virtual Private Networks*. Prentice Hall PTR Internet Infrastructure Series, New York, (2003), pp. 185-190.

Ferrante A., Piuri V. & Owen J. *IPSec Hardware Resource Requirements Evaluation*. E-business and Telecommunications: 4th International Conference, ICETE 2007, Barcelona, Spain, (28-31 Jul. 2007), Revised Selected Papers, Volume 2007.

Guillen E. P. & Chacon D. A. *VoIP Networks Performance Analysis with Encryption Systems*. World Academy of Science, Engineering and Technology 58, (2009).

Gupta P. & Shmatikov V. *Security Analysis of Voice-over-IP Protocols* . In Computer Security Foundations Symposium, (2007). CSF ’07. 20th IEEE, pp. 49–63.

IUT-T [G.114] (May 2003). *Recommendation G.114 - One-way Transmission Time.* Retrieved June 2010 from www.itu.int/itudoc/itu-t/aap/sg12aap/history/g.114/g114.html

Koren T., Casner S., Geevarghese J., Thompson B. & Ruddy P. (Jul. 2003). *Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc5456

Labovitz C. & Nazario J. *Worldwide Infrastructure Security Report.* Annual Security report, Volume V., (12 Jan. 2010). Retrieved June 2010 from www.arbornetworks.com/report

Li F. & Thottan M. *End-to-end Service Quality Measurement Using Source-routed Probes*. 5th Annual IEEE Conference on Computer Communications (INFOCOM), Barcelona, Spain (Apr. 2006).

Min H., Chong & Matthews H. S. *Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems.* Proc. IEEE Int'l Symp. Electronics and the Environment, (May 2004), pp. 106-111

Nassar M., State R. & Festor O. *VoIP Honeypot Architecture* . International Symposium on Integrated Network Management, IM '07. 10th IFIP/IEEE, (25 Jun. 2007), pp. 109-118.

Ormazabal G. *Secure SIP: A scalable prevention mechanism for DoS attacks on SIP based VoIP systems*. Proceeding the 2nd international conference on Principles; System and Applications of IP Telecommunication (IPComm), (Jul. 2008), pp. 107-132.

Jackson K. *VoIP Abuse Project Blacklists Attackers.* Dark Reading, Security Center, Insiders, (29 Sep. 2010). Retrieved October 2010 from http://www.darkreading.com/insiderthreat/security/attacks/showArticle.jhtml?articleID=227500994

Pauli D. *Thousands lost in rising VoIP attacks*. ZDNet Security Magasine, (8 Oct. 2010). Retrieved October 2010 from http://www.zdnet.com.au/thousands-lost-in-rising-voip-attacks-339306478.htm

Pérez J. A., Zárate V., Montes A. & García C. *Quality of Service Analysis of IPSec VPNs for Voice and Video Traffic.* in Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on, (19- 25 Feb.2006), pp. 43–43.

Rix A. W., Beerends J. G, Hollier M. P. & Hekstra A. P. (2001). *Perceptual evaluation of speech quality (PESQ) – A new method for speech quality assessment of telephone networks and codecs.* Proc. ICASSP '01, Salt Lake City, USA, (2001), pp. 749–752.

Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J.,Sparks R., Handley M. & Schooler E. (June 2002). *RFC3261: SIP: Session Initiation Protocol.* Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc3261

Salama G. I., Elemam Shehab M., Hafez A. A. & Zaki M. (2009). *Performance Analysis of Transmitting Voice over Communication Links Implementing Ipsec*. 13th International Conference on Aerospace Science & Aviation Technology, ASAT-13, (26-28 May 2009).

Schulzrinne H., Casner S., Frederick R. and Jacobson V. (Jul. 2003). *RFC3550: RTP: A Transport Protocol for Real-Time Applications*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc3550

Steffen A., Kaufmann D. & Stricker A. *SIP Security*. E-Science and Grid, Ad-hoc network, Verlag, (2004), pp. 397-410.

Spencer M., Capouch B., Guy E. & Miller F. (Feb. 2010). *IAX: Inter-Asterisk eXchange Version 2*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/rfc5456

Thermos P. & Takanen A. *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures.* Addison-Wesley Professional, (2007).

Tschofenig H. & E. Rescorla (25 Feb. 2006). *Real-Time Transport Protocol (RTP) over Datagram Transport Layer Security (DTLS) [draft-tschofenig-avt-rtp-dtls-00.txt]*. Retrieved June 2010 from the IETF database at http://tools.ietf.org/html/draft-tschofenig-avt-rtp-dtls-00

Voznak M. *Speech bandwith requirements in IPsec and TLS environment*. 13th WSEAS International Conference on Computers, (23-25 Jul. 2009) Rhodes, Greece.

