Summer 2017

Security Enhancements in Voice Over Ip Networks

Seyed Amir Iranmanesh
College of William and Mary - Arts & Sciences, iranmanesh@gmail.com

Security Enhancements in Voice over IP Networks

Seyed Amir Iranmanesh

Williamsburg, VA

Computer Engineering, B.Sc, Shahed University
Computer Engineering, M.Sc, Isfahan University of Technology

A Dissertation presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Computer Science

The College of William and Mary
January 2018

APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Seyed Amir Iranmanesh

Seyed Amir Iranmanesh

Approved by the Committee, January 2018

Committee Chair
Adjunct Professor Haining Wang, Computer Science
College of William & Mary

Professor Qun Li, Computer Science
College of William & Mary

Adjunct Associate Professor Kun Sun, Computer Science
College of William & Mary

Associate Professor Gang Zhou, Computer Science
College of William & Mary

C. J. Cotton

Professor Chase Cotton, Electrical and Computer Engineering
University of Delaware

ABSTRACT

Voice delivery over IP networks including VoIP (Voice over IP) and VoLTE (Voice over LTE) are emerging as the alternatives to the conventional public telephony networks. With the growing number of subscribers and the global integration of 4/5G by operations, VoIP/VoLTE as the only option for voice delivery becomes an attractive target to be abused and exploited by malicious attackers.

This dissertation aims to address some of the security challenges in VoIP/VoLTE. When we examine the past events to identify trends and changes in attacking strategies, we find that spam calls, caller-ID spoofing, and DoS attacks are the most imminent threats to VoIP deployments. Compared to email spam, voice spam will be much more obnoxious and time consuming nuisance for human subscribers to filter out. Since the threat of voice spam could become as serious as email spam, we first focus on spam detection and propose a content-based approach to protect telephone subscribers' voice mailboxes from voice spam.

Caller-ID has long been used to enable the callee parties know who is calling, verify his identity for authentication and his physical location for emergency services. VoIP and other packet switched networks such as all-IP Long Term Evolution (LTE) network provide flexibility that helps subscribers to use arbitrary caller-ID. Moreover, interconnecting between IP telephony and other Circuit-Switched (CS) legacy telephone networks has also weakened the security of caller-ID systems. We observe that the determination of true identity of a calling device helps us in preventing many VoIP attacks, such as caller-ID spoofing, spamming and call flooding attacks. This motivates us to take a very different approach to the VoIP problems and attempt to answer a fundamental question: is it possible to know the type of a device a subscriber uses to originate a call? By exploiting the impreciseness of the codec sampling rate in the caller's RTP streams, we propose a fuzzy rule-based system to remotely identify calling devices.

Finally, we propose a caller-ID based public key infrastructure for VoIP and VoLTE that provides signature generation at the calling party side as well as signature verification at the callee party side. The proposed signature can be used as caller-ID trust to prevent caller-ID spoofing and unsolicited calls. Our approach is based on the identity-based cryptography, and it also leverages the Domain Name System (DNS) and proxy servers in the VoIP architecture, as well as the IP Multimedia Subsystem (IMS) architecture. Using OPNET, we then develop a comprehensive simulation test-bed for the evaluation. Our simulation results show that the call setup delays induced by our infrastructure are hardly noticeable by the subscribers and the extra signaling overhead is negligible. Therefore, our proposed infrastructure can be adopted to widely verify caller-ID in telephony networks.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to all the American people who have been part of my research journey as an international student and made it truly wonderful experience. I especially would like to express my gratitude to my PhD advisor Dr. Haining Wang for his guidance, advice and support. In spite of my difficult situation, he has been always patient and supportive. I want him to know that he is not only my PhD advisor but also my life advisor.

I would like to thank my committee members Dr. Gang Zhou, Dr. Qun Li, Dr. Kun Sun, and Dr. Chase Cotton for their valuable insight and directions. I also extend my gratitude to all the faculty members, staff, and fellow students at the College of William and Mary. Special thanks goes to Vanessa Godwin and Stephen Sechrist for their endless support.

Finally, I give my deepest thanks to my parents, family, and friends for their unconditional love and support. Special thanks goes to Fereshteh Yousefirizi, my love, for being always with me, in good and bad moments, it should be not easy at all to suffer my stress situations over this long way.

For Gholamreza Yousefirizi, my father-in-law who left fingerprints of grace in his last days on my life, in memoriam ...

LIST OF TABLES

LIST OF FIGURES

Security Enhancements in Voice over IP Networks

Chapter 1

Introduction

Voice over IP (VoIP) telephony is emerging as an alternative to traditional public switched telephone networks (PSTN). IP telephone service providers are moving fast from low-scale toll bypass deployments to large-scale competitive carrier deployments; thus it gives an opportunity to enterprise networks a choice of supporting less expensive single network solution rather than using multiple separate networks. The broadband-based residential customers also switch to IP telephony due to its convenience and cost effectiveness. In contrast to a traditional telephone system (where the end devices are dumb), the VoIP architecture pushes intelligence towards the end devices (e.g., PCs and IP phones) and enables many new services. This flexibility coupled with the growing number of subscribers becomes an attractive target to be abused by malicious users.

As the number of VoIP subscribers hits a critical mass, it is expected that voice spam will emerge as a serious threat. Evidently, the effectiveness of telephone calls presents strong incentives for spammers to establish voice channels with many subscribers at the same time. Such machine generated unsolicited bulk calls, known as SPIT (Spam over Internet Telephony), may hinder the deployment of IP telephony. If the problem remains unchecked, then it may become as serious as email spam today. However, voice spam is much more obnoxious and harmful than email spam. The ringing of telephone at odd time, answering of spam calls, phishing attacks, and inability to filter spam messages from the voicemail box without listening each one, are real nuisance and waste of time. Thus, how to detect voice spams and filter them out is a challenging task.

The delivery of voice and other multimedia streams such as video over IP needs two separate protocols: one for signaling to initiate the multimedia session and another one for multimedia itself. Over the past decade, Voice over IP (VoIP) and Voice over LTE (VoLTE) as emerging technologies utilize the Session Initiation Protocol (SIP) [42] as their signaling protocol for media session initiation, and have replaced many traditional telephony deployments [55]. In spite of being widely used, SIP does not provide any effective standards for calling party identification in a SIP session. Since SIP sends `INVITE` requests in plaintext with no authentication method to initiate a call, Caller-ID included inside a request can be spoofed by malicious users to mount different attacks.

Caller-ID has long been used to let the callee parties know who is calling, verify his identity for authentication and his physical location for emergency services. VoIP and other packet switched networks such as all-IP Long Term Evolution (LTE) network provide flexibility that helps subscribers to use arbitrary Caller-ID. There are also many fake ID providers such as TeleTurd [8] that allow their customers to claim any Caller-ID. Thus, Caller-IDs are no longer dependable. Moreover, interworking SIP with PSTN has ultimately reduced the security of Caller-ID systems. Caller-ID spoofing can be used for many malicious attacks, including vishing attacks, illegal robocalls, and swatting. In addition to Caller-ID spoofing, there are many other VoIP attacks such as spamming, DoS attacks, and call flooding attacks, in which the faked identities of calling devices are also widely used. Thus, how to identify a calling party and prevent Caller-ID spoofing is another challenging task that essentially helps in preventing other VoIP attacks such as voice spamming.

There have been a number of solutions proposed to address voice spam, Caller-ID spoofing, and flooding attacks. However, based on simple assumptions or requiring too much network overhaul, these previous defenses are limited to work well in practice. In this dissertation, we first focus on the SPIT problem and propose a content-based solution to protect telephone subscribers' voice mailboxes from voice spam. We then examines the Caller-ID spoofing problem by exploiting the impreciseness of the codec sampling rate in the caller's RTP streams and propose a fuzzy rule-based system to remotely identify calling devices. Finally, we propose a distributed key infrastructure for building identity-based Caller-ID signatures that can address the Caller-ID spoofing problem explicitly and the SPIT problem implicitly. The ultimate goal of this dissertation research is to shed the light on how to ensure voice delivery in a more secure and dependable manner.

1.1 SPIT CLEANER: A Voice Spam Filter To Clean Subscribers' Voice Mailbox

On one hand, a voice spam is similar to an email spam in many aspects. Hence the execution style of email spam can be easily adapted to launch voice spam attacks. For example, a voice spammer harvests a user's SIP URI (Uniform Resource Identifier) or telephone number from the telephone directories or by using spam bots crawling over the Internet, sends out call setup request messages and plays a pre-recorded .wav file. On the other hand, many mechanisms that work for email spam fail completely in the context of VoIP.

Previous research has made some efforts to address the voice spam problem. The proposed solutions generally distinguish a legitimate subscriber from a spammer using only SIP signaling messages. Instead of analyzing the SIP signaling messages to identify a spammer, we propose a speaker independent speech recognition scheme for content filtering to avoid spam message deposition on the subscribers' voice mailboxes.

The main contributions of our work are summarized as follows:

1. We design and develop a voice mailbox filtering approach based on three observations listed below, in addition to analyzing SIP signaling messages for caller identification:
   - The spammers prefer to achieve a high hit ratio for their spamming attacks and send as much spam as possible within a short duration of time.
   - The spammers play pre-recorded messages to many spam victims at the same time.
   - The spammers are expected to run interactive voice response (IVR) systems that can either interact with users or leave a voice mail if a call is not answered.
2. To the best of our knowledge, this is a first attempt to clean subscribers' voice mailboxes from voice spam messages. Moreover, the proposed approach also provides a way to identify spamming sources from spam messages.
3. The uniqueness of our spam filtering approach lies in its independence on the generation of voice spam, regardless whether spammers play same spam content recorded in many different ways, such as human or machine generated voice, male or female voice, and different accents.
4. We validate the efficacy of the proposed scheme through realistic case scenarios. The experimental results show that our approach is computationally efficient and very scalable.

1.2 Exploitation of Imprecise CODEC Sampling Rate: A Method for Remote IP Phone Identification

Caller-ID spoofing can be used to initiate different malicious attacks. Many of such attacks are possible because (1) the caller ID could be manipulated in VoIP clients, (2) a number of calls can be generated simultaneously, and (3) there is no limitation on how many active sessions can be maintained. Most of the existing efforts to tackle VoIP related attacks are focused on determining the identity of callers, developing stronger authentication mechanism, and analyzing the signaling messages.

In VoIP, once a user account is compromised, a malicious software running on a general purpose PC can mimic any device behavior and launch attacks. Moreover, the caller-ID could be faked by unscrupulous ID providers to initiate various attacks. As caller-ID can be easily spoofed in VoIP environments, many security threats are imposed by unsolicited communications, such as voice spam, voice phishing (vishing), and flooding attacks. Such VoIP attacks are mainly launched from PC-based malicious programs pretending to be IP hard phones.

Today VoIP service providers can neither identify compromised subscriber accounts in their domains nor prevent unsolicited calls originated by faked caller-ID in other domains. We aim to propose a defense system to identify the type of calling devices, which can address the problems above mentioned. Different SIP devices have different clocks; IP hardphones not only have higher-quality clocks than PC-based soft clients, but also have different clocks themselves. Hence, we utilize clockskew to build a fuzzy based system to remotely identify the type of calling devices.

The main contributions of our work are summarized as follows:

1. We demonstrate how to locally measure the *clock skew* for calling devices that can be

used for device identification. We further implement the device identification module as a loadable module of the SBC (Session Border Controller) device to prevent the modification of calling devices.

2. Our proposed approach is not specific to a particular type of VoIP attacks. Instead, it can be used to prevent many known and unknown attacks, in which hackers use and recruit zombies over the Internet to behave like a SIP user agent and make unsolicited fraudulent calls. Our fuzzy rule-based identification system works in a passive manner without any interaction with the caller, and it does not require the overhauling of network infrastructures or protocols.

3. We set several experimental testbeds to evaluate the efficacy of our proposed system to identify calling devices in different scenarios. Our experimental results show that our approach can accurately distinguish between hardphones and softclients as well as identify make/model of hardphones with detection accuracy rate of 95%.

1.3 Caller-ID as Digital Signature: A Method to Automatically Authenticate Calling Parties

New mobile communication technologies have emerged in recent years due to the high demand for new multimedia services, such as IP telephony and live conferencing that require high-speed data rate. Thus, the 3rd Generation Partnership Project (3GPP) has introduced the Long Term Evolution (LTE) standard. The LTE is designed to provide high throughput and low latency capabilities for mobile operators to migrate voice from their congested and costly circuit-switched networks to an efficient and simplified all-IP network architecture. LTE proposes a long term solution for packet-based voice telephony, named VoLTE (one voice), which utilizes the IP Multimedia Subsystem (IMS) network. Since IMS, similar to VoIP, utilizes SIP as its signaling protocol, it inherits the security vulnerabilities of SIP. SIP transmits caller-IDs in plaintext without using any authentication mechanisms, which has enabled malicious attackers to launch various attacks.

As our research has focused on making voice delivery over IP more secure and dependable, we aim to secure the caller-ID system in IP telephony, including both VoIP and VoLTE. Our research is conducted in a two-pronged approach as follows:

1. We propose an implicit authentication method for SIP telephony and voice over IMS. We modify the registration phase of SIP and IMS to extract specific properties of a registering user, such as MAC address, to make his unique profile and record it for future implicit authentication.

2. We propose a distributed public key infrastructure for callerID based signature generation and verification to restore caller-ID trust for the whole telephony network, including both packet-switched and circuit-switched networks. Our solution benefits from the distributed nature of SIP architecture and its existing elements to build a key distribution system for signature generation/verification.

1.4 Organization

This dissertation is organized as follows. In Chapter 2, we present SPIT CLEANER that cleans voice spam from subscribers' voice mailboxes. In Chapter 3, we present our proposed fuzzy based system that exploits imprecise CODEC sampling rates to remotely identify calling devices. In Chapter 4, we describe our distributed key infrastructure for building caller-ID based signatures to restore caller-ID trust for the whole telephony network. Finally, we conclude this dissertation in Chapter 5.

Chapter 2

SPIT CLEANER: A Voice Spam Filter To Clean Subscribers' Voice Mailbox

IP telephone service providers are moving fast from low-scale toll bypass deployments to large-scale competitive carrier deployments; thus giving an opportunity to enterprise networks for supporting less expensive single network solution rather than multiple separate networks. The broadband-based residential customers also switch to IP telephony due to its convenience and cost effectiveness. In contrast to traditional telephone system in which the end devices are dumb, the VoIP architecture pushes intelligence towards the end devices like PCs and IP phones, creating many new services. This flexibility coupled with the growing number of subscribers has attracted attackers for malicious resource abuse.

As the number of VoIP subscribers hits a critical mass, it is expected that voice spam will emerge as a serious threat. In fact, Japan, where VoIP market is much more mature than USA, has witnessed some recent voice spam attacks. The SoftbankBB, a VoIP service provider with 4.6 million users has reported three incidents of spam attacks within its own network [73]. These incidents include unsolicited messages advertising an adult website, scanning of active VoIP phone numbers and requesting personal information of users. Similarly, Columbia University experienced a voice spam attack, with someone accessing the SIP proxy server and "war dialing" a large number of IP phone extensions [75]. There are many reported incidents of spam messages on Google voice too [32]. Evidently, the effectiveness of telephone calls presents strong incentives for spammers to establish voice channels with many subscribers at the same time. Such machine generated unsolicited bulk calls known as SPIT (Spam over Internet Telephony) may hinder the deployment of IP telephony, and if the problem remains unchecked then it may become as serious as email spam today. In many aspects, the voice spam is similar to an email spam. Moreover, voice spam will be much more obnoxious and harmful than email spam. The ringing of telephone at odd time, answering of spam calls, phishing attacks and inability to filter spam messages from the voicemail box without listening each one are real nuisance and waste of time.

In the past, a number of anti-spam solutions have been proposed. These solutions generally distinguish a legitimate subscriber from a spammer using only SIP signaling messages. However, in this chapter we take a very different approach. Instead of analyzing the SIP signaling messages and identifying the spam originating source(s) or ascertaining the real identity of spammers, we try to avoid spam message deposition on the subscribers' voice mailboxes. The goal of the proposed approach is two-pronged. First, we allow only legitimate messages to be deposited on the subscribers' mailbox account, unsolicited spam messages are blocked at the media server itself. Second, the proposed approach also provides a way to identify spamming sources based on spam messages. We deisgned our solution based on the assessment criteria provided in a recent work [72]; Usability, Deployability and Robustness. To the best of our knowledge, this is a first attempt to clean subscribers' voice mailboxes from voice spam messages.

2.1 Overview of Proposed Approach.

Beyond the basic observation that SIP signaling messages needs to be analyzed for its source and caller identification, we make three additional observations that are central to our approach. First, the spammers would prefer to see high hit ratio for their spamming attacks. Thus, most of the spamming attacks are expected to occur in bulk (i.e., as much spam as possible within a short duration of time) and most of the spam messages will be delivered to voice mailboxes. Second, during the spam attack instance, a spammer will play pre-recorded messages to many of the spam victims at the same time. Third, the originating spam source is expected to be some sort of interactive voice response (IVR) system, which can interact with the users if the calls are answered and it should also be able to leave a voice mail if the calls are not answered. However, it should be noted that in most of the spam attacks the voice stream originating from the spam source is machine generated. Based on these observations, we design and develop a voice mailbox filtering approach.

In our approach, we first segment voice messages in their voiced segments using a silence removal technique. Our silence removal technique is based on two audio features; the *signal energy* and the *spectral centroid*. After calculating the partial similarity between each pair of voiced segments coming from two different voice messages, we can determine how similar are the two voice messages content-wise. To measure the similarity between two voiced segments as a metric for content comparison, we use the technique of Dynamic Time Warping (DTW) to compute the cosine similarity between two sequences of speech feature matrices. A popular speech feature representation known as RASTA-PLP (Relative Spectral Transform - Perceptual Linear Prediction) is used to extract speech feature matrices from voice messages. After a message is left on the server by a caller, it is divided into voiced segments using our segmentation method and RASTA-PLP spectra for its voiced segments being calculated. Using our DTW based system, the RASTA-PLP matrix is then matched against a set of spam signatures. If a match is not found, our system is further coupled with Bayesian filtering to reveal the hidden spam words/phrases within a voice message to show how closely (probabilistically) it matches with the known spam messages seen in the past. Normally during a spam attack, many of the deposited voice messages share the same content, we finally use our speaker independent speech recognition technique to find how many similar messages (in content) are deposited within a predefined time interval of ΔT.

We conduct two sets of experiments to evaluate the effectiveness of our proposed solution against realistic spam attack scenarios. In the first experiment, we investigate the most generic spam attack scenario, where a spammer repeatedly sends the same spam message to many of the subscribers at the same time. Three hundred voice messages in various size are deposited from thirty speakers with different accents (such as American, British, or Indian English), different sex and ages to form the scenario. In the second set of experiment, we investigate the power of our method to classify voice messages as spam and non-spam, in which the deposited voice messages include spam words/phrases. Our experimental results show that our approach is computationally efficient, and speaker independent to identify a common segment of voice message out of a database of known spam signatures and classify the voice message correctly.

The remainder of the chapter is structured as follows. The SIP-based IP telephony and voice message deposition process is presented in Section 2.2. Related work to this chapter are presented in Section 2.3. In Section 2.4, we describe technical details and the steps of spam filtering scheme. In Section 2.5, we discuss spam detection methodology. Section 2.6

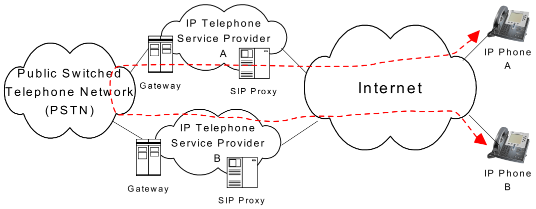


Figure 2.1: Island-based SIP VoIP Deployment

analyzes the performance of the proposed solution. Finally, Section 5.1 concludes this chapter.

2.2 Background

Voice spam is an extension of email spam in the VoIP domain. The technical know-how and execution style of email spam can easily be adapted to launch voice spam attacks. For example, a voice spammer first harvests user's SIP URIs or telephone numbers from the telephone directories or by using spam bots crawling over the Internet. Then, a compromised host is used as a SIP user agent (UA) that sends out call setup request messages. Finally, the established sessions are played with a pre-recorded .wav file. However, voice spam is much more obnoxious and harmful than email spam. The ringing of telephone at odd time, answering of spam calls, phishing attacks and inability to filter spam messages from the voicemail box without listening each one are real nuisance and waste of time.

Before we delve into voice spam problem, we briefly describe the basic VoIP architecture as it serves two purposes: first, it explains as why we do not hear much of voice spam attacks today as compared to email spam; second, it also describes as why it could be a serious problem for VoIP subscribers in the near future.

2.2.1 VoIP Architecture

As shown in Fig. 2.1, in today's IP telephony world most of the VoIP service providers (such as Vonage, AT&T Callvantage, and ViaTalk) operate in partially closed environments and are connected to each other through the public telephone network. VoIP service providers allow only their own authenticated subscribers to access SIP proxy server resources. The authentication of call requests is feasible because user accounts are stored locally on the VoIP service provider's SIP servers. However, in general the threat of spam calls is associated with the open architecture of VoIP service, where VoIP service providers interact with each other through the IP-based peering points. It provides an ability for individual subscribers to connect with each other without traversing the PSTN cloud. Therefore, it is quite possible that an `INVITE` message received by a VoIP service provider from another service provider (through IP network) for one of its subscriber may not have any type of authentication credentials for the calling party.

Recently, we are witnessing a large demand for SIP trunks. A SIP trunk is a service offered by a VoIP service provider permitting business subscribers to reach beyond the enterprise network and connect to the PSTN through IP-based connections. Generally most of the SIP trunks are set up without authentication. Only few of the service providers use TLS or IPSec to secure SIP signaling. In this scenario, a spam attack can be launched from within the enterprise network (e.g., a corporate network is infected with malicious worm) or by a man-in-the-middle where SIP signaling is transported over the Internet in plaintext without any encryption.

2.2.2 SIP-based IP Telephony

The Session Initiation Protocol (SIP) [62], belonging to the application layer of the TCP/IP protocol stack, is used to set up, modify, and tear down multimedia sessions including

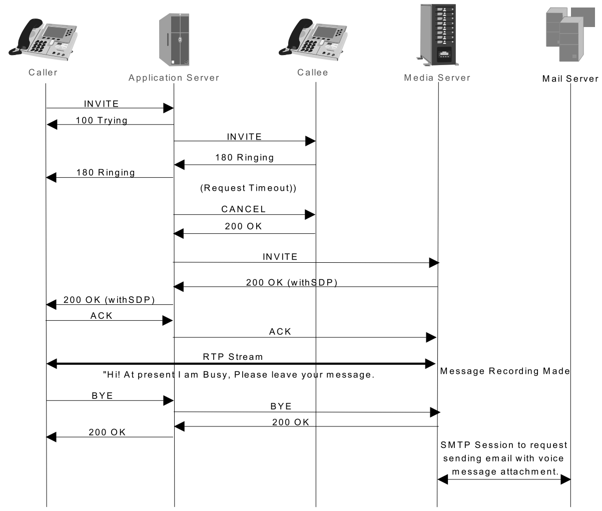


Figure 2.2: Voice Message Deposition

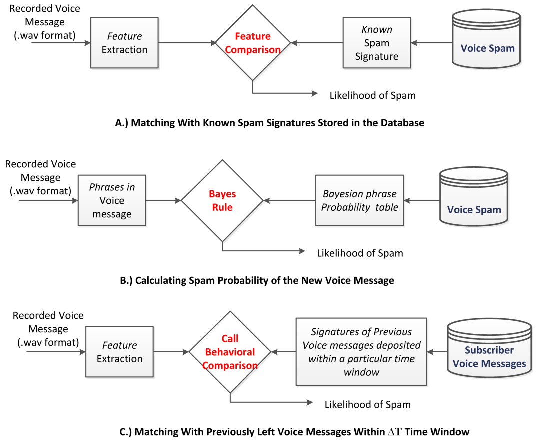
telephone calls between two or more participants.

SIP-based telecommunication architectures have two types of elements: end devices referred to as user agents (UAs) and SIP servers. Irrespective of being a software or hardware phone, UAs combine two sub-entities: the connection requester referred as the user agent client (UAC) and the connection request receiver referred to as the user agent server (UAS). Consequently, during a SIP session, both UAs switches back and forth between UAC and UAS functionalities.

SIP messages consisting of request-response pairs are exchanged for call set up, from six kinds including `INVITE, ACK, BYE, CANCEL, REGISTER`, and `OPTIONS` - each identified by a numeric code according to RFC 3261 [62].

2.2.3 Voice Mail Deposition

A simple voice message deposition scenario is shown in Fig. 2.2. A caller calls a callee who is busy and unable to take phone call, in this particular case, the call is answered by a voice messaging system. The call is set up between caller and callee's voice messaging system


Recorded Voice Message (.wav format)
Feature Extraction
Feature Comparison
Known Spam Signature
Voice Spam
Likelihood of Spam
Recorded Voice Message (.wav format)
Phrases in Voice message
Bayes Rule
Bayesian phrase Probability table
Voice Spam
Likelihood of Spam
Recorded Voice Message (.wav format)
Feature Extraction
Call Behavioral Comparison
Signatures of Previous Voice messages deposited within a particular time window
Subscriber Voice Messages
Likelihood of Spam

A.) Matching With Known Spam Signatures Stored in the Database

B.) Calculating Spam Probability of the New Voice Message

C.) Matching With Previously Left Voice Messages Within ΔT Time Window

Figure 2.3: Overview of Spam Filtering Approach

that plays a “busy” greeting message and asks the caller to leave a voice message. The caller records the voice message and then hangs up. With the *SendMail* command, the application (i.e., call control) server requests the media server to deliver the recorded voice message to the callee’s inbox. The media server sends email with the recorded message as an attachment (in .wav file format) to the user account on SMTP mail server.

2.2.4 Overview of Spam Filtering Approach

As shown in Fig. 2.3, our spam filtering approach can be briefly described as a three-step process. Given a recorded voice message, we first verify if it matches with any of the known spam signatures stored in the database. For example, when a caller leaves a voice message for a callee, media server records the RTP stream and converts it into a .wav file. The *feature extraction* process takes this .wav file as an input and extracts few features from the corresponding spectrogram. This set of features is searched in the database to find a match with known spam signatures. In the second step, even if a match is not found with known spam signatures, we observe the words and phrases and their spamicity. The overall spam score of the message determines its likelihood of being a spam message. In the third step, we observe how many similar messages (in content) are deposited within a predefined time interval of ΔT.