Wieser C., Laakso M. & Schulzrinne H. (2003). *Security testing of SIP implementations*. Columbia Univ. http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2003/cucs-024-03.pdf

Wowra J. P. (2007). *RTP over Datagram TLS*. http://www.informatik.uni-goettingen.de/Filepool/Theses/gaug-zfi-bm-2007-28.pdf

Völker L., Schöller, M. and Zitterbart M. *Introducing QoS mechanisms into the IPsec packet processing.* Proc. 32nd IEEE Conference on Local Computer Networks LCN 2007, (15–18 Oct. 2007), pp. 360–367.

Zimmermann P. (2010). *The Zfone™ Project*. Retrieved May 2010 from http://zfoneproject.com/prod_zfone.html

Zimmermann P., Johnston A., Avaya Ed. & Callas J. (May 21, 2010). *ZRTP: Media Path Key Agreement for Unicast Secure RTP [draft-zimmermann-avt-zrtp-21]*. Retrieved May 2010 from http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-21

VOIP QOS MEASUREMENTS

This section presents screen-shots when measuring VoIP parameters.

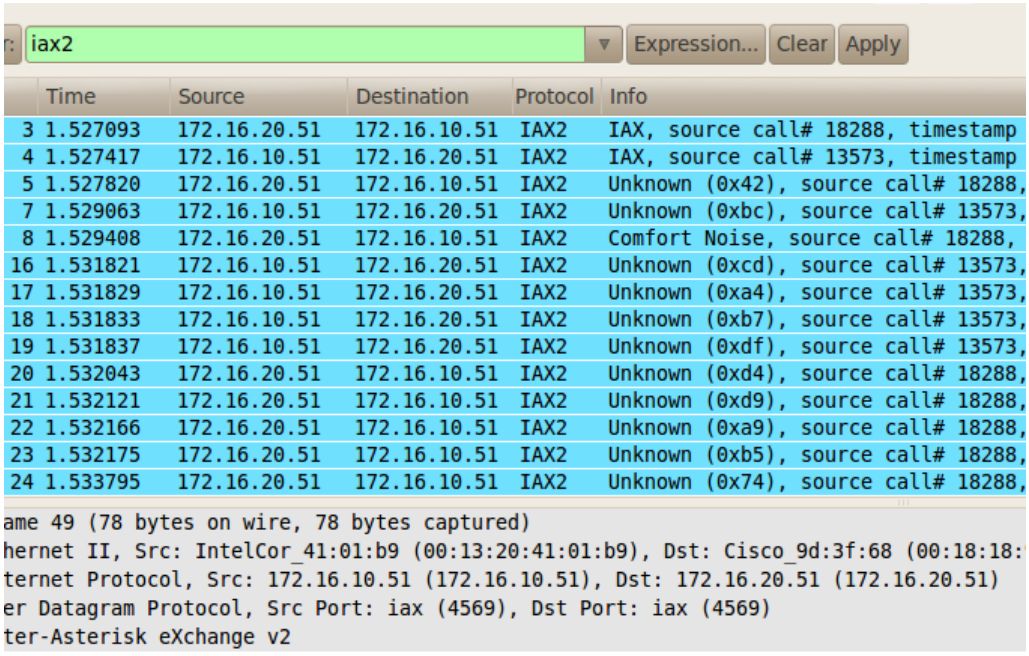


Figure 34: SIAX2 call set-up between Asterisk PBX

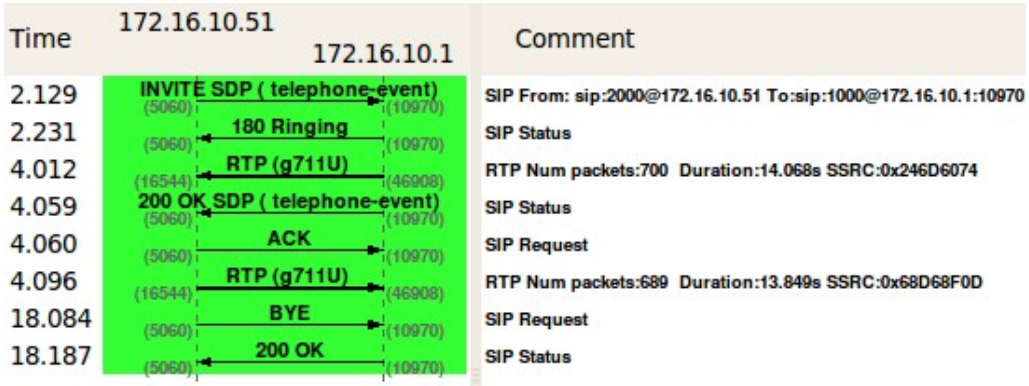


Illustration 35: SIP call between PBX and user (SIPp client)

No.	Time	Source	Destination	Protocol	Info
16798	7.197876	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16799	7.198145	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16800	7.198237	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16801	7.198757	172.16.10.1	172.16.10.51	RTP	PT=ITU-T G.711 PCMA, SSRC=0x
16802	7.198859	172.16.10.1	172.16.10.51	RTP	PT=ITU-T G.711 PCMA, SSRC=0x
16803	7.198866	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16804	7.199012	172.16.10.51	172.16.20.51	ESP	ESP (SPI=0x0a36c7db)
16805	7.199019	172.16.10.51	172.16.10.1	RTP	PT=ITU-T G.711 PCMA, SSRC=0x
16806	7.199370	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16807	7.199391	172.16.20.51	172.16.10.51	ESP	ESP (SPI=0x0ae1929f)
16808	7.200699	172.16.10.51	172.16.10.1	RTP	PT=ITU-T G.711 PCMA, SSRC=0x
16809	7.200807	172.16.10.51	172.16.20.51	ESP	ESP (SPI=0x0a36c7db)

```
+ Frame 16812 (246 bytes on wire, 246 bytes captured)
+ Ethernet II, Src: Cisco_9d:3f:68 (00:18:18:9d:3f:68), Dst: IntelCor_41:01:b9 (
+ Internet Protocol, Src: 172.16.20.51 (172.16.20.51), Dst: 172.16.10.51 (172.16
- Encapsulating Security Payload
    ESP SPI: 0x0ae1929f
    ESP Sequence: 1466008
```

Figure 36: IPSEC ESP and RTP traffic

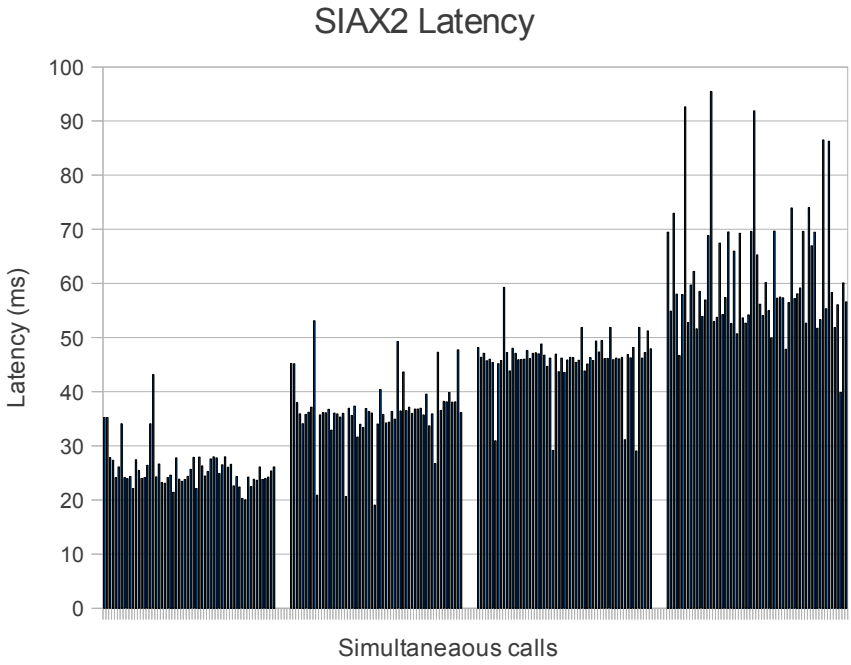


Figure 37: SIAX2 latency measurements

CONFIGURATION FILES

Dialplan configuration file on Asterisk PBX 1 (from Domain A).

extensions.conf

```
[default]
exten => s,1,Wait(1)                ;Wait a second,
just for fun
exten => s,n,Answer                 ;Answer the line
exten => s,n,Playback(Ibelieve)     ;Play a
congratulatory message

exten => s,n,Hangup()
exten => 2000,1,Dial(IAX/asterisk2/2000,30,r)
exten => 2000,2,Congestion

exten => 1717,1,Answer
exten => 1717,n,Enroll_zrtp ;permit transfer
exten => 1717,n,Hangup


exten => 1009,1,Goto(default,s,1)
include => internal

[internal]
exten => 1000,1,Dial(SIP/1000)
;exten => 2000,1,Dial(SIP/2000)
exten => 9000,1,Goto(default,s,1)

[phones]
include => internal

[trunkiax]
;exten => _2XXX,1,Dial(IAX/asterisk2/${EXTEN:1},30,r)
;exten => _2XXX,2,Congestion

[trunk]
;exten => 2XXX,1,Dial(SIP/${EXTEN}@asterisk2,120)

[sipp]
exten => 6000,1,Answer
exten => 6000,2,SetMusicOnHold(default)
exten => 6000,3,WaitMusicOnHold(20)
exten => 6000,4,Hangup
include => internal

[test]
exten => 2121,1,Dial(SIP/sippuas,20)
include => internal
```