2.3 Related Work

The SIP IETF working group has published a couple of informational drafts proposing *computational puzzles* to reduce spam in SIP environment and an extension of SIP protocol to send user's feedback information to the SPIT identification system [50, 61]. To some extent, the combination of user's whitelist with the *Turing tests* or computational puzzles can prevent spam calls. However, the capability of a SIP UA to solve the computational puzzle relies on its computing resources. Therefore, it cannot be ignored that a spammer can potentially have significantly more resources than a normal user. The solving of audio Turing tests requires caller's time and manual intervention. Still, the Turing tests cannot be a solution for deaf (or blind) users and can be thwarted by employing cheap labor. Recently, a number of products such as Sipera's IPCS [69] and NEC's VoIP SEAL [49] incorporate audio Turing test to solve the voice spam problem. However, an attacker may abuse these security devices as reflectors and amplifiers to launch a stealthy DDoS attack [65]. A recent work [72] analyzes existing antispam techniques to systematically categorize them into the following classes: (1) Call Request Header Analysis, (2) Voice Interactive Screening, and (3) Caller Compliance. Now we review some of the other related works on SPIT prevention from these different categories.

Inferring Spoken Words. Closest to our approach is a method where spam detection module detects spoken words within an established voice stream. Most intuitive way to detect a spam message is to use *"speech-to-text"* engine where deposited voice message can be converted to text format and then use well-known email filtering approaches. However, the performance of speech-to-text engine is largely depends on speaker, speaking style, ambient environment and language. Because of the higher error rate, this approach is still far away to become a commercially viable way to filter voice spam messages.

Collaborative Approach. Google Voice [32] had a feature to report calls as spam and block future calls from that number. This is a reactive approach requiring spam call to be received by a user and then block that number. However, this approach has few drawbacks to be applicable in telecommunication service provider network: 1) what will happen if the spam message is generated from a spoofed number, every time a new telephone number is used to send a spam message; 2) the current generation of hardphones do not provide any button to send feedback about received spam calls; 3) It is based on inferring spoken words and thus suffers from same drawback as discussed above; 4) there is no study as what will happen if the message content itself mutates (i.e., spam messages use different accents or male/female speakers) making it difficult to infer spoken words.

Content analysis. The *V-Priorities* [33] system developed by Microsoft is explored to filter spam calls. V-Priorities works on three levels: first, analysis examines the prosody – rhythm, syllabic rate, pitch, and length of pauses – of a caller's voice; secondly, rudimentary word and phrase recognition is done to spot target words that could indicate the nature of a call; and finally, at the third level analysis involves metadata, such as the time and length of a message. The voice content analysis does not require maintenance of caller's call history and remains independent of signaling. However, this approach suffers from scalability issue since it is difficult to monitor hundreds of voice streams simultaneously. The real-time content analysis is an exceedingly difficult task. By the time, calls are analyzed to be spam calls, it has already affected the receiver (human recipient or voice mailbox). The prosody analysis of machine generated voice may give different results compared to human generated voice. As mentioned earlier, inferring spoken words makes it error-prone and its success largely depends upon user, its ambient environment and language.

Black/Whitelists, trust and reputation system. The unwanted callers and domains are blacklisted so that their future calls can be filtered as spam calls. Whereas, the known callers are put in a whitelist and the calls from such callers are given preference by allowing them to go through. The trust and reputation system is used in conjunction with black/whitelists. The *social network* mechanism is used to derive a reputation value for a caller. Dantu et al. [26] use the Bayesian algorithm to compute the reputation value of a caller based on his past behavior and callee's feedback. Rebahi et al. [59] derive caller's reputation value by consulting SIP repositories along the call path from call's source to its destination. As an anti-spam solution, Sipera's IPCS [69] also relies on caller's reputation value. These solutions can block the spam call during the call setup phase itself. However, the derivation of caller's reputation value requires building a social network; the notion of user's feedback requires modification of SIP clients and an extension of SIP protocol [50]. The construction of whitelist suffers from the *introduction problem* and the calculation of reputation value is vulnerable to "bad-mouthing attacks" where malicious users may collude and provide unfair ratings for a particular caller. Furthermore, these schemes rely on caller's identity which can be spoofed.

Call duration-based Approach. Sengar et al. [66] observed the significance of call duration in spam detection and raised a fundamental question as how small it could be for normal conversations. Their proposed statistical approach lacks the consideration of those calls that are hidden behind a firewall, SBC or B2BUA agents. Later, Balasubramaniyan et al. [16] used the call duration to develop call credentials where a caller provides a call credential to the callee when he makes a call. However, a spammer could set up at least two accounts to build call credentials by calling each other and then later on use these trusted accounts to launch spam attacks.

Recently, Wu et al. [79] proposed a spam detection approach involving user-feedback and semi-supervised clustering technique to differentiate between spam and legitimate calls. However, the current generation of telephone sets do not provide an option to give feedback of a call to service provider's system. Sengar et al. [67] used callers calling behavior (day and time of calling, call duration etc.) to detect an onslaught of spam attack. However, it is difficult to capture calling pattern for each of the subscribers and secondly, being an after the fact method, by the time we detect spam attack many of the subscribers are already affected by the spam.

2.4 Voice Message Signature Construction

This section provides technical details as how a voice message is recorded on the media server and then how we can extract some specific characteristic features that later on can be used to construct a signature of the deposited message.

2.4.1 Recording of a Voice Message

As a telephone subscriber talks over phone, the telephone device captures audio using microphone. The captured audio signals are then sampled and digitized. This uncompressed digital data is passed to encoder to produce compressed frames. Now the frames are ready to be packetized as RTP packets before transmission over the network toward the other end i.e., voice mailbox or callee. As the RTP packets are received on the media server, the RTP payload is decoded. The decoded content can be stored as *Waveform Audio File Format* (commonly known as .wav file) for later replay. The recorded .wav file is send to the user account (i.e., voice mailbox) on SMTP server for storage. That later on can be retrieved as an email attachment using the email client or replayed on the media server to listen using the telephone.

2.4.2 Visual Representation of a Voice Message

Now assume that a telemarketer has left a voice message in one of the callees voice mailbox saying:

> *Take off those unwanted pounds - without strict diets. Just because you live a busy life doesn't mean you can't lose weight. Look and feel 20 years younger. You will Love how it makes you feel. Please give us a call now at 777 666 5555*

When we analyze the recorded .wav file, the Figure 2.4 shows visual representation of human speech vibrations in the form of *waveform* and *spectrogram.* At the top, the waveform tracks variation in pressure as a function of time for a given point in space. Although we can learn quite a lot by a visual inspection of a speech waveform, it is impossible to detect individual speech sounds from waveforms because speech consists of vibrations produced in the vocal tract. The vibrations themselves can be represented by speech waveforms. To read the *phonemes* in a waveform, we need to analyze the waveform into its frequency components i.e., a spectrogram which can be deciphered (the bottom of Figure 2.4). In the spectrogram the darkness or lightness of a band indicates the relative amplitude or energy present at a given frequency.

2.4.3 Silence Removal From Deposited Voice Message

In our spam content analysis, we are interested in only voiced portions of the deposited message. Therefore we need a method to remove all silence periods and segment the deposited message in voided segments. We use a method based on two simple audio features, namely the *signal energy* and the *spectral centroid.* In order to extract the feature sequences, the signal is first broken into non-overlapping short-term-windows (frames) of

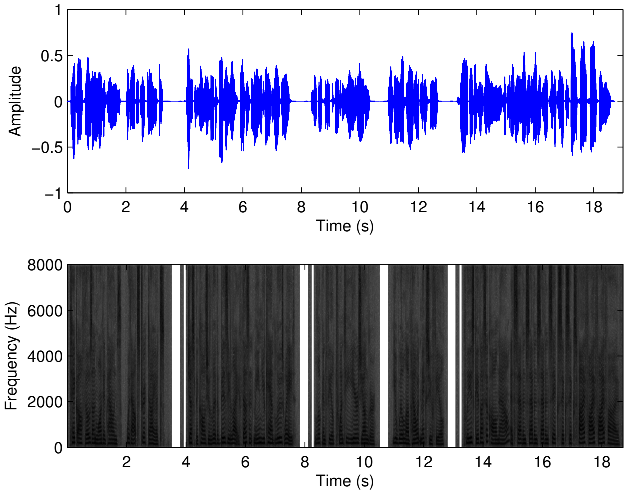


Figure 2.4: Speech Waveform and Spectrogram (US Female Speaker)

50 msec. length. For each frame, the two features, described below, are calculated, leading to two feature sequences for the whole deposited voice message.

Signal Energy: Let's assume that the deposited voice message's i^{th} frame has N audio samples $x_i(n), n = 1, 2, \ldots.., N$. The i^{th} frame energy is calculated as:

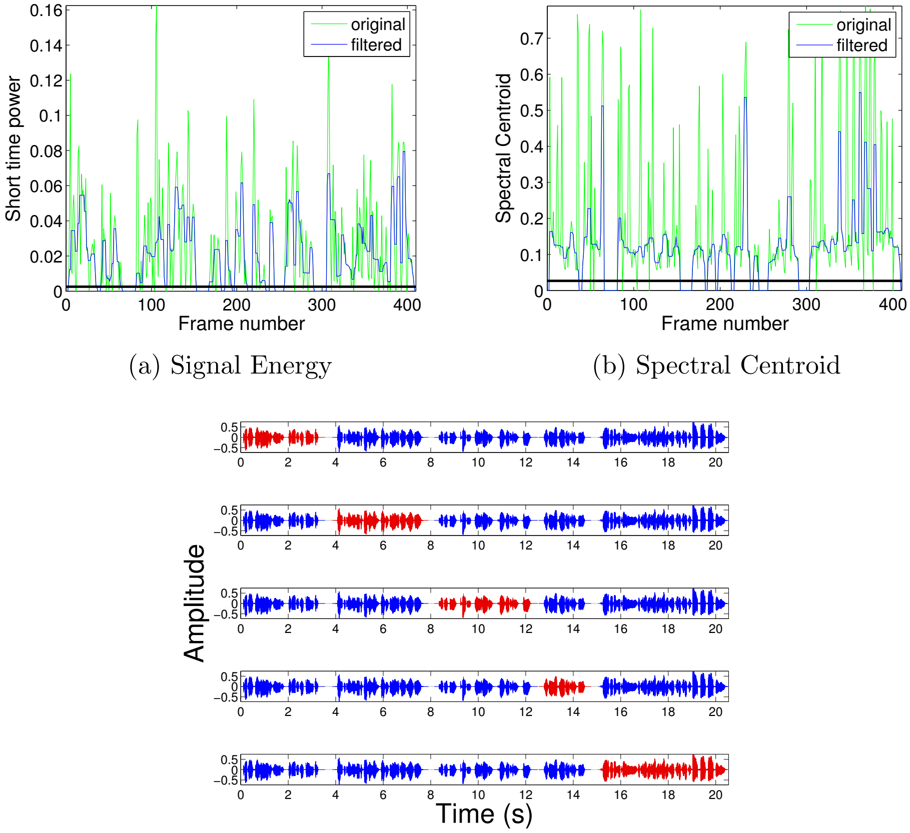
$$E(i) = \frac{1}{N}\sum_{n=1}^{N}|x_i(n)|^2 \tag{2.1}$$

Spectral centroid: The spectral centroid, C_i , of the i^{th} frame is defined as the center of gravity of its spectrum

$$C_i = \frac{\sum_{k=1}^{N}(k+1)X_i(k)}{\sum_{k=1}^{N}|X_i(k)|^2} \tag{2.2}$$

where $X_i(k)$, k=1,2,......,N is the Discrete Fourier Transform (DFT) coefficients of the i^{th} short-term frame, where N is the frame length.

Estimating two thresholds – $T1$ and $T2$, the two feature sequences are compared with their respective thresholds. The voiced segments are formed by successive frames for which



(a) Signal Energy

(b) Spectral Centroid

(c) Voiced Segments

Figure 2.5: Detected Voiced Segments from a Deposited Voice Message

respective feature values are larger than their thresholds. The detailed description of the method can be found in [31]. We use same example spam message recorded by Crystal, a US native English speaker and apply silence removal method. Figure 2.5 (a) and (b) show energy and spectral centroid sequences and its threshold values. The detected voice segments are shown in Figure 2.5 (c). These individual voiced segments serve as fundamental units to build our spam detection methodology.

2.4.4 RASTA-PLP Spectrogram Characterization

As the first step towards comparing two voiced segments, Short-time Fourier transform (STFT) can be adopted. Using STFT features, the sinusoidal frequency and phase content of local sections of a signal as it changes over time, can be determined. Since STFT, similar to most of speech parameter estimation techniques, is easily influenced by the frequency response of the speech channel, e.g. from a telephone line, we use another popular speech feature representation known as RASTA-PLP, an acronym for Relative Spectral Transform - Perceptual Linear Prediction. PLP is a speech analysis technique for warping spectra to minimize the differences between speakers while preserving the important speech information [37]. RASTA was proposed to make PLP more robust to linear spectral distortions. RASTA applies a band-pass filter to the energy in each frequency subband to remove any constant offset resulting from steady-state spectral factors of the speech channel and to tolerate short-term noise variations [36]. After a deposited message is segmented to voiced segments, RASTA-PLP spectra for all voiced segments of the voice message is calculated. For each spam voice message, its RASTA-PLP spectral matrices, corresponding to its voiced segments, are stored in the spam signature database. Fig. 2.6 shows the RASTA-PLP spectrograms for the first voiced segment ("Take off those unwanted pounds without strict diets.") of two deposited messages from different speakers, Diane (Female English speaker) and Dallas (Male English speaker), with the same content.

2.4.5 Matching Process

Figure 2.7 shows the spam filtering architecture that can work as standalone and distributed collaborative manner. In standalone mode, the voice messages left by the callers are under going through the behavioral analysis and signature matching based on the locally stored signatures. However, in collaborative distributed scheme the member VoIP

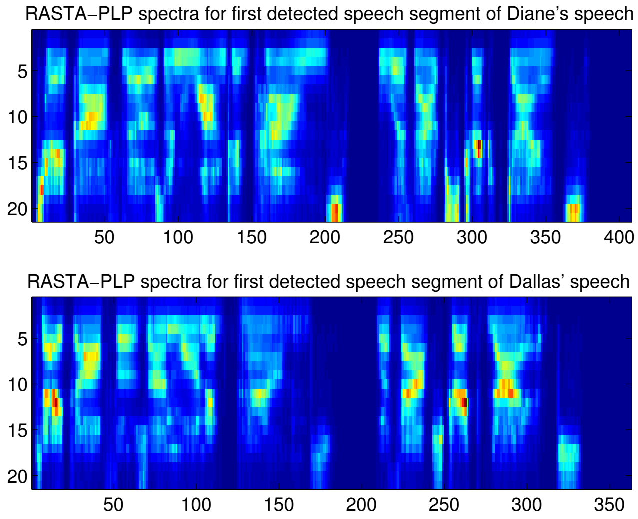


Figure 2.6: RASTA-PLP Spectral Features For The First Voiced Segment of Diane (Female) and Dallas (Male) Native Speakers

service providers can query for signature matching as per need basis and at the same time newly found spam message is made available so that it can be signaturised and used by the other service providers.

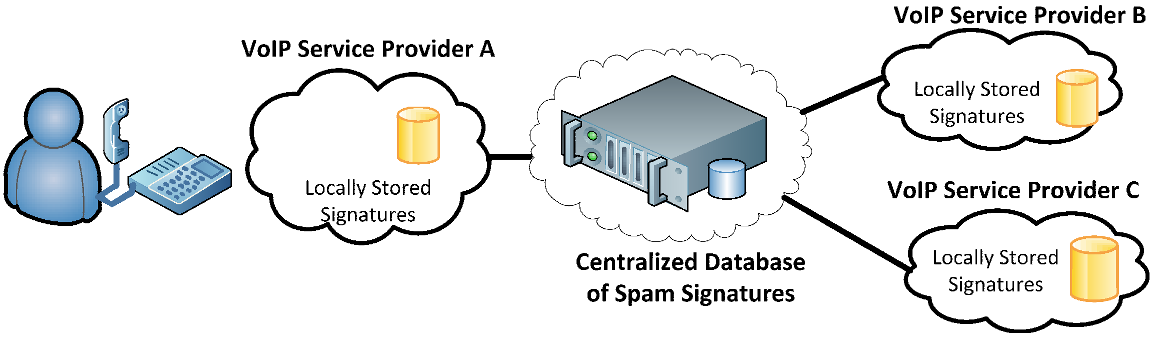

VoIP Service Provider A
Locally Stored Signatures
Centralized Database of Spam Signatures
VoIP Service Provider B
Locally Stored Signatures
VoIP Service Provider C
Locally Stored Signatures

Figure 2.7: Standalone and Distributed Collaborative Architecture

For signature matching and call behavior analysis, the newly arrived voice message is divided to voiced segments and corresponding RASTA-PLP matrices are calculated. As a first step, the database of known spam signatures is queried to find the voice spam message that has similar content to the newly arrived voice message. If the computed cosine distance between newly arrived and an already known spam message is less than a threshold, we confidently declare that a match has been found. However, in case there is no match found, then we perform call behavior analysis. Within a predefined time interval of ΔT (say ≈ 5 minutes), we segment all of the voice messages left on the media server to their voiced segments and calculate their corresponding RASTA-PLP matrices to observe how many messages are of similar content. Beyond a threshold value (say 3 messages per 5 minutes), the matched messages are considered to be a part of impending spam attack and demands further analysis. The unmatched messages are deposited to their respective user accounts (i.e., mailboxes).

2.5 Detection Methodology

To either find if the newly arrived voice message has similar content to a spam signature or observe as how many similar messages (content-wise) are recorded on the media server within a predefined time interval, a speaker independent speech recognition method is proposed. The newly deposited message is first divided to small voiced segments using the silence removal technique described in section 2.4.3. For each of the voiced segments, we create RASTA-PLP matrices. As a similarity measure, we use Dynamic Time Warping (DTW) method and calculate cosine distance for each pair of voiced segments coming from two different voice messages. Based on these partial scores for the corresponding speech segments, it is finally determined if the two voice messages are similar enough and a match is found. The next four subsections explain these phases in more details.

2.5.1 Scoring similarity between two speech segments

2.5.1.1 Constructing scores matrix

Cosine Similarity is considered here as the similarity measure between two speech segments. Calculating the cosine distance between every pair of frames from RASTA-PLP spectral matrices for two segments, the *local match* scores matrix is Constructed. The left side of Figure 2.8 shows spectrogram-like scores matrix for the first voiced segment ("Take off those unwanted pounds without strict diets.") of two speech snippets of Diane (female) and Dallas (male) native speakers. High similarity values can be seen as a dark stripe approximately down the leading diagonal in the figure.

2.5.1.2 Dynamic Time Warping (DTW)

Although two different voice segments (speaker's utterances) with same content have more or less the same sounds in the same order, the durations of each sub-segment (words and letters) may not match. As a consequence, matching between two voice segments without temporal alignment may fail. To cope with different speaking speeds and differences in timing between two segments, we use a dynamic programming method named Dynamic Time Warping (DTW) [27]. Considering a 2D space with X-axis of time frames from one segment and Y-axis of time frames from another segment, DTW tries to find the path through this 2D space that maximizes the local match between the aligned time frames. The total *similarity cost* found by DTW can be considered as a proper indication of how well these two segments match. The right side of Figure 2.8 illustrates how DTW finds the lowest-cost path between the opposite corners of the scores matrix. As we can see in the right side of Figure 2.8, path on the scores matrix follows the dark stripe depicted in the left side of Figure 2.8.

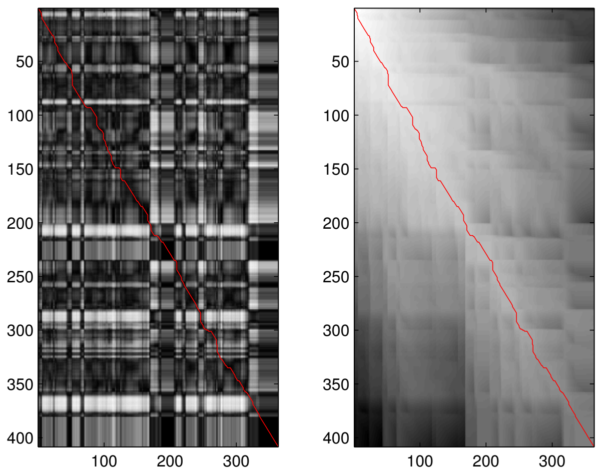


Figure 2.8: Using DTW to find similarity between constructed scores matrices for the first voiced segment of Diane's speech and Dallas's speech

Similar to other dynamic programing, bottom right corner of the *minimum-cost-to-this point* matrix returns cost of minimum-cost alignment of the two speech segments. This value as partial score, can be considered as our similarity measure. The smaller is partial score, the closer are the two corresponding segments of different voice messages. Since the value of partial score has a relationship with the size of spectral matrices (duration of voiced segments), we divided the partial score by the minimum duration of two segments to define a more comparable weighted partial score. To specify a threshold to find if two segments are similar enough, the method against many different voice messages is tested. Hence, we empirically found 10 as a proper threshold for acceptance or rejection of similarity between two segments.

2.5.2 Voice message content matching

To find if two speech messages are similar enough, weighted partial scores for all pair of corresponding segments of both messages are calculated. Comparing the weighted partial scores to the threshold value of 10 for each pair of corresponding segments, it is determined if the two segments have same content. If a certain number of corresponding segments for both messages have same content, the two whole speech message are also similar enough and a match has been found.

2.5.3 Bayesian content filter

Based on the idea of Bayesian filtering for email spam, we propose a similar method for voice spam filtering. In this method, we have a database of known spam words named spam speech database. In the training phase, the spam words are converted to speech using text-to-speech (TTS) system and stored in a spam speech database. Speech words here can be a single word, a combination of words (i.e., phrase), phone number or URL address with high spamicity. In other words, we transformed known email spam database and its probabilities to voice spam world. Since there is no speaker independent speech segmentation method (without language-specific knowledge) to perfectly segment speech messages in word level, we take an alternative approach. In our approach, entries of the spam speech database are tested against the voice message to find if the voice message includes that entry of the database. As an example, suppose Mike left a voice message, "Free mortgage consultations available now", for his friend. To check if the deposited message is spam, entries of the database are tested against this voice message. Assuming that "mortgage" is an entry in the spam speech database, that was previously detected from another speaker (Crystal) and stored in the database, we try to find if the voice message includes this speech word "mortgage". Starting from the beginning of the voice message, a frame in size of the entry of the database (speech word “mortgage”) traverses the waveform of the speech message. While the frame traversing the message, the dissimilarity of the current frame of the speech message and the speech word from database (“mortgage”) is calculated using DTW. Reaching the end of the speech message, the frame of speech message with maximum similarity is the determiner if the message includes the spam word (“mortgage”). This similarity score is compared to a threshold to find if the speech message includes that spam word. Using Bayes’ Formula and based on the number and spamicity of spam words from database that the spam message contains, we can judge if the speech message is spam or not. To justify threshold, as the most important part of this method, we have tested the method for different words and phrases in different sizes. Hence, it is empirically found that the similarity score using DTW is tightly related to the size of speech words. For example, DTW similarity score for word in size of “mortgage” is about 4.5-5, and for word in size of “777 5555 666” (as a phone number) is about 50. Therefore, the threshold is set in a dynamic way based on the size of the speech word to be tested.

2.5.4 Searching

As explained in Section 2.4, we construct two separate databases to store RASTA-PLP matrices; Spam Signature Database for spam signatures, and Spam Speech Database for spam words and phrases with high spamicity. After voice messages are left on the media server by callers, Spam Signature Database is first queried to find a match. Entries in Spam Signature Database can be organized in categories based on VoIP service providers where they have been locally stored from to speed up the search process. In case a match is not found (i.e., signature does not exist in the Spam Signature Database), entries of the Spam Speech Database are searched against the voice message to find if the voice message includes that entry of the database. After performing this search, Bayesian spam filtering is used to determine the final probability of the voice message being spam. To reduce the search time, we propose a cluster-like structure for the Spam Speech Database, where cluster heads are speech words with the highest probabilities in each cluster. For example, two clusters of the database are described here:

- Cluster 1:
    - cluster head: Viagra
    - cluster members: sex, cheap, night, www.buyviagraonline.com
- Cluster 2:
    - cluster head: Mortgage
    - cluster members: 100% free, lower interest, “555 666 7777” (phone number)

To perform a search, we start with cluster heads. If none of cluster heads matches, the voice message is classified as non-spam. If one of the cluster heads matches, we narrow our search to the corresponding cluster to consider all other relevant words in relevance order. The Baye’s Formula will take care of calculating the probability of being spam based on the number of spam segments it contains and their spamicities.

2.6 Performance Evaluation

We conduct a series of experiments to evaluate the performance of our solution. In our experiments, we left voice messages on Google voice [32] and then later on analyzed for its legitimacy and spam detection rate.

In addition to these manually deposited voice messages, three popular TTS systems are used to generate various voice messages with different speakers in different sizes. Eight speakers were selected from AT&T Natural Voices$^{®}$ TTS system [13]. Twelve speakers were selected from Cepstral engine [22], as a TTS system that makes realistic synthetic voices. Moreover, ten speakers were selected from PlainTalk [77], the advanced built-in TTS technology of Mac OS. These thirty selected speakers have different accents (such as American, British or Indian English), different sex (male and female) with ages ranging from 10 to 60 years old.

2.6.1 Arrival of Same Content Voice Messages

This is a most generic spam attack scenario where a spammer repeatedly sends same spam message to many of the subscribers at the same time. If a newly arrived voice message matches with any of the signatures stored in the database, the message is categorized as a spam message.

Ten totally different text messages with different size and content were converted to voice messages spoken by the thirty above mentioned different speakers to form 10 different sets of 30 voice spam messages with same content. All of these 300 different voice messages were first segmented to small voiced speech segments. Then the RASTA-PLP spectral matrices for all segments were calculated as well. Randomly selecting 3 voice messages of different speakers out of total 30 messages from each set of speech messages (with same content), a database with 30 entries were generated. For each subexperiment, this process were repeated 10 times and each time one voice message from one of 10 sets is selected to check if it is spam. Iterating the sub-experiment 10 times forms a complete experiment. To take average, the complete experiment was conducted three times and the results are summarized in Table 1:

In our experiments we find that our speaker independent spam detection algorithm can detect similar content message with 91% accuracy while generating 0.67% of false positive

Table 2.1: False Positive and False Negative rates of Voice Message Content Filtering

Case	Correct	False Positive	False Negative
#1	91	0	9
#2	87	1	12
#3	95	1	4

and 8.33% false negative rates.

However, if the newly arrived message does not match with any of the spam signatures stored in the database, in that case we record its signature and observe if this signature matches with any of the future deposited messages within a predefined time interval of ΔT ($\simeq$ 5 minutes). The similar message count beyond a threshold value within a time period can be categorized as an impending spam attack and needs further analysis.

We are aware that there are some legitimate applications that can generate calls in bulk. For example, it is possible that an emergency response system within a company, city or college may call many of the telephone numbers at the same time alarming about some untoward incident. It is also possible the a credit card company may send a prerecorded generic message at a particular time to many of its members regarding fraudulent activity in their accounts. In all such cases, there will be a number of matches (beyond a defined threshold value) within a predefined time interval ΔT and therefore possibly be labeled as spam messages without delivering to their respective mailboxes.

These legitimate call scenarios may cause false positives. To avoid such false positives, before labeling these legitimate voice messages as spam, our Bayesian content filtering method is used to calculate the probability of being spam for one of the newly deposited voice message. Moreover, if we are provided with calling number and originating source IP address used by these bulk call applications in advance then combining the SIP signaling information and content filtering approach can also avoid such false positives.

2.6.2 Hiding Spam Words/Phrases Within A Voice Message

In this set of experiments, the Spam Speech Database was built with 137 entries in five clusters; Employment, Financial (Business and Personal), Marketing, Medical and Calls-to-Action. In addition to having one or more cluster heads, each cluster has several cluster members converted from email spam trigger words/phrases, and some special elements, such as URL address, email address and phone number, that have been extracted from our Spam Signature Database. Table 2 summarizes the details of the clusters in our Spam Speech Database.

Table 2.2: Cluster details of the Spam Speech Database

Cluster	# of cluster members	# of special elements
Employment	24	4
Financial (Business)	15	2
Financial (Personal)	18	2
Marketing	35	5
Medical	18	3
Calls-to-Action	9	2

To evaluate the efficiency of the proposed Bayesian based content filtering method, we recorded 30 various voice messages in different size from mentioned speakers with various accents, sex and ages. This set of voice messages includes three types of voice messages as follows:

1- *Spam voice message*: a voice message that includes at least one cluster head and either at least one special element or significant numbers of relevant cluster members. This type of voice messages should be classified as spam.

2- *Doubtful voice message*: a voice message that includes at least one cluster head but neither special element nor significant numbers of relevant cluster members. This type of voice messages should be classified as non-spam. In other words, a couple relevant words/phrases from a cluster of the Spam Speech Database do not classify a deposited message as spam. There have to be enough words/phrases with a high spamicity to outweigh the rest of the voice message that includes words/phrases with a low spamicity. For example, a voice message from your spouse discussing taking out a second mortgage on the house to finance some kitchen remodeling should not be misclassified as spam.

3- *Non-spam voice message*: a voice message that does not include even one cluster head. This type of voice messages should be classified as non-spam.

Our Bayesian based spam detection method was performed to classify this set of voice messages. The results show that the method correctly classified 83.33% of voice messages while 13.33% of either non-spam or doubtful voice messages were misclassified as spam and 20% of spam voice messages were not detected. We further looked into the results and details of the method to find the causes of these false positives and false negatives. It is discovered that the problem arises when voice messages are deposited by speakers with accents rather than US English, such as British or Indian English. Since the entries of our Spam Speech Database were converted by Crystal, a US native English speaker from spam email world, the dissimilarity score computed by our DTW based algorithm is not dependable enough to compare small-size speech words of speakers with different accents.

2.7 Conclusion

Although there are very few reported incidents of voice spam today, with the growth of VoIP and its openness, the voice spam could become a big threat in near future. At the heart of the problem lies the fact that a spammer can send unsolicited advertisements and messages cheaply or free of cost while being remain anonymous. Unfortunately, many of the mechanisms which work for email spam fail completely in the context of VoIP. Most of the previous solutions for voice spam problem are proposed to distinguish a legitimate subscriber from a spammer using SIP signaling messages. Instead of analyzing the SIP signaling messages to identify the spammer, in this chapter we propose a speaker independent speech recognition scheme for content filtering to avoid spam message deposition on the subscribers' voice mailboxes. Our proposed solution not only protects subscriber's voice mailboxes from spam messages, but also provides a way to identify spamming sources. Our work is evaluated against realistic case scenarios and performing real-world experiments. The experimental results show that our spam filtering approach is speaker independent, computationally efficient, and very scalable that can successfully classify a voice message into spam with 91% accuracy while generating 0.67% false positive and 8.33% false negative rates.

Chapter 3

Exploitation of Imprecise CODEC Sampling Rate: A Method for Remote IP Phone Identification

Voice over IP (VoIP) telephony is emerging as an alternative to traditional public switched telephone network (PSTN). It is a cost effective and convenient way of communication, and provides a platform for creating many new services that cannot be envisaged using traditional telephone systems. IP telephone service providers are moving quickly from low-scale toll bypass deployments to large-scale competitive carrier deployments, thus giving an opportunity to enterprise networks for supporting a less expensive, single-network solution rather than multiple separate networks. Broadband-based residential customers also switch to IP telephony because of its convenience and cost effectiveness. In contrast to a traditional telephone system (where the end devices are dumb), the VoIP architecture pushes intelligence toward the end devices. This flexibility coupled with the growing number of subscribers becomes an attractive and potential target to be abused by malicious users.

Over the past decade, most VoIP communications use the Session Initiation Protocol (SIP) [42] as their signaling protocol. In spite of being widely used, SIP does not provide any effective standards for calling party identification in a SIP session. Using VoIP client software such as Yate [80], softclients are allowed by many VoIP providers to claim arbitrary caller IDs. There are also many fake ID providers such as TeleTurd [8] that allow their customers to claim any caller ID. Thus, caller IDs are vulnerable to be easily spoofed. Moreover, interworking SIP with PSTN has ultimately reduced the security of Caller ID systems. Caller ID spoofing can be used for many malicious attacks, including vishing attacks, illegal robocalls, and swatting. In addition to caller ID spoofing, there are many other VoIP attacks such as spamming, DoS attacks, and call flooding attacks, in which the faked identities of calling devices are also widely used. Wangiri (one ring and cut in Japanese) fraud, which is a well-known voice scam, also abuses the lack of caller ID authentication in voice over IP delivery [63]. In this scam, the fraudster leaves missing calls on a huge number of victims phone numbers motivating the curious victims to call back a premium rate number (PRN) to financially benefit a certain share of the call revenue.

For example, Japan where the VoIP market is more mature than USA, has witnessed some recent voice spam attacks. The SoftbankBB, a VoIP service provider with 4.6 million users, has reported three incidents of spam attacks within its own network [74]. These incidents include unsolicited messages advertising an adult website, scanning of active VoIP phone numbers, and requesting personal information of users. Similarly, Columbia University at New York experienced a voice spam attack, with someone accessing the SIP proxy server and “war dialing” numerous IP phone extensions [76].

Many of such attacks are possible because VoIP clients are flexible enough to manipulate the caller ID, a number of calls can be generated simultaneously, and also there is no limitation on how many active sessions can be maintained. There have been a number of solutions proposed to address voice spam, caller ID spoofing, and flooding attacks. However, based on simple assumptions or asking too much network overhaul, these previous defenses are limited to work well in practice.

3.1 Overview of Proposed Approach

Now we discuss two fundamental problems in VoIP. The first problem arises when an attacker has compromised the telephone subscribers' credentials to initiate fraudulent or malicious activities. VoIP customers can assign different devices to their accounts and specify if they may sometimes use softclients to place calls. As a part of *device management*, VoIP service providers are aware of what types of calling devices are used and deployed at the customer site and accordingly the *call feature profiles* are applied on those devices. The device management is mainly used by the service provider for auto-provisioning that provides easy installation, remote management of devices, call services and dynamic update of core firmware of the devices. *However, it should be noted that there is no way to bind user account credentials with its calling device(s).* As a result, once user account is compromised, a malicious software running on a general purpose PC can mimic any device behavior and launch attacks from anywhere around the world using the Internet connection.

The second problem is encountered when the call-id is faked by unscrupulous ID providers to initiate various attacks. As caller-id can be easily spoofed in VoIP environments, many security threats can be imposed by unsolicited communications such as voice spam, voice phishing (vishing) and flooding attacks. Such VoIP attacks are mainly launched from PC-based malicious programs pretending to be IP hard phones. *There is no way for a callee site to verify the truthfulness of the calling device used.* Therefore, caller-id spoofing cannot be prevented and consequential attacks such as vishing attacks can succeed.

Today VoIP service providers have neither any solution to identify compromised subscriber (i.e., user) accounts in their domain nore to prevent unsolicited calls originated by faked caller-id in other domain. In this chapter, we aim to propose a system to identify the type of calling devices that can address the two above mentioned problems. Different SIP devices have different clocks; IP hardphones not only have higher-quality clocks than PC-based soft clients, but also have different clocks themselves. Hence, we utilize clockskew as a metric to identify the type of calling devices.

By analyzing the calling device's voice stream (i.e., RTP packets) based on the negotiated codec between call parties, we first show how to locally measure the clockskew as the slope of the offset between the time of receiving RTP packets in the identification module and the timestamp inside RTP packets. We then show how to remotely measure clockskew based on statistical measures of the offset points such as variance, skewness, and kurtosis. Our proposed fuzzy based system can remotely identify the type of calling devices based on the statistical measures.

Through a series of real experiments, we show that our proposed fuzzy system can be used in different places of a call path as a loadable module to accurately distinguish between hardphones and softclients as well as identify make/model of hardphones with the detection accuracy rate of 95%. As our system analyzes RTP packets in real time and is based on their timestamps and the time of capturing them at the identification module, malicious softclient cannot pretend to be hard phone with a high quality clock replaying spoofed RTP packets.

Our system can be placed as a loadable module in the calling side to identify compromised subscriber accounts and block any fraudulent activities. Our system can also be placed as a loadable module in the callee side to prevent caller-id spoofing. In the latter case, the system in the callee side identifies the calling device and asks the calling side (i.e., the original SIP proxy server) to verify the truthfulness of the device used and check if the caller-id is faked by another unscrupulous ID providers. Service providers can provide different policy settings for their users to deal with such cases, either giving a warning to the callee or denying the call.

The significance of our approach can be summarized as follows: (1) it works in a passive mode without any interaction with the caller—no need to resolve puzzles or CAPTCHAs; (2) it does not rely on the capability of the callee's SIP UAs and their feedbacks—no need to derive or rely on the trust value of the caller; (3) it does not require for overhauling of network infrastructure or protocols; and finally, (4) this approach is not specific to a particular type of VoIP attacks, instead it can be used to prevent many known and unknown attacks, in which hackers use and recruit zombies over the Internet to behave like a SIP user agent and make unsolicited fraudulent calls.

The remainder of the chapter is structured as follows. Section 3.2 describes the SIP-based IP telephony, the VoIP architecture, and the threat models. Section 3.3 presents the VoIP quality of service requirements. Section 3.4 details our fuzzy rule-based system design, including local and remote measurements of sampling impreciseness. Section 4.12 evaluates the efficiency of our proposed system using testbed experiments. Section 3.6 surveys related work. Finally, Section 3.7 concludes the chapter.

3.2 Background

3.2.1 SIP-based IP Telephony

The Session Initiation Protocol (SIP) [42], belonging to the application layer of the TCP/IP protocol stack is used to set up, modify, and tear down multimedia sessions including telephone calls between two or more participants. SIP call control uses the *Session Description Protocol* (SDP) [35] for describing multimedia session information.

3.2.1.1 SIP Architectural Components

SIP-based telecommunication architectures have two kinds of elements: end devices referred to as user agents (UAs) and SIP servers. Irrespective of being a software or hardware phone, UAs combine two sub-entities: the connection requester referred as the *user agent client (UAC)* and the connection request receiver referred to as the *user agent server (UAS)*. Consequently, during a SIP session, both UAs switches back and forth between UAC and UAS functionalities. RFC 3261 [62] describes four types of SIP implementation dependent logical servers: *Location Servers*, *Redirect Servers*, *Registrar Severs* and *Proxy Servers*.

3.2.1.2 SIP Messages and Operations

Influenced by two widely used Internet protocols: Namely, the *Hypertext Transfer Protocol (HTTP)* [28] and the *Simple Mail Transfer Protocol (SMTP)* [44], SIP messages consisting of request-response pairs are exchanged for call set up, from six kinds consisting of `INVITE`, `ACK`, `BYE`, `CANCEL`, `REGISTER`, and `OPTIONS` - each identified by a numeric code according in RFC 3261 [62].

Now as an example, we discuss a typical message flow for a call setup between UA-client UA-A and UA-server UA-B. Assuming that the two UAs belong to different domains with their own proxy servers, UA-A's proxy server uses its Domain Name Service to locate a proxy server for UA-B. After obtaining the IP address of UA-B's proxy server, UA-A's proxy server sends an `INVITE` request to the latter with UA-B's name. In response, UA-B's proxy server consults a location service database to find out the current location of UA-B, and forwards the `INVITE` request to the UA-server residing on UA-B's SIP phone.

Exchanging `INVITE`, `200 OK` and `ACK` messages completes the three-way handshake and establishes a SIP session. Then, a SDP compliant set of parameters are exchanged in SIP message bodies, and lastly establishes a RTP stream to exchange audio data.

SIP proxy servers have no media capabilities and only facilitate the two end points (i.e., IP telephones) to discover and contact each other through SIP signaling. Once the end points have been located, the media flows directly between user agents (generally it is not allowed as we discuss later) without going through proxies using a path independent of the one used by SIP signals. At the end of the call, one party hangs up, resulting in that party's agent sending a `BYE` message to terminate the session and receives a `200 OK` response from its counterpart. This example shows the basic functionality of SIP, described in more detail in RFC 3261 [62].

3.2.2 VoIP Architecture and The Threat Models

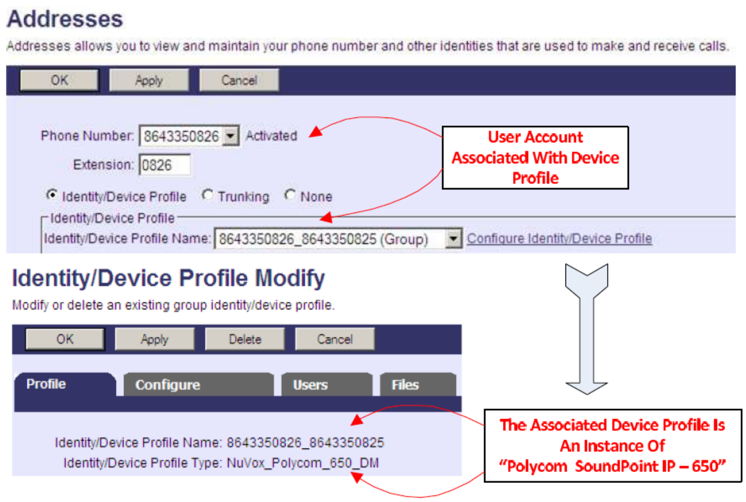
In today's IP telephony world, the VoIP service providers (such as Vonage, AT&T Callvantage, and ViaTalk) operate in partially closed environments and are connected to each other through the public telephone network. In a partially closed environment, the VoIP service providers allow only their own authenticated subscribers (coming through the IP network) to access SIP proxy servers. The authentication of call requests is possible because user accounts (containing authentication credentials, subscribed call features, and policy etc.) are stored locally within the VoIP service provider's SIP servers. In this scenario, the fraudulent activities are possible by misbehaving subscribers or by compromising subscriber credentials. Once subscriber credentials are compromised, a hacker can launch various VoIP attacks over the Internet by using malicious software (acting as a SIP UA) installed on an already compromised PC.

Recently we are witnessing a large demand for SIP trunks. A SIP trunk is a service offered by VoIP service providers permitting business subscribers to reach beyond the enterprise network and connect to the PSTN through IP-based connections. The use of SIP trunks along with IP-PBX presents an alternative to replace the traditional PRI and analog circuits. Generally most of the SIP trunks are set up without authentication (also no registration is involved). The IP address of the customer's edge device is put into access control list along with few other call related constraints (e.g., the number of established sessions and the number of call setup requests within a predefined time window). Only few of the service providers use TLS or IPSec connections between edge devices to secure SIP signaling. In this scenario, VoIP attacks can be launched from the enterprise network or by a man-in-the-middle where SIP signaling is transported over the Internet in plain-text without any encryption.

SIP does not provide any effective mechanisms for calling party identification in a SIP session. Thus, caller-id can be easily faked or spoofed to initiate various attacks such as spamming, vishing, and flooding attacks. As some VoIP service providers allow their costumers to claim arbitrary caller-ids, attackers can subscribe to such VoIP services and initiate VoIP attacks using PC-based soft clients. Moreover, as fake ID providers can establish VoIP connections with different telephone carriers, attackers are able to connect to victims and initiate spoofing attacks using any type of phones.

3.2.3 VoIP Telephone Subscriber Device Management

At the customer (VoIP telephone subscriber) site there are many types of access devices such as analog terminal adapters (ATAs), IP phones, integrated access devices (IADs), and IP PBX that need configuration profiles, firmware, and other files to provide proper operations of call services. Generally, the VoIP service providers manage and control device configuration centrally from their networks to ease the deployment and provisioning



Addresses

Addresses allows you to view and maintain your phone number and other identities that are used to make and receive calls.

Phone Number: 8643350826 Activated

Extension: 0826

● Identity/Device Profile ○ Trunking ○ None

Identity/Device Profile

Identity/Device Profile Name: 8643350826_8643350825 (Group) Configure Identity/Device Profile

User Account Associated With Device Profile

Identity/Device Profile Modify

Modify or delete an existing group identity/device profile.

Identity/Device Profile Name: 8643350826_8643350825

Identity/Device Profile Type: NuVox_Polycom_650_DM

The Associated Device Profile Is An Instance Of "Polycom SoundPoint IP – 650"

Figure 3.1: Screenshots (From Broadsoft Softswitch) Showing Relationship Between Subscriber and Its Device Type

of customer end devices. Here we take a real world example of a VoIP subscriber whose call features are managed by Broadsoft softswitch[1].

The top part of the Figure 3.1 shows a user telephone number and its association with a specific device profile name. However, this device profile name is an instance of the "Polycom Soundpoint IP - 650" phone device type (named as NuVox Polycom 650 DM). There is a long list of access device types that are supported by the softswitch. When a user account (i.e., a telephone number) is provisioned in the system, its associated device type relationship is also established. It should be noted that this relationship is to ease the device management from service providers' perspective to offer advanced call features available on that device. However, this relationship does not forbid a user to make *simple* calls from any other (unsupported) devices available in the market or PC-based soft-clients (SIP UAs) as long as the telephone number and authentication credentials are valid.

[1] BroadSoft is deployed in more than 450 telecommunications service providers' networks and serves 15 of the top 25 largest telecommunications carriers.

3.2.4 Binding Subscriber With Its Own Device Profile

If we look at past and current events to identify trends and changes in attacks and targets, then we find that the most imminent threats to VoIP deployments are either bot-generated attacks, as they have been a constant threat to data networks, or caused by faking the identity of calling devices. The success of bot-generated attacks depends upon three main factors: first, the attacker remains anonymous; second, the vastness and diversity of the army of bots; and finally, the bot's distribution over the Internet.

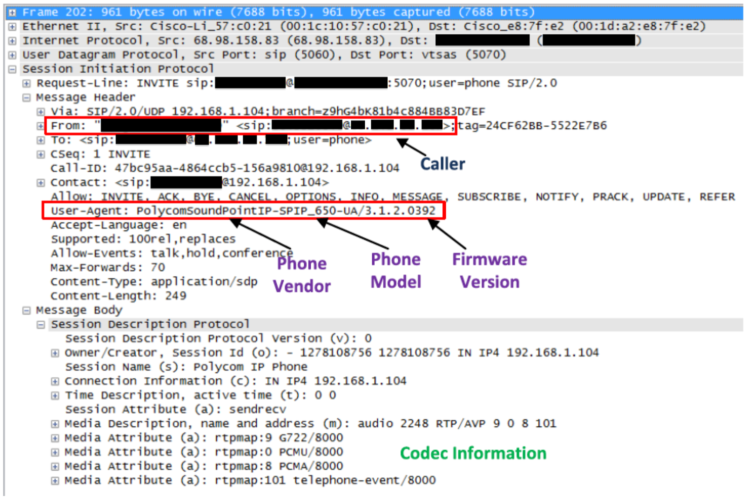
As hacking into hardphones and compromising their firmware is difficult, attackers usually take two comparatively easier approaches: guessing password (either by brute force, social engineering, or man-in-the-middle attack) and bypassing service provider's network (by subscribing to either fake ID providers or flexible VoIP service providers to claim arbitrary caller IDs). Either ways, attackers use malicious software installed on compromised PCs to launch various types of attacks (such as Caller ID spoofing, spam calls, and DoS attacks). Now two following questions, which we aim to address in this chapter, arise:

- *Is it possible for a VoIP service provider to detect if any of its subscribers is using a calling device other than configured against his user account to place a call?* The answer to this question can help service providers to block any outgoing calls that are originated from an unauthorized calling device (i.e., a device that is not associated with the user account).

- *Is it possible for a VoIP service provider to detect if an incoming call to any of its subscribers is originated from an unscrupulous service provider?* The answer to this question can help service providers to block any incoming calls with faked/spoofed caller-IDs. Our insight is that if a call is placed using an unauthorized device and has not been blocked by its original service provider, it is likely originated from an unscrupulous service provider.

3.2.5 Is `INVITE`'s User-Agent Field Enough?

Figure 4.2 shows `INVITE` message structure captured using Wireshark [24]. The SIP `INVITE` message consists of two parts, the header fields and a message body separated by a blank line. The session description such as media type, codec and sampling rate are contained in `INVITE`'s message body. The connection information field (i.e., `c=`) contains media connection information such as media's source IP address. Similarly, the media information field (i.e., `m=`) contains media type and the port number.



```
⊞ Frame 202: 961 bytes on wire (7688 bits), 961 bytes captured (7688 bits)
⊞ Ethernet II, Src: Cisco-Li_57:c0:21 (00:1c:10:57:c0:21), Dst: Cisco_e8:7f:e2 (00:1d:a2:e8:7f:e2)
⊞ Internet Protocol, Src: 68.98.158.83 (68.98.158.83), Dst: [redacted] ([redacted])
⊞ User Datagram Protocol, Src Port: sip (5060), Dst Port: vtsas (5070)
⊟ Session Initiation Protocol
  ⊞ Request-Line: INVITE sip:[redacted]@[redacted]:5070;user=phone SIP/2.0
  ⊟ Message Header
    ⊞ Via: SIP/2.0/UDP 192.168.1.104;branch=z9hG4bK81b4c884BB83D7EF
    ⊞ From: "[redacted]" <sip:[redacted]@[redacted].[redacted].[redacted].[redacted]>;tag=24CF62BB-5522E7B6
    ⊞ To: <sip:[redacted]@[redacted].[redacted].[redacted].[redacted];user=phone>
    ⊞ CSeq: 1 INVITE
      Call-ID: 47bc95aa-4864ccb5-156a9810@192.168.1.104
    ⊞ Contact: <sip:[redacted]@192.168.1.104>
      Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, INFO, MESSAGE, SUBSCRIBE, NOTIFY, PRACK, UPDATE, REFER
      User-Agent: PolycomSoundPointIP-SPIP_650-UA/3.1.2.0392
      Accept-Language: en
      Supported: 100rel,replaces
      Allow-Events: talk,hold,conference
      Max-Forwards: 70
      Content-Type: application/sdp
      Content-Length: 249
  ⊟ Message Body
    ⊟ Session Description Protocol
        Session Description Protocol Version (v): 0
      ⊞ Owner/Creator, Session Id (o): - 1278108756 1278108756 IN IP4 192.168.1.104
        Session Name (s): Polycom IP Phone
      ⊞ Connection Information (c): IN IP4 192.168.1.104
      ⊞ Time Description, active time (t): 0 0
        Session Attribute (a): sendrecv
      ⊞ Media Description, name and address (m): audio 2248 RTP/AVP 9 0 8 101
      ⊞ Media Attribute (a): rtpmap:9 G722/8000
      ⊞ Media Attribute (a): rtpmap:0 PCMU/8000
      ⊞ Media Attribute (a): rtpmap:8 PCMA/8000
      ⊞ Media Attribute (a): rtpmap:101 telephone-event/8000
```

Figure 3.2: Structure of an INVITE Message

The `From` header field contains the SIP URI of the caller who is originating the call request. Similarly, the `User-Agent` header field contains information about the UAC originating the request. It is not a mandatory field, nevertheless most of the IP phones and SIP UAs available in the market implement this field. However, VoIP service providers cannot rely on this field as it can easily be spoofed or suppressed by the malicious clients.

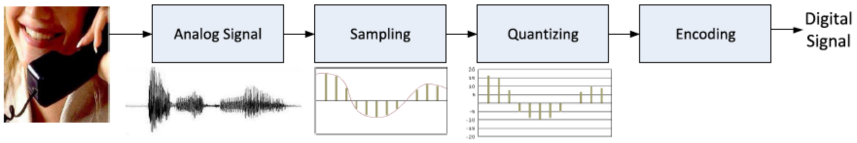


Figure 3.3: Analog to Digital Conversion

As we show later, we develop a method that can verify the truthfulness of this information if presented by a client. Even if the client has suppressed this information intentionally, still we can determine the device type originating the call.

3.3 VoIP Quality of Service

Being the best-effort service, a key challenge for the Internet is to guarantee quality of service (QoS) for realtime applications such as voice. As for a voice application, the sender intends to transmit the audio packets in a regular interval and the receiver wants to play out them with the same regular interval. However, in real world, there are various types of delays introduced at the sender-side, network, and also at the receiver-side. The sender-side delay includes packetization delay which depends upon the particular codec employed. The network delay includes the fixed propagation delay, the network interface transmission delay, and the queuing delays of routers and switches along the path. Of these three, the queuing delay is the most significant resulting in the variance in inter-packet arrival time (jitter) at the receiver. At the receiver-side there is always a finite buffer to absorb jitter and make the packets be played out in regular interval. Now we look more closely into the packetization delay as our proposed approach is based on it.

3.3.1 Audio Packetization

Figure 3.3 shows the basic three-step process to convert analog signals to digital signals, namely sampling, quantization and encoding. The speaker's audio utterances are captured as analog signals that are sampled, digitized and stored into an input buffer. Once a fixed number of samples are collected into the buffer, it is made available to the voice application. This imposes some delays because the first sample in a frame is not available until the last sample has been collected. The interactive application such as telephone call selects the buffer size closest to the codec frame duration. Generally, it is either 10, 20 or 30 ms to reduce the delay.