SIP configuration file on Asterisk PBX 1 (from Domain A).

sip.conf

```
[general]
context=default   ; Default context for incoming calls

allowoverlap=no   ; Disable overlap dialing support.
(Default is yes)

bindport=5060           ; UDP Port to bind to (SIP
standard port is 5060)
bindaddr=0.0.0.0        ; IP address to bind
srvlookup=yes           ; Enable DNS SRV lookups on
outbound calls
;domain=example.com
canreinvite=no

[1000]
type=friend
host=dynamic

[1001]
type=friend
host=dynamic

[asterisk2]
secret=ultrabrice
type=peer
canreinvite=no
host=172.16.20.51
qualify=yes

[sipp]
type=friend
host=172.16.10.100
port=6000
user=sipp
canreinvite=no

[sippuac]
type=friend
username=sippuac
host=172.16.10.12
port=5060
dtmfmode=rfc2833
insecure=very
canreinvite=no
nat=yes

[sippuas]
type=friend
username=sippuas
host=192.168.10.1
dtmfmode=rfc2833
insecure=very
canreinvite=no
```

```
nat=yes
```

SIP configuration file on Asterisk PBX 1 when attempting to configure the SIP secure trunk, using both TSL and SRTP (from Domain A). As discussed, the set-up implementing SRTP could not be functional.

sip.conf

```
[general]
defaultexpirey=1800
maxexpirey=3600
pedantic=yes
srvlookup=no
srtpcapable=yes
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/cert/asterisk1.pem
tlsdontverifyserver=yes
tlscafile=/etc/asterisk/cert/ca1.crt

disallow=all
allow=all

[interboxserver2]
type=friend
host=192.168.2.2
context=callfromserver2
canreinvite=no
;transport=tls
srtpcapable=yes

[1000]
type=friend
context=phones
defaultuser=1000
username=1000
secret=1000
host=dynamic
port=5060
nat=yes
qualify=yes
;transport=tls
srtpcapable=yes
 ;canreinvite = very
;disallow=all
;allow=all

[1001]
type=friend
context=phones
defaultuser=1001
username=1001
secret=1001
host=dynamic
port=5060
srtpcapable=yes
```

```
[1002]
type=friend
context=phones
defaultuser=1002
username=1002
secret=1002
host=dynamic
port=5060
```

IAX2 configuration file on Asterisk PBX 1 (from Domain A).

iax.conf

```
[general]
bindport=4569
;NOTE: bindport must be specified BEFORE
;bindaddr or may be specified on a specific
;bindaddr if followed by colon and port
;(e.g. bindaddr=192.168.0.1:4569)
bindaddr=172.16.10.51
bandwidth=high
trunktimestamp=yes

encryption=yes
forceencryption=yes
;allow=all              ; same as bandwidth=high
;disallow=g723.1
disallow=lpc10
;allow=gsm

jitterbuffer=no
forcejitterbuffer=no
;maxjitterbuffer=1000
;maxjitterinterps=10
;resyncthreshold=1000
autokill=yes

register => asterisk1:ultrabrice@172.16.20.51

[asterisk2]
type=friend
host=dynamic
trunk=yes
auth=md5
user=asterisk2
secret=ultrabrice
qualify=yes
requirecalltoken=auto
encryption=yes
```

iax.conf on Asterisk 1:

```
[general]
bindport=4569
;NOTE: bindport must be specified BEFORE
;bindaddr or may be specified on a specific
;bindaddr if followed by colon and port
;(e.g. bindaddr=192.168.0.1:4569)
bindaddr=172.16.20.51
bandwidth=high
trunktimestamp=yes

encryption=yes
forceencryption=yes
;allow=all              ; same as bandwidth=high
;disallow=g723.1
disallow=lpc10
;allow=gsm

jitterbuffer=no
forcejitterbuffer=no
;maxjitterbuffer=1000
;maxjitterinterps=10
;resyncthreshold=1000
autokill=yes

register => asterisk2:ultrabrice@172.16.10.51

[asterisk1]
type=friend
host=dynamic
trunk=yes
auth=md5
user=asterisk2
secret=ultrabrice
qualify=yes
requirecalltoken=auto
encryption=yes
```

SIPP CONFIGURATION FILES

The SIPp server emulates a SIP server (or SIP client) that will answer to incoming calls. This configuration consist in picking up incoming phone calls, and acknowledge packets from the caller (SIPp uac, described next).