The uncompressed audio frame returned from the capture device can have a range of choices. For example, it can have 8, 16, or 24 bit resolution using linear, μ-law or A-law quantization at rates between 8,000 and 96,000 samples per second and either in mono or stereo. The uncompressed captured audio frames are passed to the encoder for compression. Depending on the codec choice, state may be maintained between frames that must be made available to the encoder along with each new frame of data. Some codecs produce fixed-size frames as their output; others produce variable-size frames. Those with variable-size frames commonly select from a fixed set of output rates according to the desired quality or signal content; very few are truly variable-rate. As compressed frames are generated, they are passed to the `RTP` packetization routine. Each frame has an associated timestamp, from which the RTP timestamp is derived [54].

3.3.2 Timestamps and the RTP Timing Model

As shown in Figure 3.4, the general format of an `RTP` packet consists of four parts: 1) The mandatory RTP header; 2) An optional header extension; 3) An optional payload header; and 4) The payload data. The `RTP` packet is generally transmitted using UDP/IP. The

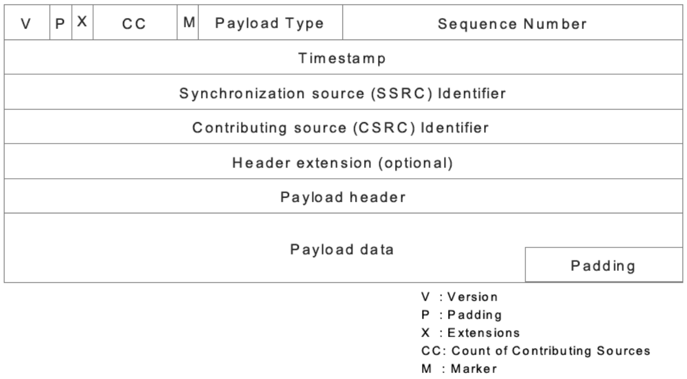

V
P
X
CC
M
Payload Type
Sequence Number
Timestamp
Synchronization source (SSRC) Identifier
Contributing source (CSRC) Identifier
Header extension (optional)
Payload header
Payload data
Padding
V : Version
P : Padding
X : Extensions
CC: Count of Contributing Sources
M : Marker

Figure 3.4: Format of an RTP Packet

mandatory RTP data packet header is typically 12 octets in length, although it may contain a contributing source list, which can expand the length by 4 to 60 additional octets. The fields in the mandatory header are the payload type, sequence number, timestamp, and synchronization source identifier.

The RTP *timestamp* represents the sampling instant of the first octet of data in the frame. It starts from a random initial value and increments at a media-dependent rate. For example, during the live audio conversation, the timestamp is incremented by the packetization interval times the sampling rate. If the audio packets contain 20 ms of audio payload where audio is sampled at 8000 Hz then the timestamp for each frame (or block) of audio increases by the increments of 160, even if the block is not sent due to silence suppression. The timestamps are assigned per frame, therefore if a frame is fragmented into multiple RTP packets, each of the packets will have the same timestamp.

3.3.3 Receiver Playout Algorithm

The sender of the voice stream is responsible for choosing an appropriate clock source with sufficient accuracy and stability. It should be noted that the RTP specification does not place any requirement as far as resolution, accuracy or stability of the media clock is concerned. The timestamps in RTP packets establish a relationship between the sampling process and a reference clock. The actual sampling rate will differ slightly from this nominal rate of 8000 Hz. A significant body of work has focused on the receiver buffer design and to accommodate the divergence of sampling rate. A receiver is expected to reconstruct the timing of the media from timestamps included in the RTP packets and should be able to handle the variability of the sender media clock. The RTP specification does not say anything about the buffering amount and leaves it to the vendors to design their own playout algorithm. Instead of designing a playout algorithm, our goal here is to measure the sampling rate divergence and establish a relationship to the calling device type.

3.3.4 Telephony Clock Sources

In PSTN world, central offices (COs) and digital loop carriers (DLCs) are made to work with precision stratum clock references. To achieve better VoIP quality of service, it is important to maintain VoIP voice interface clock accuracy matching with PSTN system clock. For example, even the voice codecs have their own clock precision requirements. Within PSTN and VoIP networks, the most commonly used waveform codec all over the world is G.711 [40]. Its formal name is Pulse code modulation (PCM) of voice frequencies. It defines μ-law and A-law as two main compression algorithms. The μ-law algorithm is used in North America and Japan, whereas the A-law algorithm is used in Europe and the rest of the world.

G.711, also known as Pulse Code Modulation (PCM) uses a sampling rate of 8000 samples per second, with the tolerance on that rate ±50 parts per million (PPM). VoIP adapter is used to connect an analog telephone to a VoIP service provider network through its ethernet interface. The other names used for VoIP adapter are terminal adapter, customer premise equipment (CPE), and gateway. VoIP adapter can support multiple tip/ring two wire telephone interfaces. It is more akin to traditional telephone connection over the PSTN where central office (CO) switch feeds battery and ringing to the phone. The phone itself completes the tip/ring circuit to place a call or receive a call.

In VoIP world, the FXS (Foreign Exchange Subscriber) circuit within adapter emulates CO switch providing both battery, detecting loop current, and ringing to the phone. The FXS circuit consists of two parts, a subscriber line interface circuit (SLIC) and a CODEC. A CODEC is comprised of analog-to-digital (ADC) and digital-to-analog (DAC) functions. When a speaker speaks, the analog voice signal from the phone is converted to a four wire interface in the SLIC functional block, and it is sampled in the hardware CODEC at 8 kHz. These samples are interfaced to the processor through a pulse code modulation (PCM) interface. The PCM interface is a serial interface maintained at multiples of 64 kHz clock frequency.

In most VoIP systems, the PCM clock is derived from the digital signal processor (DSP). In the VoIP adapter, the voice hardware's interfacing clock has to be maintained at $\pm 32 - 50$ PPM and the clock timing root-mean-square (RMS) jitter at less than 3 nanoseconds (ns) [48]. The PCM clock precision affects the sampling and thus propagated to the RTP packet intervals. For example, 160 samples payloads are treated as 20 ms frames. Any error in PCM clock frequency will cause the deviation in sampling and that will affect 20 ms IP packet duration. Compared to VoIP adapter where a subscriber needs two separate device (i.e., analog phone and adapter), IP phone does not require two separate devices i.e., Analog phone and adapter. However, CODEC operation is same as we discussed earlier.

The PC-based softclients use PC's processor to provide voice and network functions. The PC software clocks consist of a low grade quartz resonator stabilized oscillator and a hardware counter that interrupts the processor at regular intervals known as a *tick* to a system variable representing the software clock time. It can accumulate errors for various reasons (e.g., temperature and aging), but mainly due to incorrect oscillator frequency. The hardware and physical interfaces of PC are reused for VoIP calls. However, the shared nature of the PC environments, in which these softclients run, is very different from a closed environment found in VoIP adapters and IP phones. Besides the inaccuracy of media (software) clock, there are many other factors which may also affect the preciseness of sampling rate and the time duration between media packets. For example, there could be sudden spikes in usage of PC's CPU, network bandwidth impairments, and power management policy affecting CPU clock speed.

3.4 System Design

3.4.1 Local Measurement of Sampling Impreciseness

To measure the sampling rate deviation, both signaling and media RTP streams are monitored by the detection module. We first assume that it is possible to place the detection module close to the caller side to locally monitor these streams and will then discuss the placement of the device detection module in next subsection in more detail. From signaling, we can extract the caller identification along with the codec and user-agent information. From the RTP stream, we measure the sampling rate deviation by measuring the PCM or media clock's impreciseness and therefore clock's skew. We assume that for a particular telephone device $\mathbb{A}$, the timestamp in i^{th} RTP packet is T_i and let t_i be the time when this packet is recorded by the detection module. We define x_i as the time elapsed between the first and i^{th} packet observed by the detection module:

$$x_i = t_i - t_1$$

Similarly, the w_i is the time elapsed between the first and i^{th} packet at the sender side as derived from the packet timestamp values. We define w_i as

$$w_i = T_i - T_1$$

The sender device's clock skew cannot be measured directly, and so we have to rely on the mapping of the sender's timeline on the detection module's timeline. Here we define a term named *offset* as the difference between the sender clock's notion of time and that of the detection module's clock taken as a reference. The y_i is clock offset of the i^{th} packet:

$$y_i = w_i - x_i$$

In this way we get a set of clock offset data points $(x_i;\, y_i)$ corresponding to device $\mathbb{A}$. If we plot the clock offset datapoints, we will get an approximate linear pattern with non-zero slope. Figure 3.5 shows the plots of clock offset (y_i) vs observed time (x_i) for different devices including Cisco, Panasonic, Polycom, and Aastra hardphones, as well as Linux and Windows based soft clients.

It is observed that packet offsets sometimes drift dramatically and thus cause a jump point, and sometimes drift slightly and thus cause an outlier point due to random network delays in the communication path. The main reason for such jumps is that a device may have voice activation detection (VAD) enabled. In VoIP, VAD is a mechanism to detect the absence of audio and conserve bandwidth by preventing the transmission of "silent packets" over the network. For example, the offset data points for *Aastra* hardphones show that they have VAD enabled, because there are several points far from the mean of their set of points.

To estimate the clock skew of device $\mathbb{A}$ from its clock offset-set, we need to measure the

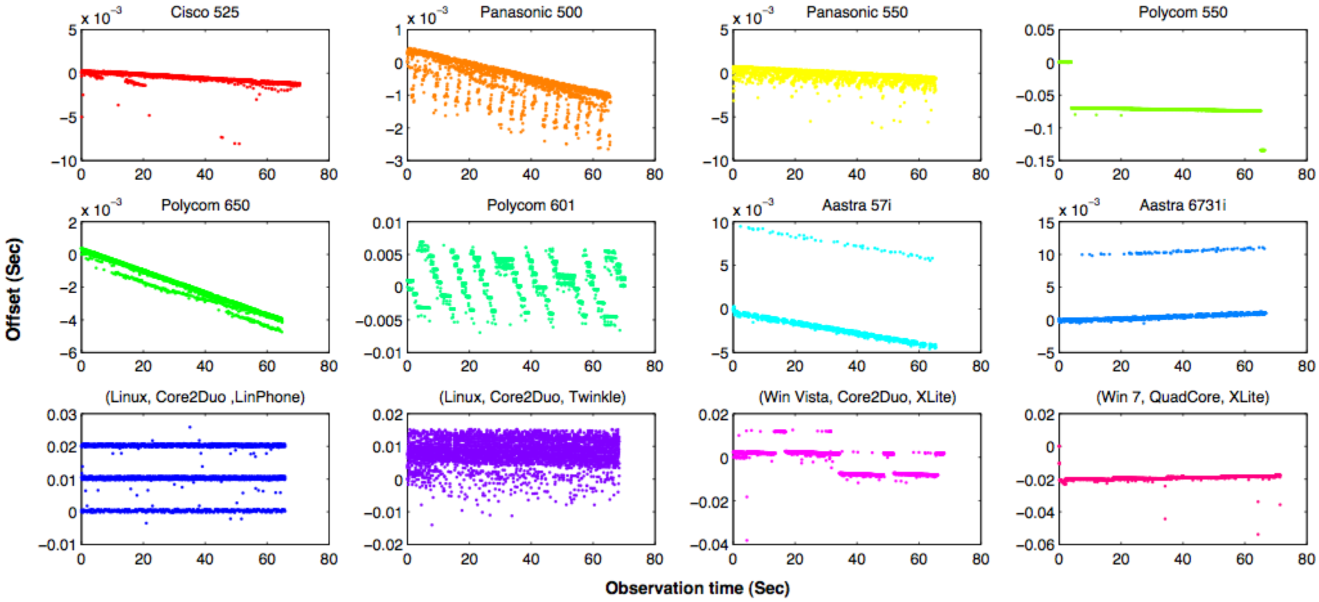


Figure 3.5: Clock Offset-sets for Different Hard Phones and Soft Clients

slope of this approximate linear pattern. For clock offset-sets with jump points, we first detect jumps and select a significant part without jump. Two different methods can be used for estimating the clock skew from the offset-set: Linear programming method (LPM) that is proposed in [47] and later used by Kohno et al. in [45] and Least Square Fitting (LSF) method that is based on finding a line that is at the least square distance from all the time offsets.

One of the major differences of LSF from LPM is its lack of tolerance towards outliers. Kohno et al. [45] showed that random delays in the network can affect the accuracy of the clock skew estimation by the LSF method. The LPM method instead minimizes the effect of any unpredictable delays as it has higher tolerance towards outliers. The clock skew estimation remains stable even if there are a significant number of outliers. Since our offset data-sets may have significant number of outliers, we use the LPM method for estimating clock skew.

3.4.1.1 Linear Programming Method (LPM)

Following the approach of Kohno et al. [45] to measure the slope of the clock offset data points, we use linear programming based algorithm. The approach involves fitting a line $\alpha x + c$ in such a way that it upper-bounds the set of clock offset data points while minimizing the sum of the distances of points from the line. The slope α of the line is a clock skew estimate. More formally, the clock skew estimation is an optimization problem with constraints

$$\alpha x_i + c \geq y_i, \quad \forall i = 1...n$$

and the following objective function that needs to be minimized

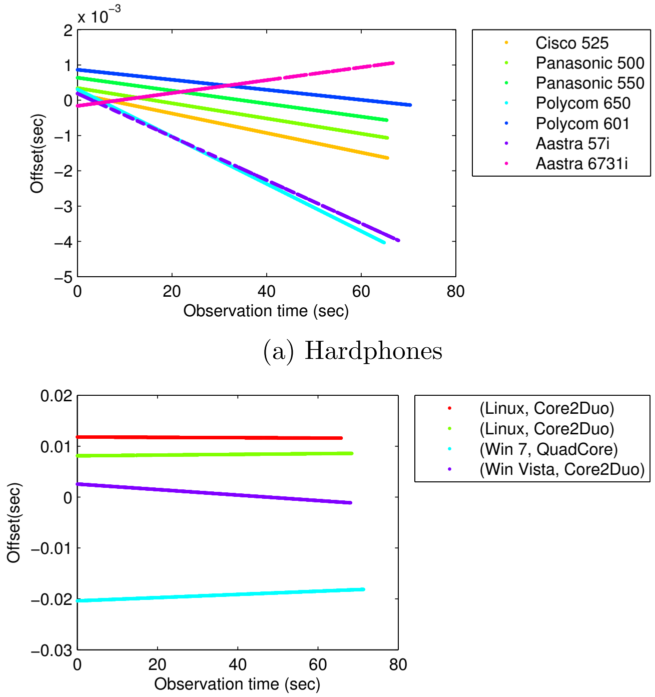
$$\frac{1}{n} . \sum_{i=1}^{n} (\alpha x_i + c - y_i)$$

This linear programming problem can be solved using standard linear programming methods, but there exist two variations also having linear time solution [45]. It is an effective approach when clock offset data points contain many outliers due to random network delays in the communication path.

Using LPM, we estimated clock skew for 15 devices, which consist of 9 hardphones and 6 soft clients participating in sixty call scenarios as either caller or callee. For the sake of resolution, only one LPM based fitted line to estimate clock skew as its slope is shown in Figure 3.6. Based on the resulted clock skews, as shown in Figure 3.6a and Figure 3.6b, we observed that:

- Different devices have different clock skews. Each individual device has its own almost unique approximate linear pattern and clock skew.

- The approximate linear pattern of clock offset for an individual device stays the same in different scenarios, independent of being caller or callee and independent from the



(a) Hardphones

(b) Soft clients

Figure 3.6: LPM Based Clock Skew Estimation for (a) Hard Phones and (b) Soft Clients

device at the other side.

- Individual devices from the same maker and model have almost the same approximate linear pattern of clock offset as well as clock skew.

3.4.2 Placement of Device Detection Module

As discussed earlier, calling devices can be identified based on their clock skews provided that the device detection module can be placed close to the calling party to locally monitor both signaling and media RTP streams. Since placing such a standalone module to locally collect data and send to a server for analysis is not cost effective and may be prone to attack, embedding the device detection module in one of these already existed elements of

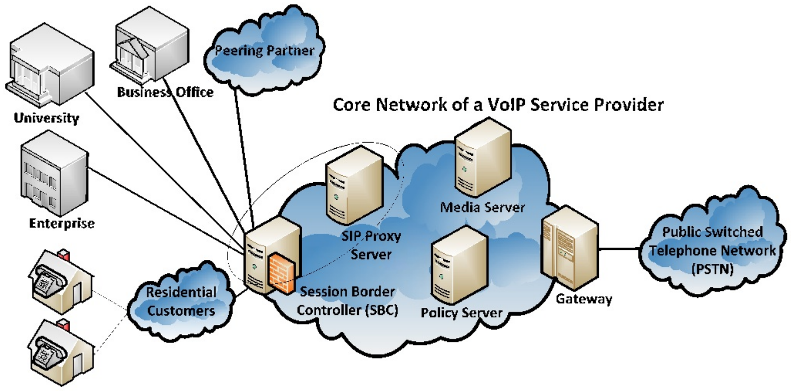


Figure 3.7: Placement of Device Detection Module

VoIP network is of interest. As shown in Figure 3.7, the Session Border Controller (SBC) refers to the point of demarcation between one part of a network and another. For example, it acts as a demarcation point between a customer access side and a service provider's core network or between two peering partners' networks. SBC is a device used in a VoIP service provider's network to exert control over the signaling (and usually over the media streams also) involved in setting up, conducting, and tearing down calls.

The Communications Assistance for Law Enforcement Act (CALEA) is a United States wiretapping law that also extends to VoIP telephony. As per CALEA requirement, the telecommunication companies make it possible for law enforcement agencies to tap any phone conversations carried out over their networks, as well as making call detail records available. Therefore, SBC acts as a mediator and the media streams between any two end-points have to come to the SBC first and then are relayed to the other end. In a practical deployment, the device detection module can be implemented as a loadable module of the SBC device because it complements the functions of the SBC device. However, it could also exist as an independent device and have the ability to observe both to-and-from SIP signaling messages used in setting up and tearing down VoIP calls and the media streams between them.

3.4.3 Remote Measurement of Sampling Impreciseness

To remotely estimate sampling impreciseness, the device detection module is placed in SBC. We monitored both signaling and RTP media streams in SBC for 15 devices, which consist of 9 hardphones and 6 soft clients participating in sixty call scenarios. The extracted offset points for these devices from SBC along with their corresponding offset points locally collected are shown in Figure 3.8. There are several similar offset points for an individual device from different calls, but we just show one set for each device in Figure 3.8. The two main observations are described as follows:

1. The offset points extracted based on monitoring streams in SBC are spreaded out in a way that they cannot be considered as approximate linear pattern. Hence it is not reasonable to use LPM to fit line to them for estimating clock skew.

2. Comparing SBC offset points to local ones, it seems that elements in a network path and SBC itself can perturb local offset points and force them to spread out in diagonal line segments because of their buffer and internal dispatch methods. We measured the slope of all diagonal line segments for all traces and realized that the slope is the same. Hence all diagonal segments seem to be parts of a parallel line that is broken down in almost equal time periods. We plan to investigate the relation between length and slope of these diagonal segments, as well as the type and number of SBCs in the path a call traversed, as our future work.

Based on these observations, we can see that the clock skew as the slope of a fitted line to SBC offset points cannot identify devices remotely. However, the statistical measures that represent the shape of offset points, such as variance, skewness, and kurtosis, can be examined as good metrics for remote device identification.

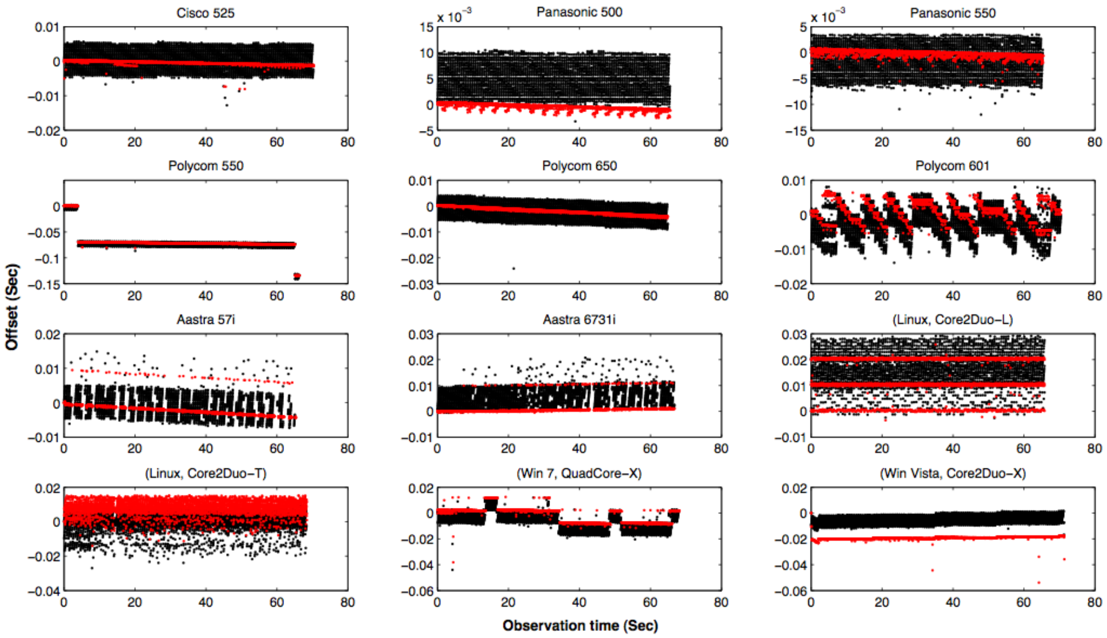


Figure 3.8: SBC Offset Points (Black Points) in Comparison to Local Ones (Red Points)

3.4.4 Fuzzy Based Device Identification System

Our extensive experiments using different metrics on SBC offset points including slope of upper-bound fitted line, mean, variance, skewness and kurtosis, show that *variance* can be the main metric to remotely identify most devices and can be combined with other metrics to gain better accuracy.

The variance is the "second central moment" that is widely used and measures the "width" of a set of points in one dimension, and is calculated using below formula (μ: mean of offset points and n: number of collected offset points):

$$Var(X) = \frac{1}{n}.\sum_{i=1}^{n}(x_i - \mu)^2$$

The skewness, as the "third central moment", characterizes the degree of asymmetry of a set of points around its mean and describes how it is skewed from its mean. It can be calculated using the formula below (μ: mean of offset points, σ: standard deviation of offset points, and n: number of collected offset points):

$$Skew(X) = \frac{1}{n}.\sum_{i=1}^{n}(\frac{x_i - \mu}{\sigma})^3$$

The kurtosis, as the “fourth central moment”, measures the relative peakedness or flatness of a set of points compared to the normal distribution of the same variance. It can be calculated using the formula below (μ: mean of offset points, σ: standard deviation of offset points, and n: number of collected offset points):

$$Kurt(X) = \frac{1}{n}.\sum_{i=1}^{n}(\frac{x_i - \mu}{\sigma})^4$$

The device identification method should be fast, simple, and scalable to be practically embedded in SBCs for identifying devices in real time. Thus, a Fuzzy Rule Based System (FRBS) is designed as the device identification system. As shown in Figure 3.9, our Fuzzy Inference System (FIS) maps a given input, which is a combination of mentioned metrics, to an output, which is the device type, using the fuzzy logic based on Mamdani method [46]. The FIS involves membership functions, fuzzy logic operators, and if-then rules [83]. It has the following four modules:

3.4.4.1 Fuzzification module

The system inputs, which are crisp numbers of extracted features such as variance and slope of the upper-bound fitted line, are transformed into fuzzy sets.

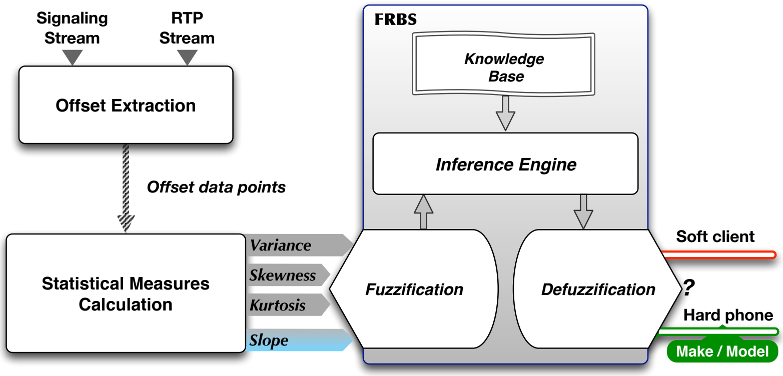


Figure 3.9: Fuzzy Rule Based Device Identification System Design

3.4.4.2 Knowledge base module

Stores the conditional statements, which are provided by experts in form of IF-THEN rules and make fuzzy logic useful. Since most of the devices can be identified based on the variance of SBC offset points, if-conditions for most of the rules have one part. However, there are cases that cannot be identified precisely by only using variance; other metrics such as slope of the upper-bound line should be coupled with variance to identify them. For example, "Aastra" devices can be identified from other devices by just using variance, but Aastra model itself cannot be identified merely based on variance, here the slope of the fitted line is needed. SBC offset points for both Aastra 57i and 673i models have almost the same variance, but the slope for 57i is negative and that for 673i is positive (as shown in Figure 3.5 and Figure 3.6a). To identify such cases, some if-conditions have more than one part combined using fuzzy operators such as *AND*.

3.4.4.3 Inference engine module

Simulates the human reasoning process using fuzzy inference on the inputs and IF-THEN rules.

3.4.4.4 Defuzzification module

The fuzzy set obtained by the inference engine is transformed into a crisp value that corresponds to a device class.

Since our fuzzy system is simple and fast, it can identify devices in real time. As it is also easy to modify an FIS just by including or excluding rules, the system is scalable and can be extended easily to identify devices that have not been yet examined.

3.4.5 Security Applications of the Proposed System

Our fuzzy identification system can be placed in SBC of VoIP service provider and play two complementary roles for outgoing and incoming calls. For outgoing calls, the system identifies the caller device and locally queries its subscriber database to check if the caller is using a calling device other than configured against his user account. Hence the system helps service providers to block any outgoing calls that is originated from an unauthorized calling device.