uas_pcap.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can
redistribute it and/or      -->
<!-- modify it under the terms of the GNU General
Public License as     -->
<!-- published by the Free Software Foundation;
either version 2 of the -->
<!-- License, or (at your option) any later version.
-->
<!--
-->
<!-- This program is distributed in the hope that it
will be useful,    -->
<!-- but WITHOUT ANY WARRANTY; without even the
implied warranty of     -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the      -->
<!-- GNU General Public License for more details.
-->
<!-- You should have received a copy of the GNU
General Public License  -->
<!-- along with this program; if not, write to the
-->
<!-- Free Software Foundation, Inc.,
-->
<!-- 59 Temple Place, Suite 330, Boston, MA  02111-
1307 USA             -->

<!--                 Sipp default 'uas' scenario.
-->
<scenario name="Basic UAS responder">
  <!-- By adding rrs="true" (Record Route Sets), the
route sets         -->
  <!-- are saved and used for following messages
sent. Useful to test   -->
  <!-- against stateful SIP proxies/B2BUAs.
-->
  <recv request="INVITE" crlf="true">
  </recv>

  <!-- The '[last_*]' keyword is replaced
automatically by the          -->
  <!-- specified header if it was present in the last
message received  -->
  <!-- (except if it was a retransmission). If the
```

```
header was not       -->
  <!-- present or if no message has been received,
the '[last_*]'       -->
  <!-- keyword is discarded, and all bytes until the
end of the line    -->
  <!-- are also discarded.
-->
  <!--
-->
  <!-- If the specified header was present several
times in the         -->
  <!-- message, all occurences are concatenated (CRLF
seperated)        -->
  <!-- to be used in place of the '[last_*]' keyword.
-->

  <send>
    <![CDATA[
      SIP/2.0 180 Ringing
      [last_Via:]
      [last_From:]
      [last_To:];tag=[call_number]
      [last_Call-ID:]
      [last_CSeq:]
      Contact: <sip:[local_ip]:
local_port];transport=[transport]>
      Content-Length: 0
    ]]>
  </send>


  <send retrans="500">
    <![CDATA[
      SIP/2.0 200 OK
      [last_Via:]
      [last_From:]
      [last_To:];tag=[call_number]
      [last_Call-ID:]
      [last_CSeq:]
      Contact: <sip:[local_ip]:
[local_port];transport=[transport]>
      Content-Type: application/sdp
      Content-Length: [len]
      v=0
      o=user1 53655765 2353687637 IN
IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[media_ip_type] [media_ip]
      t=0 0
      m=audio [media_port] RTP/AVP 0
      a=rtpmap:8 PCMA/8000
    ]]>
  </send>


 <!-- a=rtpmap:0 PCMU/8000 -->
  <recv request="ACK" optional="true" rtd="true"
crlf="true">
```

```
  </recv>


  <recv request="BYE">
  </recv>


  <send>
    <![CDATA[
      SIP/2.0 200 OK
      [last_Via:]
      [last_From:]
      [last_To:]
      [last_Call-ID:]
      [last_CSeq:]
      Contact: <sip:[local_ip]:
local_port];transport=[transport]>
      Content-Length: 0
    ]]>
  </send>


  <!-- Keep the call open for a while in case the 200
is lost to be     -->
  <!-- able to retransmit it if we receive the BYE
again.               -->
  <pause milliseconds="4000"/>


  <!-- definition of the response time repartition
table (unit is ms)   -->
  <ResponseTimeRepartition value="10, 20, 30, 40, 50,
100, 150, 200"/>


  <!-- definition of the call length repartition
table (unit is ms)     -->
  <CallLengthRepartition value="10, 50, 100, 500,
1000, 5000, 10000"/>