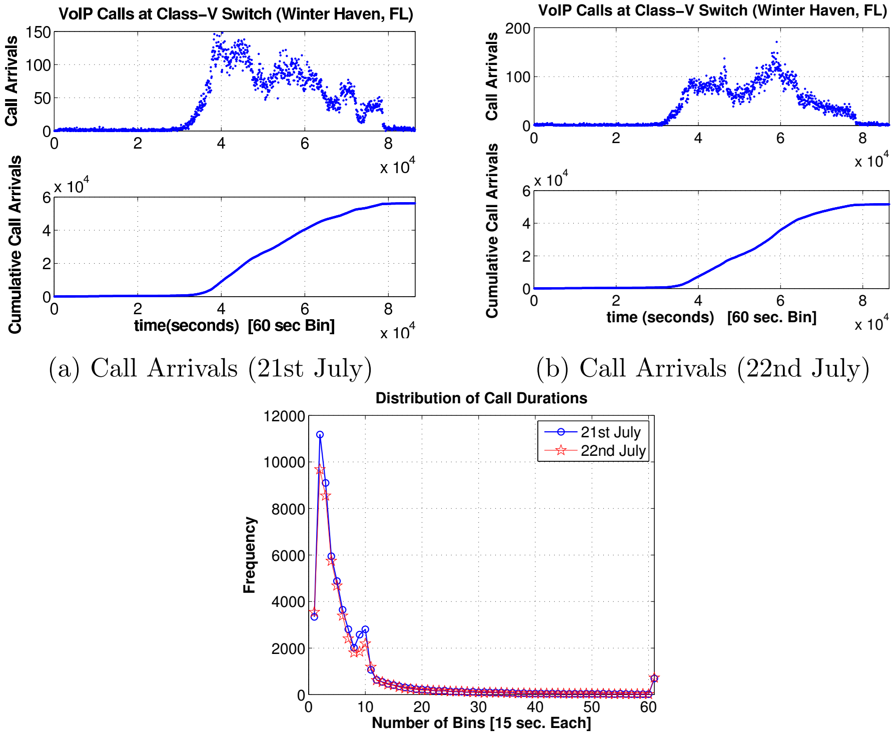
For incoming calls, the system identifies the caller device and sends a *verification* request to the original service provider where the caller claims to be originated from (i.e. the original SIP proxy server claimed by the caller-id). The original service provider receiving *verification* request is supposed to check if the identified device is the device bound with the caller account. In case that the caller uses authorized device, original service provider replies to callee side verifying the calling device. Otherwise, original service provider sends a warning of unauthorized device to callee side. As outgoing calls from unauthorized devices should have been blocked in their subscribers' service provider network, the callee side infers that the call has been originated from an unscrupulous service provider and the caller-id has been faked/spoofed. Hence the system helps service providers to block any unsolicited incoming calls that is originated from an unscrupulous VoIP service provider.

Our proposed system remotely identifies type of calling devices and helps to block unsolicited calls and prevent several well-known security threats resulting from unsolicited communication as follows:

- **Voice phishing (Vishing)**: Attackers spoof the caller ID and pretend themselves as trustworthy persons or legitimate organizations such as bank. In vishing attacks, victims may be asked for their private information such as SSN or passwords of bank accounts. Our system can identify type of calling devices and examine the truthfulness of caller-id to prevent voice phishing as well as the leakage of private information.

- **Voice spam**: Attackers fake/spoof caller-ids pretending to be someone else to place unsolicited calls for advertising, scanning of active VoIP phone numbers and requesting personal information of users. As our system can identify type of calling device and detect if the caller-id is spoofed, it can be considered as a solution for protecting voice spam in VoIP networks.

- **Flooding attack**: Attackers fake caller-ids and try to place huge number of unsolicited calls from pc-based softphones to victims. Our system can identify type of calling devices and reveal if calls are originated from unauthorized softclients. Hence such unsolicited calls can be prevented and their corresponding caller-ids can be stored locally in database of service providers to prevent incoming calls initiating flooding attacks.



(a) Call Arrivals (21st July)

(b) Call Arrivals (22nd July)

(c) Distribution of Call Durations

Figure 3.10: Call Arrivals and Distribution of Call Durations

3.5 System Evaluation

3.5.1 Baseline of Normal VoIP Call Behavior

We first establish the baseline of normal VoIP call behavior to make realistic calls in our experiments. Specifically, we use the call logs collected from a VoIP network of NuVox Communications, a voice service provider in Southeast and Midwest regions of the USA [51]. The seven days call logs were collected from a Class-V switch located at Winter Haven, Florida. The call logs correspond to VoIP calls made by subscribers of Orlando and Tampa cities in Florida. Figure 3.10 shows the call arrivals and the distribution of call duration characteristics of two different days. Each call log is of 24 hours duration starting at the midnight. The logs of 21^{st} and 22^{nd} July contain 56259 and 51625 successfully completed calls, respectively.

Call duration probability distribution. The call logs for VoIP traffic traces are analyzed to obtain call duration distribution. As shown in Figure 3.10 (c), we observe that about 50% of the calls complete within a minute. The measured call durations are used to calculate the mean μ and standard deviation σ. The mean and standard deviation pair (μ,σ) [in seconds] for the 21^{st}-22^{nd} July VoIP traces are found to be $(111.87, 264.04)$, and $(115.83, 283.58)$, respectively.

3.5.2 Experimental Setup

We set several experimental testbeds to evaluate the efficacy of our proposed identification system. Two primary testbeds, which are used as default, are explained here in detail and the other testbeds will be discussed wherever used. One primary experimental testbed is set using the open source of SBC and SIP server, and the other one is set using the IP telephony service of a popular VoIP service provider.

In the first experimental testbed, three computers are used to set up a VoIP *core* providing IP telephony services. The VoIP core is hosted within a university campus network. One computer (2 GHz Intel Pentium IV with 512 Mbytes of RAM) is used to run the session border controller (SBC) loaded with an open source OpenSBC [10] library. It is installed with two 100Mbps network interface cards, one with public IP address facing the Internet and the second with private IP address to communicate within the local private network. The SBC is deployed in the *back-to-back user agent* (B2BUA) mode. The second computer (633 MHz Pentium III PC with 128 Mbytes RAM) runs as a PC-based Linux router connecting the private-side of SBC to the SIP proxy server using 100Mbps Ethernet links. The third computer (866 MHz Pentium III PC with 256 Mbytes RAM) is used

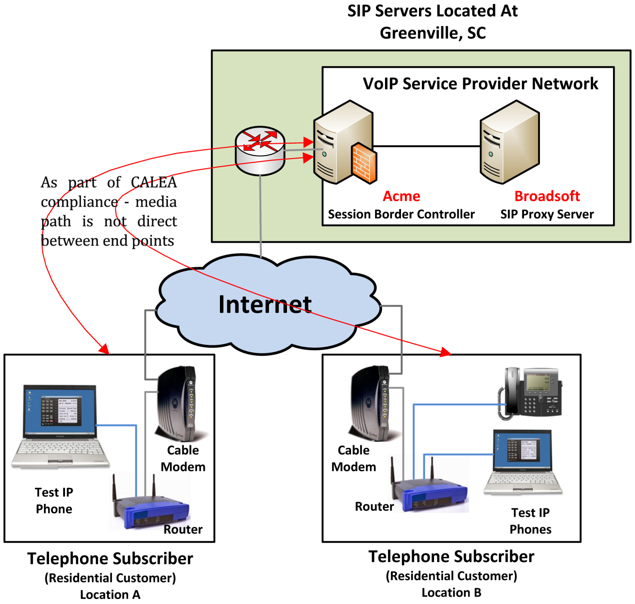


Figure 3.11: Experimental Testbed

as our SIP proxy server based on SER (i.e., SIP Express Router [58]), an open source software project that provides call control, routing and other operation service support and features. This system is scalable from a single box in a lab to carrier grade networks supporting thousands of telephone subscribers.

In the second experimental testbed shown in Figure 3.11, the IP telephony service is provided by a popular VoIP service provider. The service provider uses Broadsoft [4] as its SIP proxy server and Acme [6] as its SBC, which are located in Greenville, SC.

We place various calls using both softclients and hardphones connected to the Internet through cable modems representing residential VoIP telephone subscribers. The calls are made from many cities within the U.S. following normal VoIP call behavior discussed earlier in this section to mimic realistic local and long distance calls. As discussed in Section 3.4.1, we have already monitored signaling and RTP streams locally and shown that clock skew as the slope of a fitted line using LPM on local offset points is proved to be a good metric for device identification.

We focus on SBC offset points later on this section to examine the efficiency of our proposed system for remote device identification. Hence the signaling and voice streams of calling parties are also captured at the SBC box using tcpdump [41] packet capture tool. We then use sharktools [15], which provides Wireshark's packet inspection capabilities in Matlab [1], to extract the required information (RTP time stamp and time of recording a packet) from streams to measure offset points.

The statistical measures including variance, skewness, and kurtosis along with the slope of an upper-bound fitted line to offset points are then calculated. At the next step, our fuzzy based device identification system, which is implemented in Matlab, inputs these measures and identifies a device as its output. The following presents different scenarios and experiments that we conducted to evaluate the efficiency of the system.

3.5.3 Individual Device Identification

To check if the proposed fuzzy system can identify an individual device independent from 1) being caller or callee and 2) the type of another call party, sixteen calls were made between an individual *Cisco 525* hardphone and another devices including hardphones and soft clients. We then monitored both signaling and RTP streams for the *Cisco 525* hardphone and extracted their corresponding offset points as shown in Figure 3.12. Their variance as the main metric for device identification along with other statistical measures, skewness and kurtosis, were calculated and inputted to the system. These quantitative measures of the shape of offset points for the *Cisco 525* hardphone in the sixteen different call scenarios are summarized in Table 3.1.

The results show that the variance values for these offset points are close and fall in the specific interval of $[8.16 * 10^{-6}, 8.47 * 10^{-6}]$ with the mean and standard deviation of

$8.31 * 10^{-6}$ and 10^{-7}, respectively.

Although the skewness equal to zero can be interpreted as the data in a perfectly symmetrical distribution, a skewness of exactly zero is quite unlikely for real-world data. Thus, we interpret the skewness number based on Bulmer's rule of thumb [20] that can be summarized as below:

- If skewness is less than -1 or greater than +1, the set of points are *highly skewed.*
- If skewness is between -1 and -0.5 or between +0.5 and +1, the set of points is *moderately skewed.*
- If skewness is between -0.5 and +0.5 , the set of points is *approximately symmetric.*

Since the skewness numbers for all offset points of Cisco 525 are between -0.5 and +0.5, they are approximately symmetric.

As shown in Table 3.1, the kurtosis values for all offset points of Cisco 525 are close to each other, and they are all < 3. Based on the rules below to interpret them, their offset points are platykurtic and have similar shapes.

- As a normal distribution has kurtosis exactly 3, any set of points with kurtosis ≈ 3 is called *mesokurtic.*
- A set of points with kurtosis < 3 is called *platykurtic.* Compared to a normal distribution, its central peak is lower and broader, and its tails are shorter and thinner.
- A set of points with kurtosis > 3 is called *leptokurtic.* Compared to a normal distribution, its central peak is higher and sharper, and its tails are longer and fatter.

Since the statistical measures for offset points of the phone in all different call scenarios are almost the same (i.e., describing similar shapes), our system perfectly identifies all of them as *Cisco 525.*

Table 3.1: Statistical Measures of Offset-sets for Cisco 525 Hardphone in Sixteen Different Call Scenarios

Trace No.	Variance	Skewness	Kurtosis
1	0.00000839	-0.0290	1.9293
2	0.00000831	-0.0345	1.9706
3	0.00000836	-0.0786	2.1489
4	0.00000820	-0.0080	1.8409
5	0.00000844	-0.0058	1.8789
6	0.00000826	-0.0094	1.8631
7	0.00000830	-0.0528	2.0007
8	0.00000840	-0.0861	2.0925
9	0.00000818	-0.0101	1.8660
10	0.00000847	-0.0179	1.8980
11	0.00000828	-0.0601	2.0462
12	0.00000821	-0.0115	1.8559
13	0.00000816	-0.0074	1.8339
14	0.00000845	-0.0506	1.9980
15	0.00000824	-0.0093	1.8587
16	0.00000837	-0.0631	2.0191

3.5.4 Effect of Different Data Networks on Offset Points

We conduct a set of experiments to check if network elements in the path that a call traversed affect offset points. Further, we validate if the proposed system can identify a device independent from the traversed path. In addition to the testbed in Figure 3.11, we use another similar testbed with the same VoIP core network but in a different location, Charlotte, NC. The caller and callee are in the same location, Aldie, VA, and also the number of VoIP devices and their types are the same.

Figure 3.13 shows two set of offset points for the same hardphone Polycom 650 as the caller in two different calls to the same callee: one is served by VoIP core in Greenville, SC (blue points) and another one is served by VoIP core in Charlotte, NC (red points). The two offset sets seem to have a similar shape formed in parallel line segments with the same slope. However, as shown in the magnified part of Figure 3.13, their distributions on

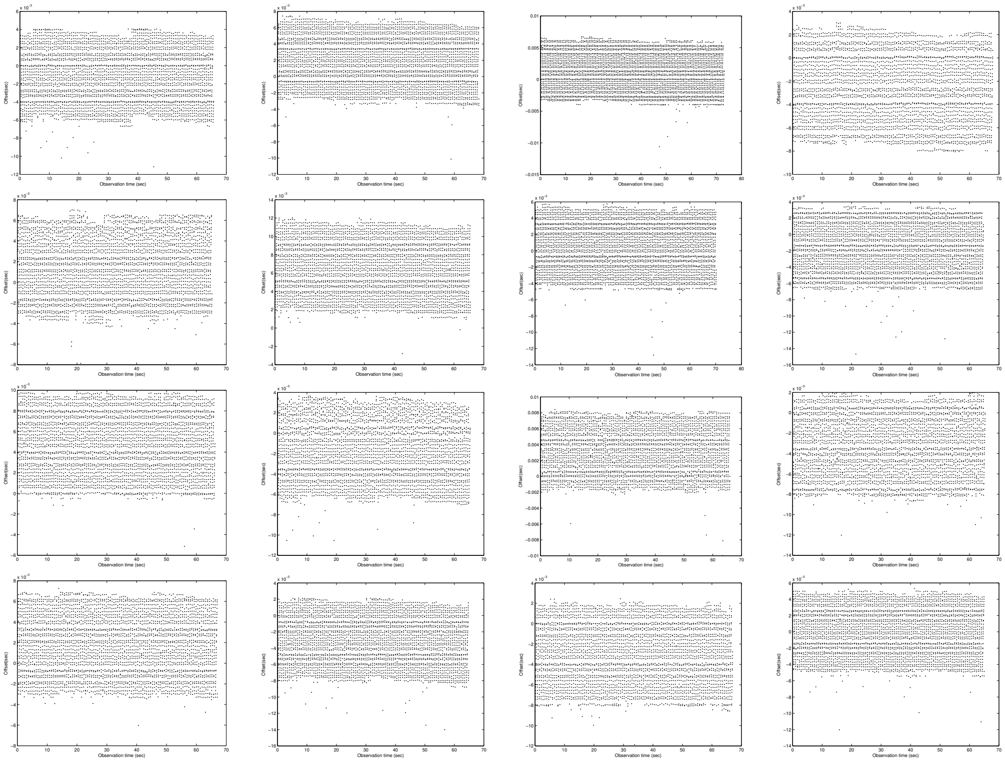


Figure 3.12: Clock Offset-sets vs Observation Time for Cisco 525 Hardphone in Sixteen Different Call Scenarios

the line segments are different. The offset points captured by SBC in Greenville, SC are almost uniform but the ones captured by SBC in Charlotte, NC have a greater mean and are affected by the queueing characteristics of more network elements in the longer path.

Although the two set of offset points are differently distributed in parallel line segments, their slopes of the upper-bound fitted lines and statistical measures including variance, skewness and kurtosis are almost the same. The variance, skewness and kurtosis for the offset points of Polycom 650 captured in Greenville, SC and in Charlotte, NC are (0.0000092, -0.11, 2.79) and (0.000010, 0.012, 2.27), respectively. Thus, our proposed system can accurately identify both devices as the same hardphone Polycom 650.

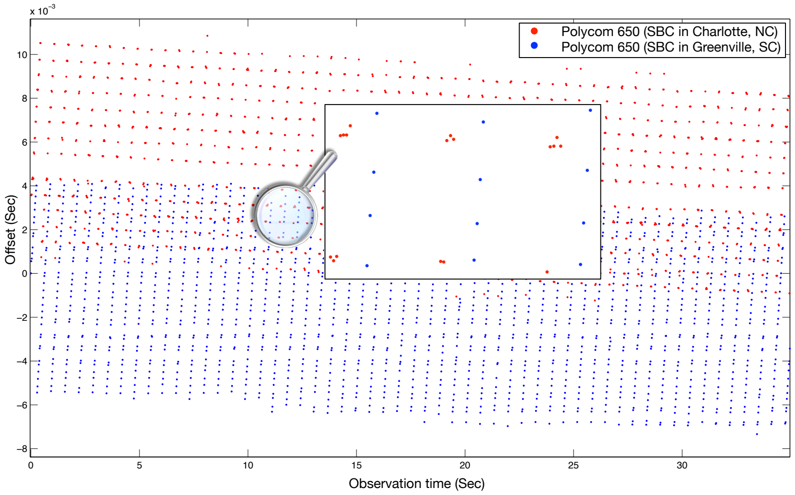


Figure 3.13: Offset Points For Polycom 650 Captured in Production SBC (Red Points) and Lab SBC (Blue Points)

3.5.5 Effect of Different Origination Call Points on Offset Points

We conduct a set of experiments to check if the origination call points affect the offset points. Further, we validate if the proposed system can identify devices independent from their origination points. As shown in Figure 3.14, we originate different calls between the same devices from two different networks: one from local network in Aldie, VA and the other from a university campus network, VA. Figure 3.15 shows the two set of offset points for the same hardphone Polycom 650 as the caller in two different calls to the same callee: one from home local network (red points) and the other from the university campus network (blue points). The two offset sets seem to have a similar shape formed in parallel line segments with the same slope and their distributions on the line segments are almost uniform.

For both sets of the offset points, the slopes of the upper-bound fitted lines and statistical measures including variance, skewness and kurtosis are almost the same. The variance, skewness and kurtosis for the offset points of Polycom 650 from home network and from

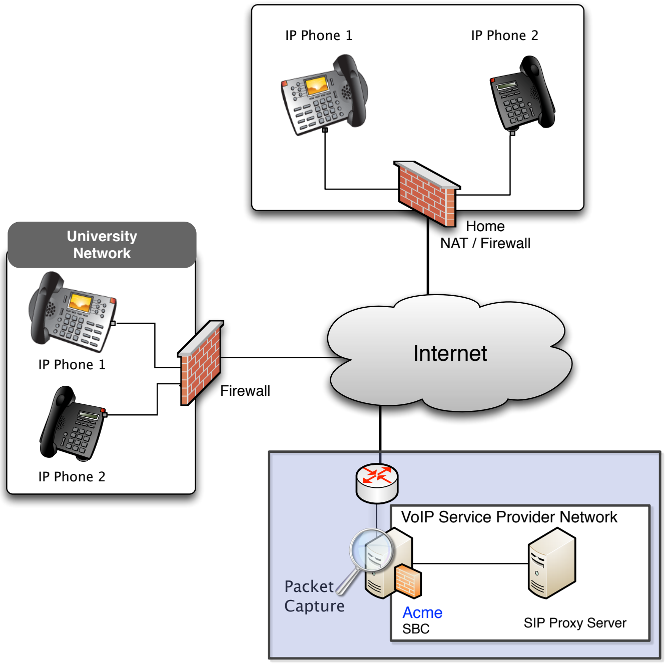


Figure 3.14: Experimental Testbed to Originate Calls from Different Points

university network are (0.0000098, 0.036, 2.29) and (0.0000092, -0.069, 2.28), respectively. Thus, our proposed system can also accurately identify both devices as the same hardphone Polycom 650.

3.5.6 Effect of Different SBC Capturing Points on Offset Points

We conduct a set of experiments to study effect of the number and type of SBCs that a call passes through to reach the destination on offset points. Further, we validate if the proposed system can identify devices independent from the number and type of SBCs in a call path. As shown in Figure 3.16, we set another testbed including two different types of SBCs, Genband S3 [5] and Acme, to monitor and capture both signaling and RTP media streams of calling parties in different SBCs. Hence we collect four different offset points:

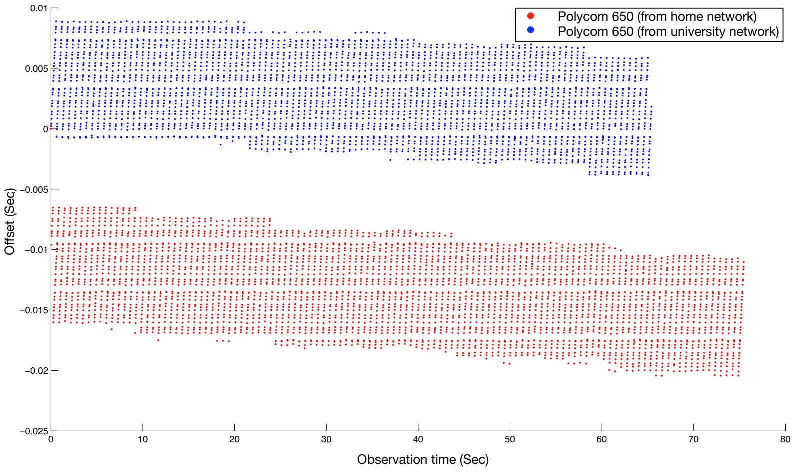


Figure 3.15: Offset Points For Polycom 650 from Home Network (Red Points) and University Network (Blue Points)

1. The offset points of the traces coming to Genband S3 SBC to be forwarded to Acme SBC to reach their destinations.

2. The offset points of the traces coming to Acme SBC to be forwarded to Genband S3 SBC to reach their destinations.

3. The offset points of the traces coming to Genband S3 SBC to be forwarded to their destinations and already passed through Acme SBC.

4. Similar to case 3, the offset points of the traces coming to Acme SBC to be forwarded to their destinations and already passed through Genband S3 SBC.

Figure 3.17 shows the three sets of the offset points for the same hardphone Polycom 650 as the caller in different calls to the same callee: one is captured in Acme SBC (red points), another one is captured in Genband S3 SBC (green points), and the last one captured in Genband S3 SBC after passing through Acme SBC (blue points). The first two offset sets seem to have a similar shape formed in parallel line segments with the same

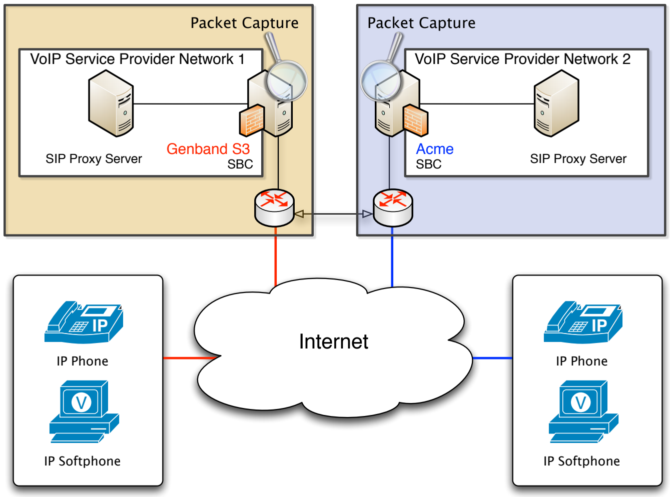


Figure 3.16: Experimental Testbed to Examine Effect of Using Different SBCs

Table 3.2: Statistical Measures of Offset-sets for Polycom 650 captured in different SBCs

Number/ Type of SBCs	Variance	Skewness	Kurtosis
1/ Acme	0.00000927	-0.1105	2.7916
1/ Genband S3	0.00000908	0.0011	2.0405
2/ Acme and Genband S3	0.00000965	0.0373	2.2048

slope and their distributions on the line segments are almost uniform. However, the last offset set has a similar shape but its distribution on the line segments is different from those of the other two offset sets using just one SBC, because its distribution is affected by both SBCs in the path with more VoIP elements.

In spite of their different distributions on parallel line segments, all three sets have almost the same slope of an upper-bound fitted line and statistical measures, including variance, skewness and kurtosis. These quantitative measures of the shape of the offset points for *Polycom 650* hardphone with different number and type of SBCs in VoIP networks are summarized in Table 3.2. Thus, our proposed system can also accurately identify all the three devices as the same hardphone Polycom 650.

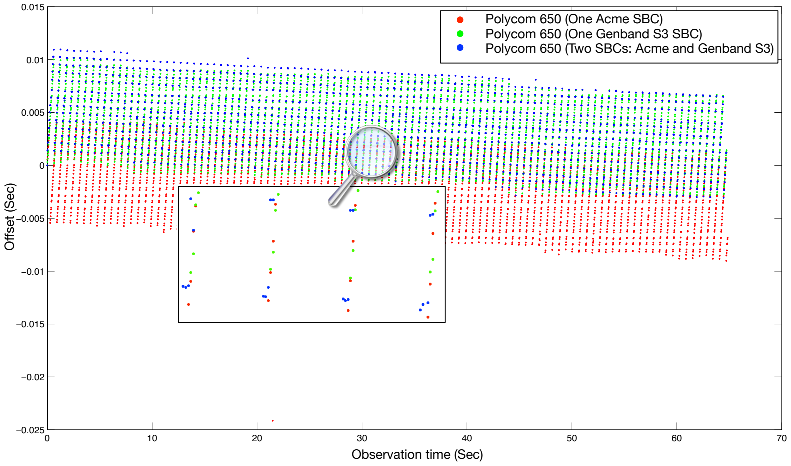


Figure 3.17: Offset Points For Polycom 650 Using One Acme SBC (Red Points), One Genband S3 SBC (Green Points) and a Combination of Both SBCs (Blue Points)

3.5.7 Detection Accuracy Rate

Different calling devices, Hardphones and Soft clients, located at many cities within the USA are connected to the Internet as residential VoIP telephone subscribers. Various hardphones with different make/model including Cisco 525, Panasonic 500, Panasonic 550, Polycom 331, Polycom 550, Polycom 601, Polycom 650, Aastra 57i, Aastra 6731i participate in the experiments. Different PCs with either Core2Duo or QuadCore cpu using Linux, Windows Vista and Windows 7 as OS running four widely used softphones, LinPhone [7], Twinkle [9], XLite [3] and YATE [80], also participate as calling parties. These devices, either as callee or caller, made different sixty calls that follow the baseline of normal VoIP call behavior, as discussed in section 3.5.1, to complete within a minute. Since our identification method is based on exploiting the impreciseness of the codec sampling rate, all devices use the most commonly used waveform codec all over the world, G.711 [40], to make these calls.

We then used our proposed system to identify devices. Figure 3.18 shows the boxplot

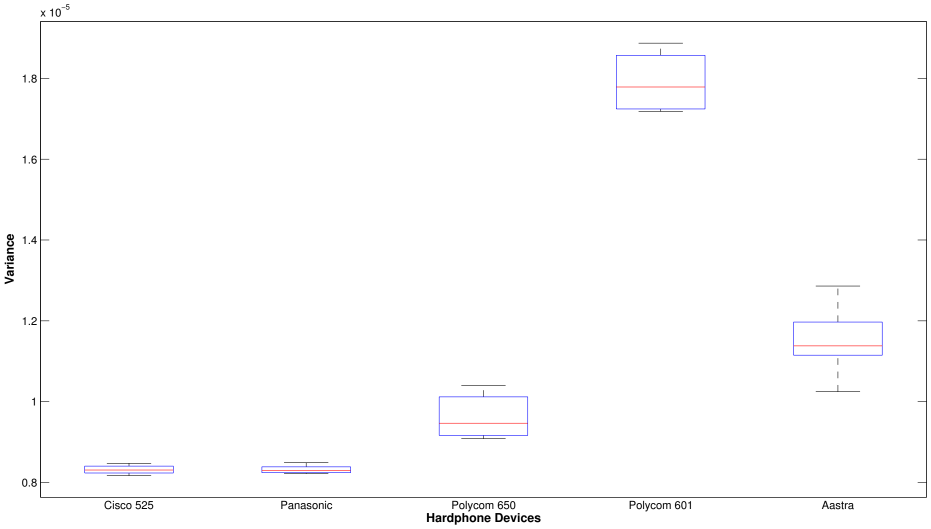


Figure 3.18: Boxplot for Calculated Variances for Offset Points of Different Hard Phones

of calculated variances for all SBC offset points that belong to each hardphone device. Based on the comparison between information about devices extracted from their signaling streams and the output of our identification system as device type, following are concluded about detection accuracy rate of our proposed system:

First, since the variance of SBC offset points for soft clients are significantly greater than those of the hardphones, the proposed fuzzy based identification system can accurately distinguish between hardphones and soft clients. The system achieves the detection accuracy rate of 100% for identifying whether the device is a hardphone or a soft client. Second, although the proposed system cannot identify the details of a soft client device, such as CPU, OS, and VoIP application type, it can identify the maker and model of a hardphone in almost all cases with the overall detection accuracy rate around 95%. As shown in Figure 3.18, the false identification of the system is mainly due to misidentifying *Panasonic 500* devices as *Cisco 525*, which have similar distribution patterns.

3.6 Related Work

The determination of true identity of a calling device helps in preventing many of the VoIP attacks such as caller ID spoofing, spamming, and call flooding attacks. Till now, most of the industry and academia efforts to tackle VoIP related attacks have been focused on: (1) determining the identity and trust value of a caller; (2) developing a stronger authentication mechanism for callers; and (3) analyzing the signaling messages to ascertain the true nature of call originating sources. For instance, Cai [21] extracted caller ID information from meta-data of a call to propose several mechanisms for caller ID verification. However, the proposed mechanisms fail if meta-data is faked by fake ID providers.

The Internet Engineering Task Force (IETF)'s RFC [61] analyzes the bulk spam calling problem in SIP environments and examines various potential solutions available for solving the email spam problem. Unfortunately, many of the anti-spam solutions that have been proposed or deployed are either heavily influenced by or directly inherited from the email spam world. For example, the anti-spam solutions based on computational puzzles [61] attempt to frustrate a VoIP bulk call generator by requiring it to solve some computational puzzles. While such methods require modification of the underlying signaling protocols, they are ineffective against distributed VoIP spam call generators, where multiple powerful PCs are compromised into zombies and used for generating bulk spam calls.

The Turing test [61, 78] based approaches require manual and active involvement of callers, which is not intuitive and may scare away many potential users. The solutions relying on social networks and callers' reputation values require infrastructure modifications of SIP UAs, yet they are susceptible to malicious reputation poisoning. The anti-spam solutions based on a trusted third party are not scalable. Although Iranmanesh et. al. [39] recently proposed an offline content-based approach to protect telephone subscribers' voice mailboxes from voice spam, it is hard to apply the content based filtering to defend against voice spam in real time.

Recently, Balasubramaniyan et. al. [17] proposed a PinDr0p method to protect caller ID by using call provenance. This method determines the traversal of a call through different service providers (i.e., VoIP, cellular, and PSTN). It is based on call audio features (such as degradation and noise characteristic) affected by the networks information which it has traversed. Being network dependent and not specific to end points, the proposed approach is coarse grained. For example, it cannot distinguish one call (or caller) from the other if both are originating from the same location and are using the same service providers. Piotrowski et al. [56] discussed voice spoofing attack as a more dangerous but less popular attack than caller ID spoofing attack, and proposed a watermarking mechanism to mitigate the threat. However, their approach requires to embed a build-in authorization function, which is based on watermarking technology, into the caller and callee's devices.

3.7 Conclusion

Most of the VoIP attacks are launched from PC-based malicious programs pretending to be IP hard phones. Our insight is that the determination of true identity of a calling device helps in preventing many of the VoIP attacks, such as caller ID spoofing, spamming, and call flooding attacks. Although calling devices present this identity information within their SIP INVITE messages, it can be easily faked by malicious ID providers. In this chapter, we propose a novel approach to verify the truthfulness of the identity information and distinguish a calling party, even if this information is deliberately manipulated.

Our proposed method is based on analyzing the caller's voice stream (i.e., RTP packets). Irrespective of whether it is PC-based softclient or hardphone, both perform audio sampling according to the negotiated audio codec between peers. However, the offset between time of receiving RTP packets in identification module and timestamp inside RTP packets shows that the sampling rate is not precise. We exploit this impreciseness as *clock skew* of a calling device type. The clock skew can be locally measured at the cost of modifying calling devices. We further implement the device detection module as a loadable module of the SBC device.

Through a series of experiments, we study the impact of data network elements (routers and switches) and VoIP network elements (different SBCs) in the call path upon offset points. In spite of these impacts, the statistical measures of offset points remain almost the same and can be used for *remote* device identification. We develop a simple and fast fuzzy based system to take these statistical measures as input and identify the type of a calling party. Our extensive experiments show that we can accurately distinguish between hardphones and softclients as well as identify make/model of hardphones with detection accuracy rate of 95%.

Chapter 4

Caller-ID as Digital Signature: A Method to Automatically Authenticate Calling Parties

Caller-ID has long been used to let the callee parties know who is calling. Caller-ID can also be used to verify the identity of caller for authentication and his physical location for emergency services. Since caller IDs in the Public Switched Telephone Network (PSTN) and circuit switched (CS) cellular network, such as Global System for Mobile Communications (GSM), are automatically generated by carriers, caller ID spoofing in such networks is almost impossible. In contrast, VoIP and other packet switched networks such as all-IP Long Term Evolution (LTE) network provide flexibility that helps subscribers to use arbitrary caller-id.

The delivery of multimedia such as voice and video over IP needs two separate protocols: one for signaling to initiate the multimedia session and another one for multimedia itself. Session Initiation Protocol (SIP) is the widely used signaling protocol for multimedia over

IP. Voice over IP (VoIP) and Voice over LTE (VoLTE) as an emerging technology utilize SIP as their signaling protocol for media session initiation that has replaced many traditional telephony deployments.

Since SIP sends `INVITE` request in plaintext with no authentication method to initiate a call, Caller-ID included inside requests can be spoofed by malicious users to initiate different attacks. Hence caller IDs are no longer dependable and cannot be used for verifying the identity and physical location of callers. Moreover interconnecting between SIP based telephony and other CS legacy telephone networks has also reduced the security of Caller ID systems for whole telephony network.

4.1 Overview of Proposed Approach

The goal of this work is twofold; proposing an implicit SIP authentication method to authenticate a caller while he places a call in VoIP and VoLTE, and proposing a new caller-ID based public key infrastructure for securing caller-ID to restore caller-ID trust for whole telephony network. Our approach is based on the identity-based cryptography and observations concerning the role of Domain Name System (DNS) and proxy server in VoIP architecture and the role of Home Subscriber Server (HSS) and Call Session Control Function (CSCF) in IP Multimedia Subsystem (IMS) architecture. Our proposed infrastructure also utilizes gateways used to interconnect packet switched (PS) and circuit switched (CS) telephony networks to secure caller-ID in the whole telephony system.

The proposed infrastructure produces signed caller-ID in caller side that is verified at callee side to examine caller ID trust. Our approach does not impose any infrastructure change to the existing telephony networks. The main advantage of our proposed approach is that callee can verify the signature of caller and identify his caller-id before answering the call in real time.

In our approach, we propose to use a distributed key infrastructure for building caller-id based signatures. The proposed approach originates from distributed nature of SIP architecture. We benefit from existing elements in SIP architecture to build a key distribution system. As DNS is used by SIP proxies and IMS CSCFs as a reference to find call route, we use it to distribute domain-level cryptographic keys. As inbound SIP proxies and CSCFs are also used to deliver signaling messages to SIP User Agent Client (UAC) and IMS User Equipment (UE) respectively, we also use them to distribute user-level cryptographic keys.

The remainder of the chapter is structured as follows. In Section 4.2, we describe the technical challenges/requirements that should be met in secure origin identification. In Section 4.3, we give a brief introduction on SIP-based IP telephony and Long Term Evolution (LTE) and voice delivery over LTE. In Section 4.4, we discuss VoLTE as the ultimate proposal for voice service on LTE and IP Multimedia Subsystem (IMS) as its architectural framework. In Section 4.5, we briefly introduce identity based cryptography and our identity-based public key infrastructure is proposed in Section 4.6. In Sections 4.7, 4.8 and 4.9, we propose our implicit authentication method, identity-based signature generation and verification methods for VoIP and VoLTE respectively. In Section 4.10, we explain how gateways are utilized in our infrastructure to restore caller-id trust for whole telephony network. In Section 4.11, we describe some techniques for bootstrapping the adoption of our proposed methods.

4.2 Challenges

Related works for origin identification have been mainly proposed to determine the true identity of callers in SIP based VoIP networks by attaching a signature to the call setup messages (e.g., a SIP INVITE). However, SIP messages may pass through gateways or other entities to interwork with other circuit switched telephony systems such as PSTN.

Existing approaches are not applicable in the existing communications environment where a call may traverse both PS and CS networks. Some of such scenarios, that need to be supported by any origin identification method, can be summarized as follows:

- **PS-to-PS Call:** the calling parties use PS service providers such as VoIP and the call does not transit any CS telephony networks such as PSTN.

- **PS-CS-PS Call:** the calling parties use different PS-based service providers that are not directly connected by PS networks and the call traverses a CS telephony network interconnecting two PS service providers. Consequently, the calling network uses a PS/CS gateway to route the call via a CS network to a CS/PS gateway to break out into the PS network again.

- **PS-to-CS Call:** the call is originated in a PS service provider and terminated in a CS telephony network. Hence the call hits a PS/CS gateway, such as a VoIP/PSTN gateway, after traversing the calling PS network toward the called CS network.

- **CS-to-PS Call:** the call is originated in a cs telephony network and terminated in a PS service provider network. Hence the call hits a CS/PS gateway, such as a PSTN/VoIP gateway, after traversing the calling CS network toward the called PS network.

- **CS-PS-CS Call:** the calling parties use CS phones in different networks that interconnect via a PS network. Consequently, the calling network uses a CS/PS gateway to route the call via an IP network to a PS/CS gateway to break out into the CS again.

- **CS-to-CS Call:** the calling parties use legacy CS networks such as PSTN and GSM cellular networks and the call does not transit any PS service provider network such as VoIP and IMS.

- **Hybrid Call:** the call can be originated and terminated in either PS or CS telephony networks and may traverse several PS and CS networks from caller side to callee side. Hence the call may pass through several CS/PS and PS/CS gateways. As an example, the caller using cellphone can call the callee who uses VoIP phone with *call forwarding* enabled to the callee's cellphone. The callee's cellphone may also have *call forwarding* enabled to another CS or PS network.

SIP messages may also pass through policy enforcement devices in PS telephony networks such as back-to-back user agent (B2BUA). Such intermediaries may modify field of SIP messages or even recreate them and break end-to end integrity. In one of the recent related works, AuthLoop [57] is designed to provide cryptographic authentication solely within the voice channel to work independent of the underlying technology. However it is difficult to deploy AuthLoop which needs significant modifications in the current telephony ecosystem to be adopted. Based on the works of Peterson et al. in [55] and Song et al. in [70], the requirements that should be fulfilled by an applicable origin identification method can be summarized as follows:

1. **Usability:** the caller-ID verification method must work in a passive mode without any human interaction such as resolving puzzles or CAPTCHAs.

2. **Deployability:** the caller-ID verification method must support the existing communications environment including both PS and CS networks and the presence of policy enforcement devices such as B2BUAs in the call path.

3. **Validation by intermediaries:** the caller-ID verification method must provide not only the end systems but also intermediate elements with the ability to verify the caller-ID.

4. **Minimal payload overhead:** the caller-ID verification method must impose as minimum extra overhead as possible to the SIP messages to avoid their fragmentation.

5. **Real time:** the caller-ID verification method must verify the identity of calling parties in real time before answering the call as telephony is a real-time communication.

The goal of this work is then to propose a caller-ID based public key infrastructure that is used for secure call origin identification meeting the mentioned requirements. We first detail the design of the proposed lightweight infrastructure and show how to use it for determination of true identity of callers in PS-to-PS calls. We then explain our method to utilize gateways as parts of the distributed infrastructure for caller-ID verification in all mentioned call scenarios to restore caller-ID trust for whole telephony networks.

4.3 Background

4.3.1 SIP-based IP Telephony

As the standard signaling protocol for VoIP and VoLTE, SIP [42], is a text-based application level protocol to set up, modify, and tear down multimedia sessions between one or more participants. SIP call control uses the *Session Description Protocol* (SDP) [35] for describing multimedia session information.

4.3.1.1 SIP Architectural Components

There are two basic types of components in SIP; end devices referred to as *user agents* (UAs) and *SIP servers.* Each UA (i.e. IP phone and LTE smartphone) combines two sub-entities: the connection requester referred to as the *user agent client* (UAC) and the connection request receiver referred to as the *user agent server* (UAS). Consequently, during a SIP session, both UAs switch back and forth between UAC and UAS functionalities.

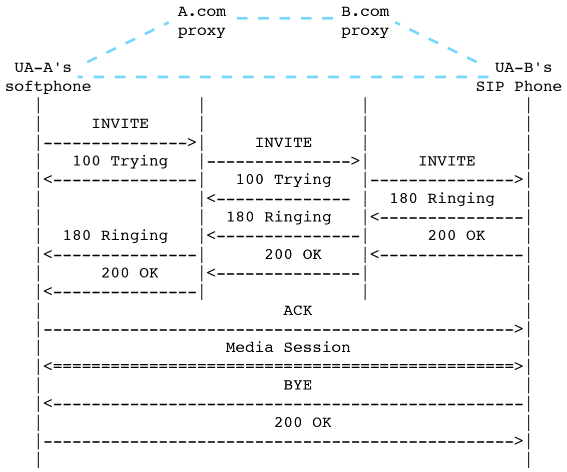


Figure 4.1: SIP Call Flow for the SIP Trapezoid Example

RFC 3261 [62] describes four types of SIP implementation dependent logical servers: *Location Servers*, *Redirect Servers*, *Registrar Severs* and *Proxy Servers*.

4.3.1.2 SIP Messages and Operations

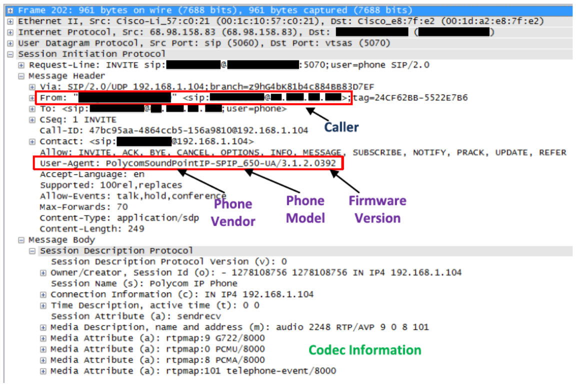
Influenced by two widely used Internet protocols; namely, the *Hypertext Transfer Protocol (HTTP)* [28] and the *Simple Mail Transfer Protocol (SMTP)* [44], SIP messages consisting of request-response pairs are exchanged for call set up, from six kinds consisting of `INVITE`, `ACK`, `BYE`, `CANCEL`, `REGISTER`, and `OPTIONS` - each identified by a numeric code according in RFC 3261 [62]. Other methods are proposed as the extensions of the original six methods. For each request of a UAC, SIP server (or UAS) generates a SIP response. Each response message is also identified by a numeric status code.

4.3.1.3 VoIP Call Flow

Now, we give an example of a typical call setup flow to highlight the usage of SIP request and response messages between user agents UA-A and UA-B. Suppose that the two UAs belong to different domains, which have their own proxy servers in a SIP trapezoid arrangement. As shown in Fig. 4.1, UA-A calls UA-B using its SIP phone over the Internet. The outbound proxy server uses the Domain Name System to locate the inbound proxy server at the other domain. After obtaining the IP address of the inbound proxy server, the outbound proxy server of UA-A sends the `INVITE` request to the domain of UA-B. The inbound proxy server consults a location service database to find out the current location of UA-B, and forwards the `INVITE` request to the UA-B's SIP phone. As the caller ID of UA-A is carried by its `INVITE` request to UA-B, we discuss the header fields of `INVITE` message in detail in Section 4.3.1.4. Exchanging `INVITE`, `200 OK` and `ACK` messages completes the three-way handshake and establishes a SIP session. A set of parameters are exchanged via SIP messages (in the message body using Session Description Protocol (SDP) [35]) between the two end points before a RTP-based voice channel is established. In general, the path of media packets is independent of that of the SIP signaling messages. At the end of the call, UA-B (or UA-A) hangs up by sending a `BYE` message. Subsequently, UA-A (or UA-B) terminates the session and sends back a `200 OK` response. This example shows the basic functionality of SIP, and the detailed description of the SIP operations is in RFC 3261 [62].

4.3.1.4 `INVITE`'s Header Fields

`INVITE` is a SIP request that specifies the action of a call that the requester (UA-A) wants the server (UA-B) to take. The `INVITE` request provides additional information about a message using its header fields. Fig. 4.2 shows the `INVITE` and its minimum required set of header fields including a unique identifier for the call, the destination address, UA-A's address, and information about the type of session that UA-A wishes to establish with UA-B. The complete set of SIP header fields is detailed in RFC 3261 [62]. The header



```
⊞ Frame 202: 961 bytes on wire (7688 bits), 961 bytes captured (7688 bits)
⊞ Ethernet II, Src: Cisco-Li_57:c0:21 (00:1c:10:57:c0:21), Dst: Cisco_e8:7f:e2 (00:1d:a2:e8:7f:e2)
⊞ Internet Protocol, Src: 68.98.158.83 (68.98.158.83), Dst: [redacted] ([redacted])
⊞ User Datagram Protocol, Src Port: sip (5060), Dst Port: vtsas (5070)
⊟ Session Initiation Protocol
  ⊞ Request-Line: INVITE sip:[redacted]@[redacted]:5070;user=phone SIP/2.0
  ⊟ Message Header
    ⊞ Via: SIP/2.0/UDP 192.168.1.104;branch=z9hG4bK81b4c884BB83D7EF
    ⊞ From: "[redacted]" <sip:[redacted]@[redacted].[redacted].[redacted].[redacted]>;tag=24CF62BB-5522E7B6
    ⊞ To: <sip:[redacted]@[redacted].[redacted].[redacted].[redacted];user=phone>
    ⊞ CSeq: 1 INVITE
      Call-ID: 47bc95aa-4864ccb5-156a9810@192.168.1.104
    ⊞ Contact: <sip:[redacted]@192.168.1.104>
      Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, INFO, MESSAGE, SUBSCRIBE, NOTIFY, PRACK, UPDATE, REFER
      User-Agent: PolycomSoundPointIP-SPIP_650-UA/3.1.2.0392
      Accept-Language: en
      Supported: 100rel,replaces
      Allow-Events: talk,hold,conference
      Max-Forwards: 70
      Content-Type: application/sdp
      Content-Length: 249
  ⊟ Message Body
    ⊟ Session Description Protocol
        Session Description Protocol Version (v): 0
      ⊞ Owner/Creator, Session Id (o): - 1278108756 1278108756 IN IP4 192.168.1.104
        Session Name (s): Polycom IP Phone
      ⊞ Connection Information (c): IN IP4 192.168.1.104
      ⊞ Time Description, active time (t): 0 0
        Session Attribute (a): sendrecv
      ⊞ Media Description, name and address (m): audio 2248 RTP/AVP 9 0 8 101
      ⊞ Media Attribute (a): rtpmap:9 G722/8000
      ⊞ Media Attribute (a): rtpmap:0 PCMU/8000
      ⊞ Media Attribute (a): rtpmap:8 PCMA/8000
      ⊞ Media Attribute (a): rtpmap:101 telephone-event/8000
```

Figure 4.2: Header Fields of `INVITE` Message

fields that we frequently refer to in this section are briefly described below:

- `Via`: contains the path traversed by the request so far including the address (pc33.A.com) at which responses to this `INVITE` request is expected to be received. It also contains a `branch` parameter value that must be globally unique across space and time for all requests sent by the UA-A to identify this transaction.

- `To`: contains a display name (UA-B) and a SIP, SIPS URI (sip:UA-B@B.com) or tel URL [64] as the address-of-record of the user or resource that is the target of this `INVITE`. Display names are described in [60].

- `From`: similar to `To`, contains a display name (UA-A) and a SIP or SIPS URI (sip:UA-A@A.com) or tel URL as the address-of-record of the user or resource that is the originator of this `INVITE`. `From` includes `tag` parameter containing a random string (1928301774) to be used for identification purposes. This header field is interpreted as Caller ID and used by callee to identify the caller.

- `Call-ID`: contains a globally unique identifier for this individual call. `Call-ID` coupled with `To` and `From` tags creates a peer-to-peer SIP relationship between UA-A and UA-B and is referred to as a dialog.

4.3.2 Long term evolution (LTE)

New mobile communication technologies have emerged in recent years due to the high demand for new multimedia services such as web browsing, interactive gaming, mobile TV, video streaming, IP telephony and live conferencing. These attractive multimedia services require high-speed data rate. Hence the Third Generation Partnership Project (3GPP) has introduced the Long Term Evolution (LTE) standard to provide high speed wireless communications on the way towards 4G mobile networks.

LTE uses Orthogonal Frequency-Division Multiple Access (OFDMA) and Single-Carrier Frequency-Division Multiple Access (SCFDMA) [30], to increase the bandwidth of wireless data networks up to 100Mbps and 50Mbps in downlink and uplink respectively [23]. The LTE network architecture is designed to be an all-IP network without any circuit-switched domain, containing less network elements, to significantly reduce transfer latency. 3GPP also decided to separate the user plane and the control plane (signaling) to make the scaling independent and let the operators adapt their network easily.

Fig. 4.3 shows a typical LTE network and its three main elements, User Equipment (UE), Evolved Node B (eNodeB) and Evolved Packet Core (EPC), in more detail. UE is connected to the EPC over E-UTRAN (LTE access network) via eNodeB. The eNodeB is the base station for LTE radio. EPC as the main component of the LTE architecture is composed of below network elements:

- Serving Gateway (S-GW): interconnects the E-UTRAN (radio-side of LTE) and the EPC to serve the UE by routing the incoming and outgoing IP packets.
- Packet Data Network Gateway (P-GW): interconnects the EPC and the external IP networks (Packet Data Network) such as IP Multimedia Subsystem (IMS) and routes packets to and from them.

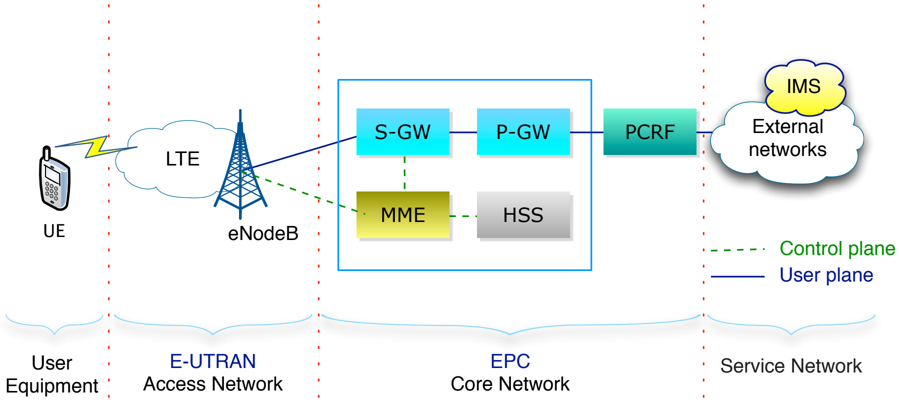


Figure 4.3: Main LTE Elements: UE, eNodeB and EPC

- Mobility Management Entity (MME): handles the signaling related to mobility and security for E-UTRAN access and is responsible for the tracking and the paging of UE in idle-mode.

- Home Subscriber Server (HSS): is a database that contains information related to subscribers and provides support functions in mobility management, call setup and subscriber authentication.

Due to the introduction of the new characteristics to LTE, it incurs a lot of new security challenges in the design of the security architectures of the LTE.

4.3.2.1 Voice Calls over LTE

In spite of providing various multimedia services using its flat all-IP architecture, the voice telephony still remains an essential service for LTE operators. LTE proposes a long term solution for packet-based voice telephony, named VoLTE (one voice), that utilizes the IP Multimedia Subsystem (IMS) network. However, because of the sophisticated infrastructure of IMS that costs a lot of time and money to be fully implemented by operators, VoLTE is yet to be operational. Hence many interim solutions have been proposed for early LTE deployments. Main existing approaches that are standardized by either 3GPP or other forums for providing voice service in LTE can be summarized as below:

- CSFB (Circuit Switched Fallback): in this method, LTE just provides data services using its packet-based network, and falls back from LTE to legacy CS-based domain, 2G/3G, during a voice call [12]. However, the disadvantage is longer call setup delay.

- SVLTE (Simultaneous Voice and LTE): this solution is solely based on the handset that works simultaneously in both LTE PS (packet-switched) mode and CS (circuit switched) mode for providing data services and voice service respectively. The disadvantage is the high power consumption of the phone.

- VoLGA (Voice over LTE via Generic Access): this approach, that has been proposed by the VoLGA Forum, adapts Universal Mobile Access/Generic Access Network (UMA/GAN) system to LTE in the way that a mobile handset can perform voice calls over a customer's private Internet connection such as Wi-Fi [29].

- OTT (Over-the-top): in this approach, VoIP proprietary applications such as Skype and Google Talk are used to deliver voice service in LTE. OTT approach has not been initiated by operators and cannot receive much support in future, as no operator can completely handover its main driver of revenue, voice, to the proprietary applications.