</scenario>
```

The SIP client emulated by SIPp establish a call to the SIPp server, reads a pre-recorded pcap file (approximatively 13 seconds) and hangs up the call. The following configuration allowed to simulate a SIP client, making such call.

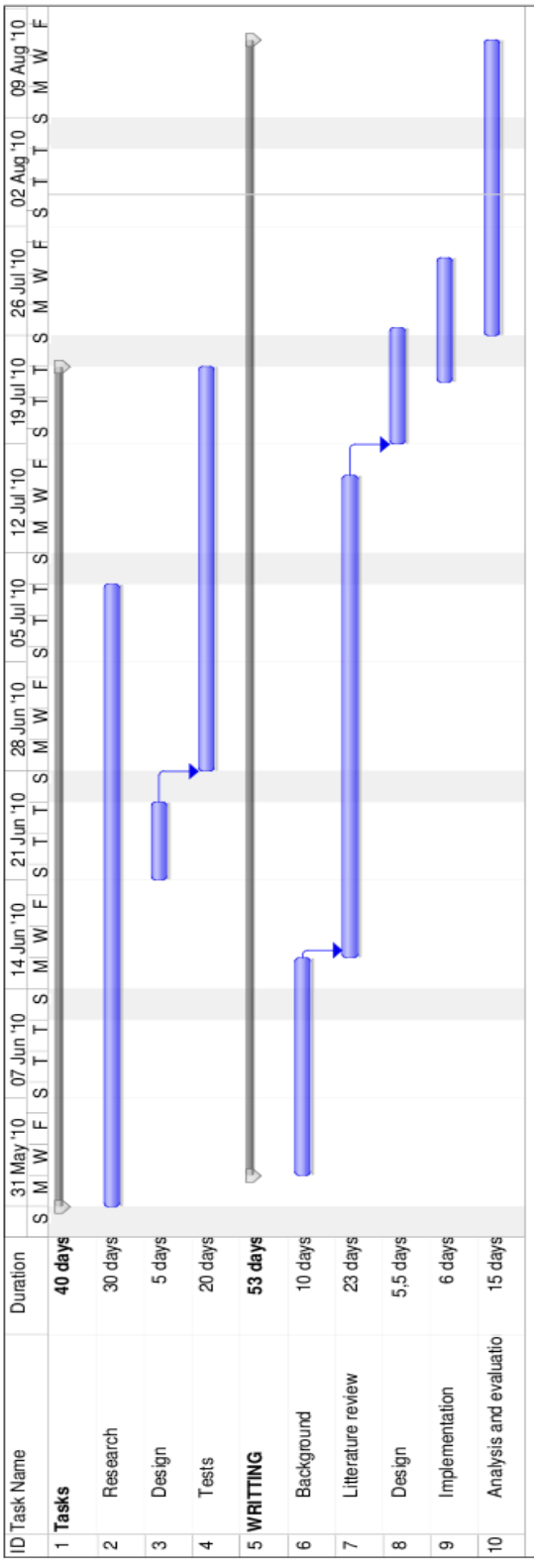
uac_pcap_custom.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can
redistribute it and/or      -->
<!-- modify it under the terms of the GNU General
Public License as     -->
<!-- published by the Free Software Foundation;
either version 2 of the -->
<!-- License, or (at your option) any later version.
-->

<!-- This program is distributed in the hope that it
will be useful,    -->
<!-- but WITHOUT ANY WARRANTY; without even the
implied warranty of     -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the      -->
<!-- GNU General Public License for more details.
-->

<!-- You should have received a copy of the GNU
General Public License  -->
<!-- along with this program; if not, write to the
-->
<!-- Free Software Foundation, Inc.,
-->
<!-- 59 Temple Place, Suite 330, Boston, MA  02111-
1307 USA             -->

<!--                 Sipp 'uac' scenario with pcap
(rtp) play           -->


<scenario name="UAC with media">
  <!-- In client mode (sipp placing calls), the Call-
ID MUST be         -->
  <!-- generated by sipp. To do so, use [call_id]
keyword.             -->

  <send retrans="500">
    <![CDATA[
      INVITE sip:[service]@[remote_ip]:[remote_port]
SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:
[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[call_number]
      To: sut <sip:[service]@[remote_ip]:
[remote_port]>
      Call-ID: [call_id]
```

```
      CSeq: 1 INVITE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Type: application/sdp
      Content-Length: [len]
      v=0
      o=user1 53655765 2353687637 IN
IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[local_ip_type] [local_ip]
      t=0 0
      m=audio [auto_media_port] RTP/AVP 8
      a=rtpmap:8 PCMA/8000
      a=rtpmap:101 telephone-event/8000
      a=fmtp:101 0-11,16
    ]]>
  </send>


  <recv response="100" optional="true">  </recv>

  <recv response="180" optional="true">  </recv>

  <!-- By adding rrs="true" (Record Route Sets), the
route sets         -->
  <!-- are saved and used for following messages
sent. Useful to test   -->
  <!-- against stateful SIP proxies/B2BUAs.
-->
  <recv response="200" rtd="true" crlf="true">
</recv>

  <!-- Packet lost can be simulated in any send/recv
message by         -->
  <!-- by adding the 'lost = "10"'. Value can be [1-
100] percent.       -->
  <send>
    <![CDATA[
      ACK sip:[service]@[remote_ip]:[remote_port]
SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:
[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[call_number]
      To: sut <sip:[service]@[remote_ip]:
[remote_port]>[peer_tag_param]
      Call-ID: [call_id]
      CSeq: 1 ACK
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Length: 0
    ]]>
  </send>

  <!-- Play a pre-recorded PCAP file (RTP stream)
-->  <nop>
```

```
    <action>
      <exec play_pcap_audio="pcap/mypcap.pcap"/>
    </action>
  </nop>

  <!-- Pause 13 seconds, the time for the pcap file
to be played        -->
  <!-- PCAP file
-->
  <pause milliseconds="13800"/>


  <!-- The 'crlf' option inserts a blank line in the
statistics report. -->


  <send retrans="500">
    <![CDATA[
      BYE sip:[service]@[remote_ip]:[remote_port]
SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:
[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[call_number]
      To: sut <sip:[service]@[remote_ip]:
[remote_port]>[peer_tag_param]
      Call-ID: [call_id]
      CSeq: 2 BYE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Length: 0
    ]]>
  </send>


  <recv response="200" crlf="true">  </recv>

  <!-- definition of the response time repartition
table (unit is ms)   -->
  <ResponseTimeRepartition value="10, 20, 30, 40, 50,
100, 150, 200"/>

  <!-- definition of the call length repartition
table (unit is ms)     -->
  <CallLengthRepartition value="10, 50, 100, 500,
1000, 5000, 10000"/>

</scenario>
```

PROJECT MANAGEMENT



PROJECT PROPOSAL

1. Student details

Last (family) name	BERTHELOT
First name	Florian
Napier matriculation number	07010875

2. Details of your programme of study

MSc Programme title	Advanced Networking
Year that you started your diploma modules	2009
Month that you started your diploma modules	September
Mode of study of diploma modules	Full-time
Date that you completed/will complete your diploma modules at Napier	01/12/10

3. Academic eligibility to continue to the Masters dissertation module

Please confirm that status of your module completions by ticking the appropriate box:

I have a minimum of 7 15-credit module passes and 1 x F1, or 5 20-credit module passes and 1 x F1, and so I am already eligible to proceed to the MSc dissertation module.	**YES**
My academic eligibility to continue to the Masters dissertation module is subject to the outcome of module results to be presented at the next exam board.	**YES**

4. Fees/debt status

Please confirm that you have no outstanding debts to the University by ticking the box below. (Students who owe debts to the University, e.g. for fees, library fines, cannot be accepted on to the Masters dissertation module. You should not submit a proposal if you cannot clear your debts in time for the proposal deadline.)

I confirm that I have no outstanding debts to the University	**YES**

5. Project outline details

Please suggest a title for your proposed project. If you have worked with a supervisor on this proposal, please provide the name. NB you are strongly advised to work with a member of staff when putting your proposal together.

Title of the proposed project	**Performance-cost analysis of secure protocols in VoIP systems**
Name of supervisor	**Prof. Bill Buchanan**
I do not have a member of staff lined up to supervise my work	

6. Brief description of the research area - background

Please provide background information on the *broad research area* of your project in the box below. You should write in narrative (not bullet points). The academic/theoretical basis of your description of the research area should be evident through the use of references. Your description should be between half and one page in length.

Transportation of voice over switched packet network have been widely implemented lately. The convergence of IP networks (carrying data) and PSTN (Public Switched Telephony Network) is a cost effective solution and allows the expansion of new services (like video-conference). However, VoIP lacks of security. Confidentiality, Integrity are compromised (Benini, 2008), and new protocols are developed to mitigate the known issues.

VoIP has more requirements than any standard Internet application: as this is a real-time application, some parameters define the Quality of Service (QoS). These parameters can be measured the QoS of the service. However, the use of secure mechanisms to provide authentication, confidentiality and Integrity inflict a loss of VoIP QoS. Some standard solutions, like tunnelling, have been adapted to provide VoIP security. However, these solutions often lack of scalability and affect seriously voice QoS. Some new protocols have been developed recently to comply to VoIP requirements.

Recently, Phil Zimmermann (creator of PGP, most used solution in secure email) has developed a new protocol to secure and authenticate in an efficient way the voice payload over IP. ZRTP is a Media Path Key Agreement for Unicast Secure RTP (Zimmermann, 2010). It uses existing and reliable cryptography tools (Diffie-Hellman key exchange and AES encryption algorithm) to encrypt voice. The protocol has been evaluated and recognized as reliable and very secure (Bresciani, 2009). However, like any public-key infra, ZRTP is sensitive to man-in-the-middle [Mitm] attack (Gupta, 2007).

Another recent and secure protocol has been developed by Digium: IAX2 (Inter-Asterisk eXchange Version 2). This protocol, all in one, was developed to interconnect Asterisk PBX. The protocol is now an IETF draft and includes security mechanisms (authentication using PKI and AES encryption). Although IAX2 protocol was developed to interconnect Asterisk PBX, it allows end-points to transfer voice and signalling into a same single channel. However, security mechanisms are not implemented yet onto end-points. Encryption / decryption is performed only by PBX. Because of its simplicity to implement in a IP network (single channel used for voice and signalisation), IAX2 is presented a new standard in VoIP architectures. Encryption features

are not implemented yet onto end-points. This aspect will be investigated by the dissertation.

A study demonstrated the impact of secure protocol onto voice communication (Guillen, 2009), and the impact on the PBX have been investigated using standard and non secure protocols, like SIP (Mohiuddin, 2008). Performances using secure protocols have not been evaluated yet onto Asterisk. As the PBX will encrypt and decrypt the traffic for the users, making concurrent calls will have an impact on the PBX performance.