In contrast to the interim solutions, VoLTE as the ultimate proposal for voice service on LTE, that is defined by the GSM Association (GSMA), is based on the IP Multimedia Subsystem (IMS) network and does not require the use of the legacy circuit switched voice networks. In VoLTE, LTE delivers both control and media planes using its all-IP network to provide multimedia services. Since the LTE coverage may be limit for early LTE deployment, it is vital to handover from LTE to legacy networks in the boundaries of

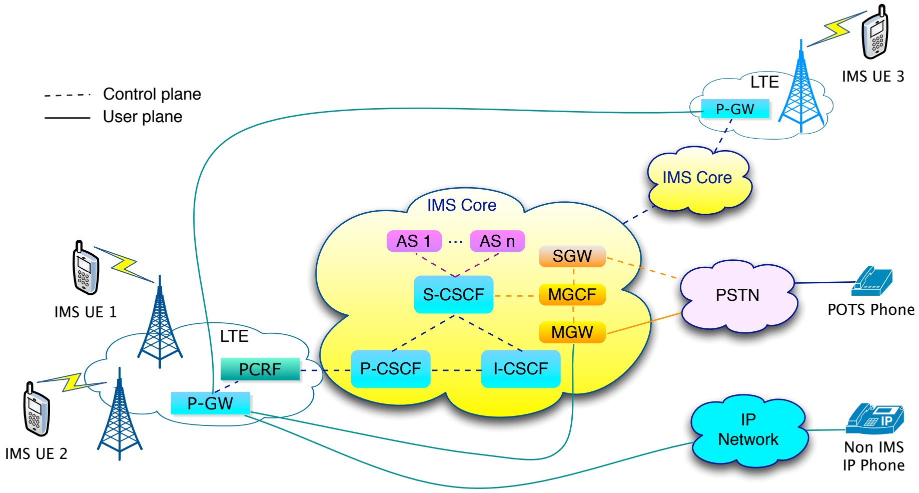


Figure 4.4: Elements of IP Multimedia Subsystem

LTE coverage. SRVCC (Single Radio Voice Call Continuity) is the LTE functionality to allow a VoLTE call in the LTE PS domain to be moved to a legacy CS domain to ensure the continuity of the voice calls.

4.4 VoLTE

LTE provides high throughput and low latency capabilities for mobile operators to migrate voice from their congested and costly circuit switched networks to an efficient and simplified all-IP network architecture. The global association of mobile operators (GSMA) in February 2010 decided that voice services over LTE shall use the IMS platform standardized by the 3GPP [25]. VoLTE is enabled through the tight interworking between IMS and EPC functions mentioned in Section 4.3.2. Hence we discuss main elements of IMS followed by VoLTE call flow in this section.

4.4.1 IP multimedia subsystem (IMS)

IMS is an IP-based network which is deployed by operators to provide various multimedia services. IMS decomposes the networking infrastructure into three separate functions, Access Plane, Control Plane and Application Plane interconnected via standardized interfaces. As shown in Fig. 4.4, the main elements of IMS infrastructure can be summarized as follows:

4.4.1.1 Access Plane

The access plane provides various ways for users to connect to IMS. Users can connect to IMS either directly if they use IP to be considered as SIP user agents or via gateways if they use other phone systems like PSTN and non IMS-compatible VoIP systems.

4.4.1.2 Signaling Plane

The signaling plane routes the call signaling, allows traffic flows from the access plane and provides the users with billing information. The core of this plane, as the most important part of IMS, is called the Call Session Control Function (CSCF) and plays several roles of SIP servers or proxies to process SIP signaling packets in the IMS. CSCF comprises the following functions:

- Proxy-CSCF (P-CSCF): as the first point of contact, acts as a SIP proxy for IMS users. The P-CSCF can authenticate subscribers and inspect every signaling messages to prevent spoofing attacks and replay attacks.

- Serving-CSCF (S-CSCF): as the central brain of the signaling plane acts not only as a SIP server, but also as a session controller. The S-CSCF processes SIP registrations to record the location of each user, and decides to which application servers the

SIP message will be forwarded to provide their services. It also looks up Electronic Numbering (ENUM) table to handle routing services.

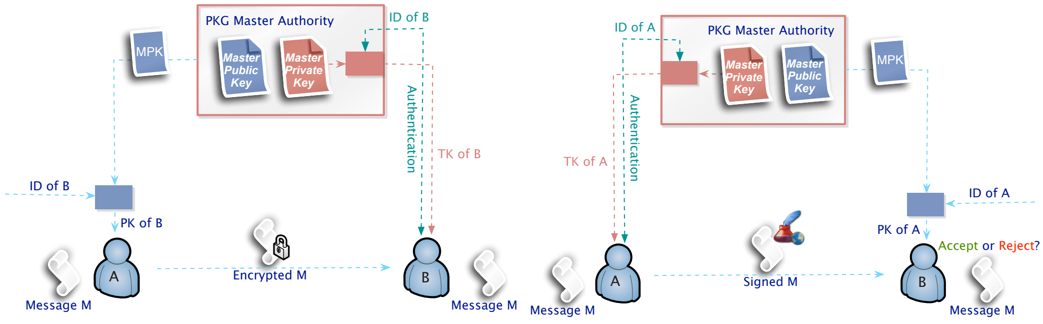
- Interrogating-CSCF (I-CSCF): as another SIP function whose IP address is published in the Domain Name System (DNS) of the domain, can be contacted from peered networks and used to forward SIP packets. It queries the HSS to determine the S-CSCF for a user as well as forwarding SIP request or response to the S-CSCF.

4.4.1.3 Application Plane

The application plane provides SIP application servers (AS) for the provision and management of services. This plane defines standard interfaces to HSS, PCRF and CSCFs for identity management, billing services and control of voice calls respectively.

4.5 Identity Based Cryptography

Shamir [68] in 1984 proposed a new concept in cryptography, named identity-based cryptography, to derive the user's public key from his identity. Although his concept remained an open problem for many years, Guillou and Quisquater [34] in 1988, and Boneh and Franklin in 2001 [18] finally proposed the first implementation for identity-based signatures (IBS) and the first implementation for identity-based encryption (IBE) respectively. Using this approach, a master private key along with its master public key are generated. The master public key is published by a master authority, named Private Key Generator (PKG), to be used publicly to mathematically derivative user's public key from his identity. The master private key is kept by PKG to generate the user's corresponding private key and is only delivered to the authorized user after authentication. Fig. 4.5a and Fig. 4.5b



(a) Identity-Based Encryption (b) Identity-Based Signature

Figure 4.5: Schematic Outline of IBE and IBS Schemes

illustrate a schematic outline of an IBE scheme and its mirror image as an identity-based signature (IBS) scheme respectively.

Identity-based cryptography radically simplifies key management in two ways:

1. The master public key is saved in the database for an organization instead of recording individual public keys for its users.

2. Private keys are constructed mathematically based on the master private key in PKG that eliminates the need for a database to save private keys and makes the process of key recovery straightforward.

4.6 Proposed Identity-Based Public Key Infrastructure

Although the theoretical identity-based cryptography considers a single master public key to be used for public key derivations, it is practically insufficient as each organization prefers to have its master public key that makes its users' public keys manageable. Originating from the distributed nature of SIP architecture, we propose for each SIP domain and IMS network to have its own master authority. SIP registrar for SIP telephony and S-CSCF for IMS network can be selected as the master authority. In our proposed public key infrastructure, each SIP domain that can also be an IMS network generates its master public key and master private key.

Since the master public key must be distributed in a way that can be requested easily to mathematically derivative public keys, we use the DNS system to distribute the master public keys. Private keys in our infrastructure are computed in the master authority based on the master private key and the identity of user and securely transmitted to the authorized user after authentication. Hence we rely on the existing secure connection between users and their registrar, SIP registrar for UACs and S-CSCF for UEs, to transmit private keys to the corresponding users. In summary, the master authority, that can be SIP registrar and S-CSCF, sends its master public key to DNS and keeps the master private key for providing its users with their private keys.

4.7 implicit authentication

As we modify registration phase of SIP and IMS to provide implicit authentication, we first briefly describe the SIP and IMS registration procedures [43] in next two subsections, before providing the detail of our method design.

4.7.1 The SIP Registration Procedure

Registration is a mandatory procedure in SIP by which SIP servers can learn the current location of their UACs. A UAC sends SIP `REGISTER` messages to a specific SIP server named SIP registrar upon initialization and at periodic intervals. The registrar writes the binding of SIP URI and its current location for the UAC to the location server. The location server generally contains information that allows a SIP proxy server to input a URI and receive a set of zero or more URIs for routing incoming SIP requests and has no role in authorization and authentication of outgoing requests.

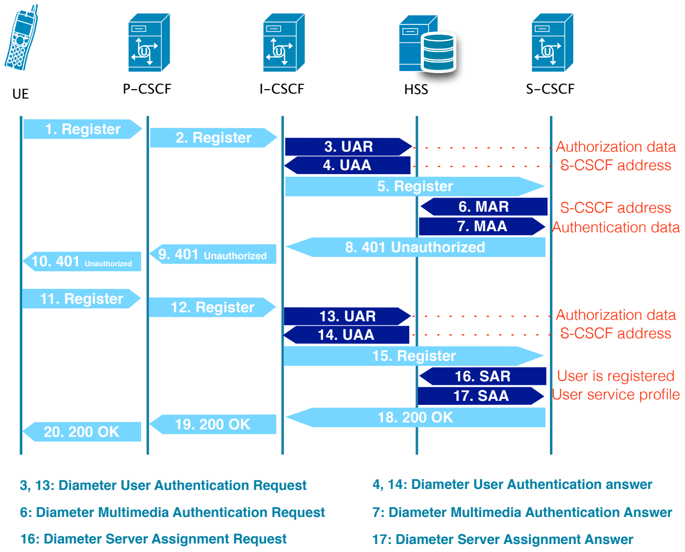


Figure 4.6: The IMS registration process

4.7.2 The IMS Registration Procedure

The IMS registration is a common procedure in which the IMS UE registers to the IMS network. The UE shares its personal information such as the IP multimedia public identity (IMPU) and the IP multimedia private identity (IMPI) with the core network database (HSS) to register to IMS network. IMPU is the SIP URI of the UE (sip:UE-A@IMS-A.net) that SIP servers use to route the SIP messages to the UE. IMPI is also used for authorization and authentication by S-CSCF in the IMS domain (UE-A@IMS-A.net).

As shown in Fig. 4.6, the UE first sends a SIP `REGISTER` request to the P-CSCF server. The P-CSCF forwards the request to the I-CSCF in the user's home IMS network. Based on some information dynamically carried by the Diameter protocol from HSS, I-CSCF then routes the request message to a S-CSCF as the serving node for this UE among several S-CSCFs in the IMS domain. Hence in addition to be the serving node, the S-CSCF also acts as the SIP registrar for the UE. After authenticating and authorizing the UE, the S-CSCF using the Diameter protocol informs the HSS of UE registration and the HSS can download the user profile.

4.7.3 Method Design

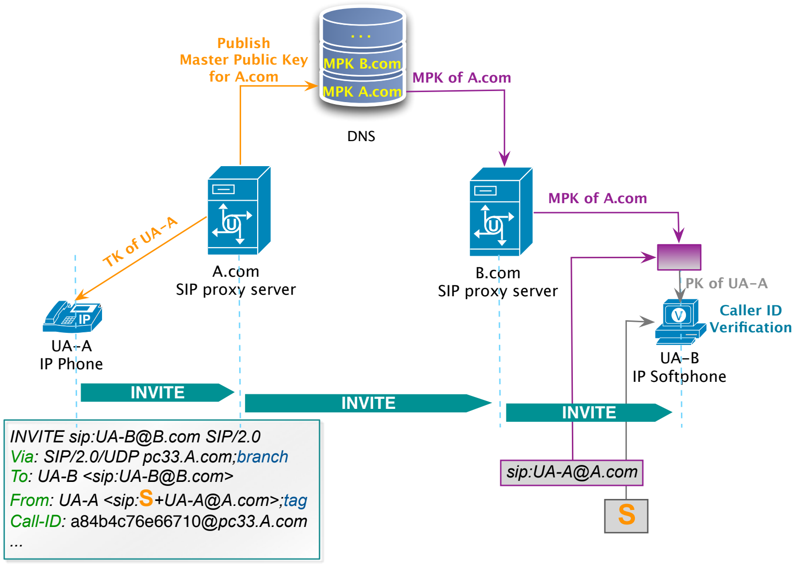
We modify registration phase of SIP and IMS to extract three specific properties of a registering user to make its unique profile and record it for future implicit authentication. These properties are as follows:

1. Mac address as a unique identifier of calling device (M): the calling device, that can be either a SIP hardphone/softphone or IMS cellphone (LTE phone), should include the hash of its mac address ($H_M = hash(M)$) in the `REGISTER` message.

2. IP address as a location identifier (I): the SIP registrar can extract IP address of caller from the `REGISTER` message. This IP address can be either the exact IP address of caller or its NAT address.

3. SIP Uniform Resource Identifier (URI) or tel URL as caller-id (U): the SIP registrar can also extract SIP or tel URL directly from the `REGISTER` message.

While a UAC or UE is registering to a SIP registrar or S-CSCF, a hash over the combination of these three identifiers is generated ($H = hash(H_M|I|U)$) and recorded, as a profile for this device, to location server or HSS for VoIP or VoLTE respectively. The caller device is supposed to sign H_M coupled with `tag` parameter ($S_{H_M} = Sign(H_M|tag)$) by its private key, that has been already received in registration procedure from SIP proxy or S-CSCF, and include it in `branch` parameter of `Via` filed of `INVITE` message to initiate call. Outbound SIP proxy in VoIP and S-CSCF in VoLTE, as the first point of receiving this request message in service network, verifies the signature of S_{H_M} by public key of the caller, decouples `tag` and extracts H_M. Putting H_M, I and U from `INVITE` message together,



```
INVITE sip:UA-B@B.com SIP/2.0
Via: SIP/2.0/UDP pc33.A.com;branch
To: UA-B <sip:UA-B@B.com>
From: UA-A <sip:S+UA-A@A.com>;tag
Call-ID: a84b4c76e66710@pc33.A.com
...
```

S = sign(UA-A@A.com XOR UA-B@B.com) by TK of UA–A

Figure 4.7: Identity-based Signatures for VoIP

generating H and comparing it against corresponding record, the server can implicitly authenticate the caller.

Using these three identifiers, we can assure that the caller is calling from the same location using the same device, otherwise the `INVITE` request is rejected and caller is asked for explicit authentication and reregistration. The caller can be explicitly authenticated to add other devices and bind other service locations.

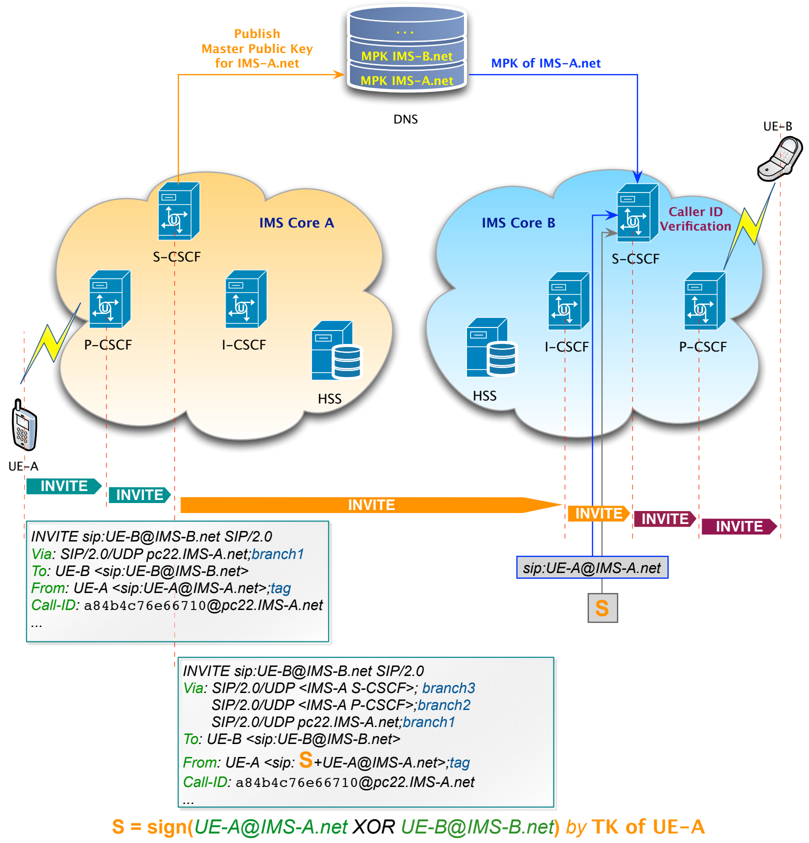
4.8 Identity-based signatures to verify caller-id for VoIP

We propose to use the identity-based public key infrastructure to sign a part of `INVITE` request in caller side to be verified in callee side as the caller-ID trust. As illustrated in Fig. 4.7, the caller UAC signs (caller's SIP URI or tel URL + callee's SIP URI or tel URL) by its private key, that has been already received in registration procedure from SIP proxy. "+" operator can be a simple XOR. Signature of (caller's SIP URI or tel URL + callee's SIP URI or tel URL) is named S. (S || caller's SIP URI or tel URL) is then used in `From` header of `INVITE` message. As `From` and `To` headers are not removable and should remain unchanged even if in presence of B2BUAs, the `From` header that acts as caller-ID remains unchanged while original and regenerated `INVITE` requests flow through the networks. The `INVITE` request including the signature is then sent to the outbound SIP proxy to be forwarded step by step toward the callee.

The callee side receiving the `INVITE` request finds the domain part of the caller's SIP URI or tel URL (that can be extracted from the end of the `From` header) and asks its SIP proxy to look up DNS for the corresponding master public key. The callee then mathematically generates the caller's public key using the master public key and SIP URI or tel URL of the caller as its identity. Using the caller's public key, the callee verifies the signature S in the `From` header. The caller ID can be trusted if the signature is verified against (caller's SIP URI or tel URL + callee's SIP URI or tel URL), otherwise the caller ID is considered to be spoofed.

4.9 Identity-based signatures to verify caller-id for VoLTE

VoLTE can use the signatures similar to what we propose in Section 4.8 for caller ID verification. In this case for VoLTE, S-CSCF can be used to play the same role as SIP proxy server does in VoIP for providing caller and callee with private key and master public key respectively. However, identity-based public key generation is computationally power consuming and the UEs are generally smartphones with limited power sources in comparison to SIP UACs that can be either softphone or hardphone supplied with more power. Hence we modify the proposed method in Section 4.8 to assign the tasks of sign, key generation and verification to S-CSCF instead of UE itself. In case that our proposed



```
INVITE sip:UE-B@IMS-B.net SIP/2.0
Via: SIP/2.0/UDP pc22.IMS-A.net;branch1
To: UE-B <sip:UE-B@IMS-B.net>
From: UE-A <sip:UE-A@IMS-A.net>;tag
Call-ID: a84b4c76e66710@pc22.IMS-A.net
...
```

```
INVITE sip:UE-B@IMS-B.net SIP/2.0
Via: SIP/2.0/UDP <IMS-A S-CSCF>; branch3
     SIP/2.0/UDP <IMS-A P-CSCF>;branch2
     SIP/2.0/UDP pc22.IMS-A.net;branch1
To: UE-B <sip:UE-B@IMS-B.net>
From: UE-A <sip: S+UE-A@IMS-A.net>;tag
Call-ID: a84b4c76e66710@pc22.IMS-A.net
...
```

S = sign(UE-A@IMS-A.net XOR UE-B@IMS-B.net) by TK of UE–A

Figure 4.8: Identity-based Signatures for VoLTE

implicit identification method is also used at the same time between UE and its S-CSCF, signature of S_{H_M} and its verification can be done based on a simple and light RSA public key infrastructure.

As illustrated in Fig. 4.8, the caller UE first sends the `INVITE` request to its P-CSCF. The P-CSCF then sends it again to its S-CSCF whose address has been already extracted in registration process. The S-CSCF receiving the `INVITE` request signs (caller's SIP URI or tel URL + callee's SIP URI or tel URL) by the UE's private key, named S. (S || caller's SIP URI or tel URL) is then used in `From` header of `INVITE` message. The `INVITE` request including the signature is then forwarded step by step toward the callee.

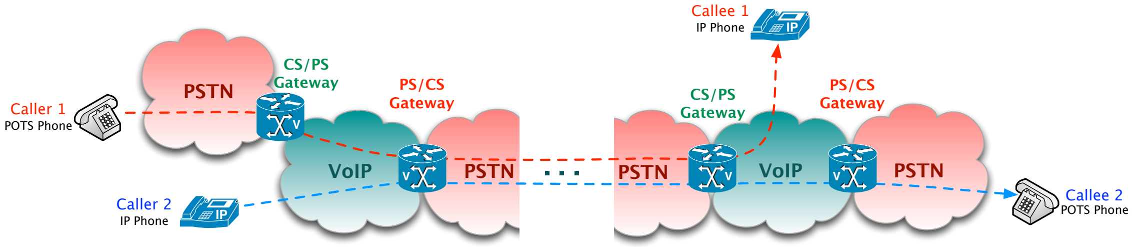


Figure 4.9: The General Cases of a Hybrid Call

The S-CSCF in the callee's IMS network receiving the `INVITE` request finds the domain part of the caller's SIP URI or tel URL (that can be extracted from the end of the `From` header) and looks up DNS for corresponding master public key to mathematically generate the caller's public key using the master public key and SIP URI or tel URL of caller as its identity. Using the caller's public key, the S-CSCF verifies the signature S in the `From` header. The caller ID can be trusted if the signature is verified against (caller's SIP URI or tel URL + callee's SIP URI or tel URL), otherwise the caller ID is considered to be spoofed. The callee is then informed if the caller ID is dependable.

4.10 Caller ID as an dependable Identity for whole telephony system

In previous sections, we presented our design for providing caller authentication and secure origin identification by attaching a signature to the SIP `INVITE` request. Our design is based on IP telephony infrastructure including both VoIP and VoLTE. However calls may still traverse the PSTN or CS legacy cellular networks at some point. These calls may be originated in PSTN or CS legacy networks as their endpoints may be a PSTN or CS-based cellular endpoints. They may also be originated as VoIP or VoLTE calls but transmitted over PSTN or CS legacy networks at some points. As a consequence, interconnecting between SIP based telephony and other CS legacy telephone networks has ultimately reduced the security of Caller ID systems. In such cases, call request messages may pass through gateways that interconnect different networks. Hence we extend our proposed infrastructure to utilize gateways as intermediate points for signature generation and verification.

Gateways are devices with different interfaces to interconnect two different networks. Telephony gateways play two important roles; media gateways convert the media format between two networks and signaling gateways also convert control messages from the format required for one type of network to the format required for another. We categorized telephony gateways in two broad categories of *"PS to CS gateways"* and *"CS to PS gateways"*. As an example of *"PS to CS gateway"*, VoIP to PSTN (SIP/IUSP) signaling gateways are supposed to change the format of control messages from SIP to SS7 (Signaling System No. 7) and VoIP to PSTN media gateways are supposed to convert digital signal into an analog signal to be sent across the PSTN.

We extend our proposed infrastructure by assigning signature generation and verification to gateways in addition to calling parties' domains. In our design, a *"CS to PS gateway"* is responsible for signature generation and a *"PS to CS gateway"* is responsible for signature verification using the proposed identity-based public key infrastructure following the below rules.

- CS-to-PS gateway such as PSTN/VoIP (ISUP/SIP)

    1. If the gateway belongs to the service provider of the callee, the gateway does not need to generate any signature.
    2. Otherwise, the gateway should locally sign the control message while changing its format from ISUP to SIP.

- PS-to-CS gateway such as VoIP/PSTN (SIP/ISUP)

  1. If the gateway belongs to the destination network that serves the callee, the gateway should verify the signature in the control message while changing its format from SIP to ISUP.
  2. If the gateway interconnects the service provider of the caller to the next network in the call path, the gateway does not need to verify the signature.
  3. Otherwise, the gateway should verify the signature in the control message, that has been locally generated in the previous CS-to-PS gateway, while changing its format from SIP to ISUP.

Based on the mentioned guidelines for signature generation/verification in gateways and the role of proxy servers and S-CSCF in PS telephony networks for signature generation/verification, a caller-ID is supposed to be signed at most one time and to be verified at most one time in the call path. Fig. 4.9 shows two hybrid calls as the general cases where the call can be originated and terminated in either PS or CS telephony networks and may traverse several PS and CS networks from caller side to callee side. Other call scenarios mentioned in Section 4.2 are considered as specific cases that can be derived from the general case of the hybrid call.

4.11 How to Get Started: Bootstrapping the Adoption of Our Proposed Methods

As we propose in Sections 4.8 and 4.9, S-CSCF for VoLTE and SIP proxy server for VoIP should be upgraded to generate the master keys for their domains and DNS should be also upgraded to distribute master public keys. The most challenging aspect of our proposed solutions is the path of upgradations to adoption and deployment. Hence this section describes some techniques for bootstrapping the adoption of our proposed methods.

4.11.1 Third Party Authority Center

A calling user itself or its domain can choose an alternate master authority center to generate master keys in case the domain is not upgraded yet to provide the keys. In such cases, the third party authority center provides the users of a domain, that has already subscribed to the center, with the master keys. To obtain the keys, users are supposed to explicitly log in to the third party authority center. A callee user itself or its domain subscribes to the alternate master authority center to obtain the master public key for caller ID verification. Users in this case are also required to log in to the third party authority center to obtain the keys.

4.11.2 Calling Domain as Both Signer and Verifier

As using our proposed identity-based public key infrastructure needs DNS to provide callee domain with the master public keys of calling domain, it may take time for DNS to be globally upgraded and keep the master keys in proper fields along with domain names. Hence we also propose an interim solution that can be implemented locally in domains without waiting for DNS to be upgraded. Using this interim solution, each domain can implement its desire public key infrastructure for signature generation independent from other domains. However it has to verify the signatures itself to ensure other party domain that the signature has been provided by the same domain.

In this section, we explain the proposed interim solution in more detail. As we discussed it, it is easy to spoof caller IDs in VoIP and VoLTE, since both voice and control data are transmitted over IP packets. It is also possible to spoof caller IDs using fake ID providers.

In such cases, an attacker calls the fake ID provider and asks for changing his caller ID to the desire caller ID of someone else for attack a victim. The fake ID provider initiates a call to the victim with the desire caller ID and then connects attacker to victim once the call is answered. In other words, fake ID providers pretend to be desired domains to provide fake SIP URI and tel URL as the desired caller ID. This mechanisms motivated us to propose the interim solution in which calling domain is supposed to verify that it has recently signed caller ID for the caller.

Similar to what we propose in Sections 4.8 and 4.9, SIP proxy server in VoIP and S-CSCF for VoLTE can implement desired public key infrastructure for its domain to sign a part of the `From` field in the `INVITE` request from its users to be forwarded to the next step toward callee. The SIP proxy server or IMS S-CSCF in the callee network receiving the `INVITE` request, generates a new message, named VERIFICATION request, based on the `INVITE` request and sends it to the calling domain corresponding to the SIP URI or tel URL in the `INVITE` request. The SIP proxy server or IMS S-CSCF in the calling domain is supposed to verify the signature included in the VERIFICATION request copied from the original `INVITE` request and send back an acknowledgment message to the requester to accept/reject the caller ID. In case that the caller ID was faked, even via fake ID provider, the original SIP proxy server or IMS S-CSCF in the calling domain cannot verify the signature and will reject the caller ID and warn the requester. The requester can either deliver `INVITE` request to the callee while warning about fake ID or ignore `INVITE` request based on the callee domain policy.

Using the proposed interim solution, each call domain can implement its own public key infrastructure that can be a simple RSA public key infrastructure or a one way hash function. In case of using a hash function, the domain is supposed to keep a random message that is added to `From` filed before doing hash to generate signature so that it can

Table 4.1: Call Setup Delays Induced by IBS Schemes

Scheme	Reference	Average induced call setup delay
ZSS	[82]	31 ms
BLS	[19]	50 ms
Paterson	[53]	56 ms
Hess	[38]	65 ms
Zhangkim	[81]	77 ms

verify the VERIFICATION requests. Although the interim solution is simple and can be implemented locally, its disadvantage is longer call setup delay as the signature in `INVITE` request should be sent to the original calling domain for verification.

4.12 Evaluation

In this section, we consider the cost of using our proposed method in terms of additional delay induced to call setup time, and extra overhead. We use the GNU Multiple Precision Arithmetic (GMP) library [71], Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [2] and the pairing-based cryptography (PBC) library [14] to implement our proposed identity-based infrastructure. Using these C libraries, various existing identity-based signature (IBS) algorithms are first implemented and compared to find the most efficient IBS algorithm for our proposed infrastructure. We then implement our infrastructure using the selected algorithm and conduct simulations to evaluate its efficiency.

4.12.1 IBS Algorithm Selection

We first evaluate the efficiency of the existing IBS algorithms using a simple voice over IMS testbed with our infrastructure installed as shown in Fig. 4.10. The IMS system consists of two separate service providers, routers and other data networking elements. The testbed

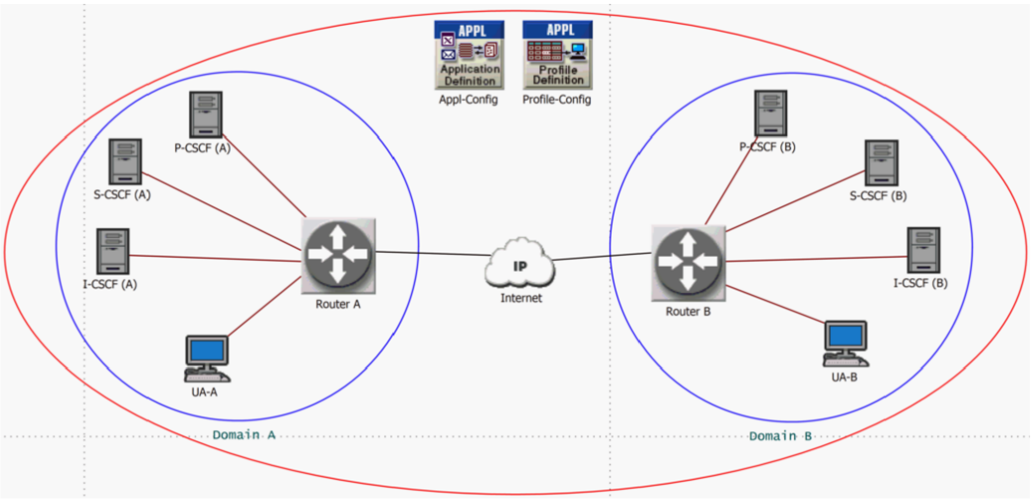


Figure 4.10: Simulation Testbed of the Simple Scenario

is simulated using the network simulator OPNET [52]. We used the IMS model proposed in [11] in this simulation testbed. Each service provider is simulated by three Sun Ultra 10 (333 MHz with 128 Mbytes RAM) machines as proxy, serving and interrogating call session control functions (P/I/S-CSCF) serving a generic Windows PC (733 MHz Pentium III with 128 Mbytes RAM) as IMS UA calling party. The IMS system simulates the scenario of UA-A making calls to UA-B. It is assumed that service providers are intraconnected via 100BaseT Ethernet links and are connected to Internet cloud by DS1 link. The Internet delay between VoIP service provider networks is assumed to be 50 ms with 0.5% packet loss rate. G.729 is used as the voice codec algorithm. The codec has a bit rate of 8 Kbps with 10 milliseconds frame size and 1 frame per packet and setting of Lookahead Size = 5 ms, DSP Processing Ratio = 1, Speech Activity Detection = Enabled. Fig. 4.11 shows call duration for the calls in the experiment that runs for two hours.

Table 4.1 summarizes different IBS schemes used for signature generation/verification in our proposed infrastructure and the average call setup delays imposed using them. Fig. 4.12 also shows the call setup time for initiating call between calling parties using these

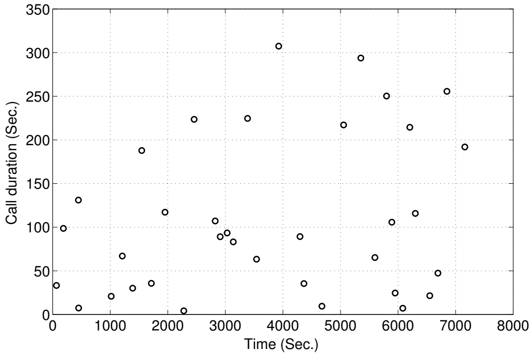


Figure 4.11: Call Duration for Simple Scenario

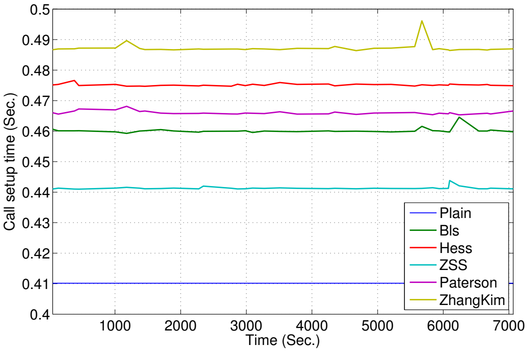


Figure 4.12: Call Setup Time Induced by Different IBS Schemes

different IBS algorithms. It can be concluded that ZSS IBS scheme [82] has the lowest time complexity and imposes the least additional delay to call setup time. Hence we choose the ZSS IBS scheme [82] in our proposed infrastructure for signature generation/verification.

4.12.2 Call Setup Time

In the telephony world that provides a real-time communication, an applicable caller-ID verification method should be able to verify the identity of calling parties in real time before answering the call. If subscribers encounter long connection delays, the proposed method may not be adopted by telephony service providers. Therefore, the extra delays induced

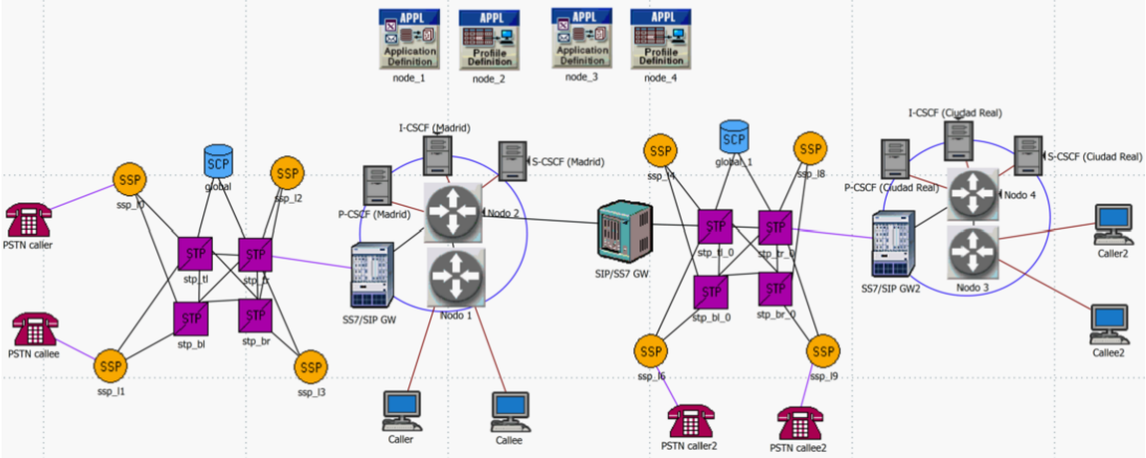


Figure 4.13: Simulation Testbed of the Hybrid Scenario

to call setup times by the verification methods is an important metric.

In general, call setup time is defined as the time interval between initiating the call in caller side and receiving ringback after answering the call in callee side. For example, in PS telephony such as VoIP and VoLTE, the call setup time can be defined as the time difference between sending an `INVITE` request and receiving a corresponding `180 ringing` message.

Using OPNET, we then develop a comprehensive simulation testbed with our proposed infrastructure installed as shown in Fig. 4.13 including both PS telephony such as VoIP and VoLTE, and CS legacy telephony such as PSTN. CS telephony and PS telephony use SS7 and SIP protocols for signaling respectively. PS-to-CS and CS-to-PS gateways are used to change the format of signaling messages for interworking of PS and CS networks. We reused and modified OPNET standard models of PSTAN and SIP and contributed models of SS7 and IMS-SIP to build the testbed.

To simulate the realistic call behaviors, in our experiments that runs for 120 minutes, half of the end users are selected to generate independent calls to the other half of the end

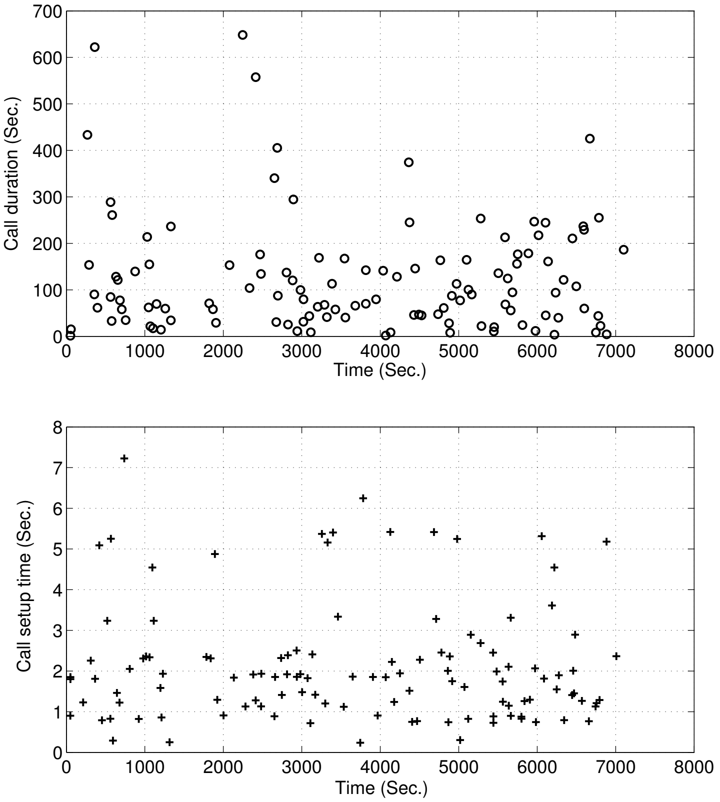


Figure 4.14: Call Duration and Setup Delay for the Hybrid Scenario

users. The call duration and calling interval between calls are also randomly distributed. Fig. 4.14 shows call duration and call setup delay for the whole telephony system in our hybrid scenario. The average call setup time is 2.106 second and the average call setup delays induced by our infrastructure is 27.125 ms that is 1.3% of the call setup time. Such an additional delay of 27.125 ms will be hardly noticeable by today telephony subscribers (i.e. UAs). Therefore, our proposed infrastructure can be adopted by telephony service provider for caller-ID verification in real-time before answering the calls.

4.12.3 Signaling Overhead of Call Setup

Packet-switched based telephony networks such as VoIP and VoLTE are mainly use SIP as their signaling protocol for call initiation and termination. SIP request and response messages have varied length header fields. SIP mandatory fields, including calling parties' addresses and media information consume about 450 bytes. Calling parties exchange three SIP messages to setup a call; `INVITE`, `100 Trying` and `180 Ringing`.

PSTN as a circuit-switched based telephony uses SS7 as the signaling protocol for call initiation and termination. The ISDN User Part (ISUP) is part of SS7 protocol to set up telephone calls in the PSTN. Two main messages are exchanged between calling parties to setup a call; Initial Address Message (`IAM`) and Address Complete Message (`ACM`). ISUP messages also have varied length header fields. Mandatory fields in `IAM` and `ACM` messages including calling parties' numbers occupy about 200 and 400 bytes, respectively.

Our simulation results show that a call in the hybrid simulation scenario introduces 2205 bytes of signaling overhead in average to be setup between calling parties. The average signaling overhead for call setup in case of using our caller-ID verification infrastructure is 2429 bytes. Hence, our proposed infrastructure introduces about 9% extra signaling overhead for call setup. Therefore, our proposed infrastructure is scalable and can be adopted to widely verify caller-ID in telephony networks.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In the traditional PSTN and mobile telephony, the telephony networks consist of intelligent core/edge elements accessed by the simple end devices. These networks use secure protocols designed for circuit-switched networks. All these characteristics provide the whole telephony network with a certain degree of security.

However, the usage of highly programmable and powerful smartphone devices and emerging VoIP and LTE cellular networks has drastically changed the telephony networks. Different from the traditional telephony, the IP-based telephony uses less secure protocols accessed by very smart devices. Such a change in the network infrastructure offers an opportunity to would-be attackers for exploiting potential security vulnerabilities.

In this dissertation, we focused on the security issues in the VoIP/VoLTE solutions for voice delivery in the IP-based telephony and operators networks. We first studied the problem of voice spam in the modern IP telephony and proposed a speaker independent speech recognition scheme for content filtering to avoid spam message deposition and spamming source identification.

We then focused on another important security issue in the IP telephony, caller-ID spoofing, which is one major technique used by various attacks. We proposed a method to establish a relationship between a calling device and its originated signaling messages and media streams. Using this method, we can distinguish and then block malicious softphone based calls.

We finally extended our caller-ID signature-based validation to propose a secure infrastructure covering the whole telephony network including the traditional PSTN and cellular networks, IP telephony and LTE operator networks.

5.2 Future work

In this dissertation, we revealed the security vulnerabilities induced by VoIP that is the emerging alternative to traditional telephony methods. We then mainly focused on voice spam detection and caller-ID related security issues, and proposed our defensive solutions. However, there are still various and new security challenges inherited from the nature of voice delivery over IP in the emerging voice and multimedia networks. In our future work, we will study these new challenges and propose the appropriate solutions to build a secure infrastructure for voice delivery.

In the traditional telephony and cellular networks including PSTN, 2G and 3G, voice and data have seemingly been able to co-exist. While the traditional 2G and 3G radio networks have been used for voice delivery, the LTE (4G) in most of the practical integrations is mainly used to deliver data. However, the LTE infrastructure is designed to deliver both data and voice simultaneously. Operators now realize that the multiple heterogeneous network layers are at their limits in terms of meeting the daily demands of mobile subscribers. These users are habitual to the faster speed and improved user experience that 4G delivers. Their progressively impulsive behaviors and growing demands for pervasive mobile broadband access, create expectations that only 5G networks can meet. The wide integration of 4G by operations all around the globe and emerging 5G paradigm will make VoIP/VoLTE the only option for voice delivery, leading to the significant savings for the operators by switching off 2G and 3G networks.

Voice delivery over IP networks leveraging 4G and 5G technologies has introduced new security challenges that attract considerable research attentions. In this dissertation, we have taken the first step toward securing VoLTE to prevent caller-ID spoofing attacks in such networks. In our future work, we will further improve and extend our current defense solutions to prevent caller-ID related attacks by developing a practical infrastructure of securing voice delivery over IP networks.

Bibliography

[1] Matlab - the language of technical computing. http://www.mathworks.com/products/matlab/.

[2] MIRACL Library. Website, http://www.certivox.com/miracl.

[3] X-Lite, Softphone from CounterPath. http://www.counterpath.com/x-lite/.

[4] BroadSoft, cloud-based Unified Communications solutions Provider, 1998. http://www.broadsoft.com/.

[5] GENBAND S3: The Intelligent Session Border Controller, 1999. http://www.genband.com/products/quantix/sbc.

[6] Acme Packet, the global provider of session border control technology, 2000. http://www.acmepacket.com/.

[7] Linphone: Free SIP VoIP client, 2001. http://www.linphone.org/.

[8] TeleTurd Caller ID Spoofing, the anonymous service to fake the Caller ID, 2001. http://www.teleturd.com/.

[9] Twinkle, SIP based softphone for VoIP and instant messaging communications, 2005. http://mfnboer.home.xs4all.nl/twinkle/.

[10] OpenSBC, an open source (MPL License) Session Border Controller and B2BUA, 2010. http://sourceforge.net/projects/opensipstack/.

[11] AHE Vazquez and JI Fernandez . SIP-IMS Model for OPNET Modeler. 2005.

[12] 3GPP. Circuit switched (cs) fallback in evolved packet system (eps); stage 2. Technical report, 3GPP TS 23.272 V9.3.0., March 2010.

[13] AT&T Labs Research. AT&T natural voices® text-to-speech system. Website, http://www2.research.att.com/ ttsweb/tts/.

[14] B. Lynn. PBC Library. Website, http://crypto.stanford.edu/pbc, 2007.

[15] Armen Babikyan. Sharktools: Tools for programmatic parsing of packet captures using wireshark functionality.

[16] Vijay Balasubramaniyan, Mustaque Ahamad, and Haesun Park. CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation. In *The Fourth Conference on Email and Anti-Spam*, 2007.

[17] Vijay A Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael T Hunter, and Patrick Traynor. Pindr0p: using single-ended audio features to determine call provenance. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 109–120. ACM, 2010.

[18] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in CryptologyCRYPTO 2001*, pages 213–229. Springer, 2001.

[19] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Advances in CryptologyASIACRYPT 2001*, pages 514–532, 2001.

[20] M G Bulmer. *Principles of statistics*. DoverPublications. com, 1979.

[21] Y. Cai. Validating caller id information to protect against caller id spoofing. *Patent Application*, 2008.

[22] Cepstral®. Cepstral text-to-speech engine. Website, http://www.cepstral.com/.

[23] Sunggu Choi, Kyungkoo Jun, Yeonseung Shin, Seokhoon Kang, and Byoungjo Choi. Mac scheduling scheme for voip traffic service in 3g lte. In *Vehicular Technology Conference, VTC-2007*, pages 1441–1445. IEEE, 2007.

[24] Gerald Combs et al. Wireshark: What's on your network? *Web page: http://www.wireshark. org/last modified*, pages 12–02, 2007.

[25] D. Warren. Voice & SMS over LTE Update. Website, https://infocentre.gsm.org, 2011.

[26] Ram Dantu and Prakash Kolan. Detecting spam in voip networks. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, 2005.

[27] D. Ellis. Dynamic time warp (dtw) in matlab. Web resource, http://www.ee.columbia.edu/ dpwe/resources/matlab/dtw/, 2003.

[28] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP1.1. RFC 2616, IETF Network Working Group, 1999.

[29] VoLGA Forum. Voice over lte via generic access; requirements specification. Phase1. VoLGA - Requirements V1.3.1, June 2009.

[30] C Gessner. Umts long term evolution (lte) technology introduction. *Rohde & Schwarz, Tech. Rep. 1MA111*, 2008.

[31] Theodoros Giannakopoulos. A method for silence removal and segmentation of speech signals, implemented in matlab. Web resource, http://www.mathworks.com/matlabcentral/fileexchange/authors/30223, 2010.

[32] Google. Google Voice. Website, www.google.com/voice, 2011.

[33] Duncan Graham-Rowe. A Sentinel to Screen Phone Calls. Website, http://www.technologyreview.com/communications/17300/?a=f, 2006.

[34] Louis C Guillou and Jean-Jacques Quisquater. A paradoxical identity-based signature scheme resulting from zero-knowledge. In *Proceedings on Advances in cryptology*, pages 216–231. Springer-Verlag New York, Inc., 1990.

[35] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, IETF Network Working Group, 1998.

[36] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[37] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[38] Florian Hess. Efficient identity based signature schemes based on pairings. *Selected Areas in Cryptography*, pages 310–324, 2003.

[39] A. Iranmanesh, H. Sengar, and H. Wang. A voice spam filter to clean subscribers mailbox. *Security and Privacy in Communication Networks*, pages 349–367, 2013.

[40] ITU-T. G.711 : Pulse code modulation (PCM) of voice frequencies. Website, http://www.itu.int/rec/T-REC-G.711/, 2013.

[41] Van Jacobson, Craig Leres, Steven McCanne, et al. Tcpdump, 1989.

[42] A.B. Johnston. *SIP Understanding the Session Initiation Protocol.* Artech House, 2nd edition, 2004.

[43] Krisztian Kiss. Signalling flows for the ip multimedia call control based on session initiation protocol (sip) and session description protocol (sdp); stage 3. *3GPP TS Specification*, 24, September 2006.

[44] J. Klensin. Simple Mail Transfer Protocol. RFC 2821, IETF Network Working Group, 2001.

[45] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, 2(2):93–108, 2005.

[46] Ebrahim H Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Electrical Engineers, Proceedings of the Institution of*, 121(12):1585–1588, 1974.

[47] Sue B Moon, Paul Skelly, and Don Towsley. Estimation and removal of clock skew from network delay measurements. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 227–234. IEEE, 1999.

[48] Sivannarayana Nagireddi. *VoIP Voice and Fax Signal Processing.* Wiley-Interscience, 1st edition, 2008.

[49] NEC Corporation. NEC Develops World-Leading Technology to Prevent IP Phone SPAM. Product News, http://www.nec.co.jp/press/en/0701/2602.html, 2007.

[50] S. Niccolini, S. Tartarelli, M. Stiemerling, and S. Srivastava. SIP Extensions for SPIT identification. draft-niccolini-sipping-feedback-spit-03, IETF Network Working Group, Work in Progress, 2007.

[51] NuVox Communications. Voice and Data Service Provider. Website, http://www.nuvox.com, 2009.

[52] OPNET. Optimum Network Performance, Modeler Tool Version 14.5. Network Simulation Tool. 2007.

[53] Kenneth G. Paterson. Id-based signatures from pairings on elliptic curves. In *Electronics Letters 38, no. 18*, pages 47–53, 2002.

[54] Colin Perkins. *RTP: Audio and Video for the Internet.* Addison-Wesley, 1st edition, 2012.

[55] Jon Peterson, Henning Schulzrinne, and Hannes Tschofenig. Secure origin identification: Problem statement, requirements, and roadmap. 2013.

[56] Zbigniew Piotrowski and Piotr Gajewski. Voice spoofing as an impersonation attack and the way of protection. *Journal of Information Assurance and Security*, 2(3):223–225, 2007.

[57] Bradley Reaves, Logan Blue, and Patrick Traynor. Authloop: End-to-end cryptographic authentication for telephony over voice channels. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, Thorsten Holz and Stefan Savage, editors, pages 963–978. USENIX Association, 2016.

[58] Y Rebahi, D Sisalem, J Kuthan, A Pelinescu-Oncicul, B Iancu, J Janak, and DC Mierla. The sip express router-an open source sip platform. In *Evolute Workshop, Guildford, UK*, 2003.

[59] Yacine Rebahi and Adel Al-Hezmi. Spam Prevention for Voice over IP. Technical report, http://colleges.ksu.edu.sa/ComputerSciences/Documents/NITS/ID143.pdf, 2007.

[60] P Resnick. Internet message format. RFC 2822, IETF (Standards Track) Request for Comments, 2001.

[61] J. Rosenberg and C. Jennings. The Session Initiation Protocol (SIP) and Spam. RFC 5039, IETF Network Working Group, 2008.

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

[63] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. Sok: Fraud in telephony networks. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 235–250. IEEE, 2017.

[64] Henning Schulzrinne. The tel uri for telephone numbers. RFC 2806, IETF Network Working Group, 2004.

[65] Hemant Sengar. *Beware of New and Readymade Army of Legal Bots.* http://www.usenix.org/publications/login/2007-10/, October 2007.

[66] Hemant Sengar. Voice Spam (SPIT) Problem. Website, http://www.vodasec.com/, March 2007.

[67] Hemant Sengar, Xinyuan Wang, and Art Nichols. Call Behavioral Analysis to Thwart SPIT Attacks on VoIP Networks. In *SecureComm*, 2011.

[68] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in cryptology*, pages 47–53. Springer, 1985.

[69] SIPERA. *Sipera IPCS: Products to Address VoIP Vulnerabilities.* http://www.sipera.com/index.php?action=products,default, April 2007.

[70] JaeSeung Song and Andreas Kunz. Towards standardized prevention of unsolicited communications and phishing attacks. *Journal of ICT Standardization*, 1:109–122, 2013.

[71] T. Granlund. The GNU Multiple Precision Arithmetic Library. Website, https://gmplib.org, 2000.

[72] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. Sok: Everyone hates robocalls: A survey of techniques against telephone spam. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 320–338. IEEE Computer Society, 2016.

[73] VOIPSA. Confirmed cases of SPIT. Mailing list, http://www.voipsa.org/pipermail/voipsec_voipsa.org/2006-March/001326.html, 2006.

[74] VOIPSA. Confirmed cases of SPIT. Mailing list, http://www.voipsa.org/pipermail/voipsec_voipsa.org/2006-March/001326.html, 2006.

[75] VOIPSA. VoIP Attacks in the News. Website, http://voipsa.org/blog/category/voip-attacks-in-the-news/, 2007.

[76] VOIPSA. Voip attacks in the news, 2007.

[77] Wikipedia. Plaintalk. Website, http://en.wikipedia.org/wiki/PlainTalk.

[78] Wikipedia. Turing test. Website, http://en.wikipedia.org/wiki/Turing_test, 2009.

[79] Yu-Sung Wu, Saurabh Bagchi, Navjot Singh, and Ratsameetip Wita. Spam Detection in Voice-Over-IP Calls through Semi-Supervised Clustering. In *IEEE Dependable Systems and Networks Conference (DSN 2009)*, June-July 2009.

[80] Yate. Yate: The Next Generation Telephony Engine, 2013.

[81] Fangguo Zhang and Kwangjo Kim. Id-based blind signature and ring signature from pairings. *Advances in cryptologyASIACRYPT 2002*, pages 533–547, 2002.

[82] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. *Public Key CryptographyPKC 2004*, pages 277–290, 2004.

[83] Hans Jürgen Zimmermann. *Fuzzy set theory-and its applications*. Springer, 2001.