The dissertation will present an evaluation of performances when using secure protocols. The investigation will first present the known solutions to secure VoIP, why they are not long-term solutions. Then the recent protocols (IAX2 and ZRTP) will be investigated. A design will present how the performances can be measured, what are the most important parameters and what scenario will be deployed. The implementation itself will demonstrate how the measures are performed, what steps are more difficult to implement (only a little documentation for ZRTP modules). Then a critical evaluation will present a critical analysis of the obtained results.

This study will demonstrate if the recent secure protocols have good performance, present a set of recommendation and help to dimension a server to use Asterisk.

7. Project outline for the work that you propose to complete

Please complete the project outline in the box below. You should use the emboldened text as a framework. Your project outline should be between half and one page in length.

The idea for this research arose from:

- Me with the help of Abou Sofyane Khedim

The aims of the project are as follows:

- Measurement of Voice quality when using secure protocols
- Measurement of PBX performance using different secure technologies
- Evaluation of most suitable protocols to secure voice-over-ip

The main research questions that this work will address include:

- How secure is VOIP in present systems?
- Is IAX2 an improvement in term of performances to secure VOIP?
- What are the limitations of a ASTERISK PBX when secure channels are established?

The software development/design work/other deliverable of the project will be:

- Designing a VoIP topology using Asterisk PBXs and soft-phones
- Implementation of secure techniques on an Asterisk PBX
- Measurement of PBX performance
- Evaluation of IAX2 protocol in comparison to other standard solutions

The project will involve the following research/field work/experimentation/evaluation:

- Recent solution to provide secure voice transmission over IP Network
- Evaluation of ZRTP using a PBX as trusted man-in-the-middle
- Performance impacts on Asterisk using ZRTP and IAX2
- Feasibility to use IAX2 as end-to-end encryption

This work will require the use of specialist software:

- Asterisk PBX, available for free at www.asterisk.org
- Zfone, available for free from http://zfoneproject.com

This work will require the use of specialist hardware:

- None required

The project is being undertaken in collaboration with:

- No one

8. References

Please supply details of all the material that you have referenced in sections 6 and 7 above. You should include at least three references, and these should be to high quality sources such as refereed journal and conference papers, standards or white papers. Please ensure that you use a standardised referencing style for the presentation of your references, e.g. APA, as outlined in the yellow booklet available from the School of Computing office and http://www.dcs.napier.ac.uk/~hazelh/gen_ho/apa.pdf

Benini M. and Sicari S. (2008). *Assessing the risk of intercepting VoIP calls*. Rerieved May 2010 from Science direct database.

Bresciani R. and Butterfield A. (2009). *A formal security proof for the ZRTP Protocol*. Retrieved May 2010 from IEEExplorer database.

Guillen E. P. and Chacon D. A. (2009). *VoIP Networks Performance Analysis with Encryption Systems*. Retrieved May 2010 from Google Scholars

Gupta P. and Shmatikov V. (2007). *Security Analysis of Voice-over-IP Protocol*s. Retrieved May 2010 from IEEExplorer database.

Mohiuddin A., Mansor A. M. (2008). *CPU dimensioning on performance of Asterisk VoIP PBX*. Retrieved May 2010 from ACM portal.

Voixzilla (2009). Part 1: VoIP in a Cloud. Retrieved May 2010 from

http://voxilla.com/2009/02/12/amazon-ec2-voip-1096

Voixzilla (2009). *Part 2: Asterisk in a Cloud*. Retrieved May 2010 from http://voxilla.com/2009/02/13/asterisk-amazon-ec2-1178

Zimmermann P. (2010). *The Zfone™ Project*. Retrieved May 2010 from http://zfoneproject.com/prod_zfone.html

Zimmermann P., Johnston A., Avaya Ed. and Callas J. (May 21, 2010). *ZRTP: Media Path Key Agreement for Unicast Secure RTP*. Retrieved May 2010 from http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-21

9. Ethics

If your research involves other people, privacy or controversial research there may be ethical issues to consider (please see the information on the module website). If the answer below is YES then you need to complete a research Ethics and Governance Approval form (available on the website).

Does this project have any ethical or governance issues related to working with, studying or observing other people? (YES/NO)	**NO**

10. Supervision timescale

Please indicate the mode of supervision that you are anticipating. If you expect to be away from the university during the supervision period and may need remote supervision please indicate.

Weekly meetings over 1 trimester	**YES**
Meetings every other week over 2 trimesters	
Other	

11. Submitting your proposal

Please save this file using your surname, e.g. macdonald_proposal.doc, and e-mail it to the module leader in time for the next proposal deadline.