NAIST-IS-DD1161030

Doctoral Dissertation

Protecting IP Telephony against SPIT and SIP Flooding Attacks

Noppawat Chaisamran

February 6, 2014

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Noppawat Chaisamran

Thesis Committee:

Professor Suguru Yamaguchi	(Supervisor)
Professor Minoru Ito	(Co-supervisor)
Associate Professor Youki Kadobayashi	(Co-supervisor)
Assistant Professor Vasaka Visoottiviseth	(Mahidol University)

Protecting IP Telephony against SPIT and SIP Flooding Attacks[*]

Noppawat Chaisamran

Abstract

The global communication market is rapidly moving toward IP (Internet Protocol) telephony. Similar to other IP-based applications, it is vulnerable to several attacks. Therefore, security concerns become more important for users and service providers. In this dissertation, real-time attack detection systems are proposed to protect the IP telephony against Spam over Internet Telephony (SPIT) and Session Initiation Protocol (SIP) flooding attacks. It consists of three main contributions.

First, a trust-based SPIT detection based on calling behavior and human relationships is introduced to classify calls. A call duration and its direction as well as a calling ratio of each user are used to calculate a trust value. This trust value is automatically adjustable according to the call characteristics in order to keep track of a current user's behavior and avoid bias in trust value assignment.

Second, an anomaly-based SIP flooding attack detection system is proposed to detect a significant deviation in SIP traffic. Three statistical algorithms are used to analyze incoming traffic to a server: an application of Tanimoto Distance, an adaptive threshold, and a Momentum Oscillation Indicator. Due to a stateless and low computational cost of these algorithms, the proposed system can classify traffic in nearly real-time that is suitable for an IP telephony system.

Lastly, false positive alarms of the flooding attack detection are reduced by using a trust filtering. A reliable trust value is calculated through the call activities and the human behavior of each user. The trust value of suspicious callers will be checked before raising any alarm.

[*] Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1161030, February 6, 2014.

The comprehensive synthetic datasets containing various malicious traffic patterns are used to validate the effectiveness of the proposed system. The results showed that it accurately identified attacks and has the flexibility to deal with many types of attack patterns with a low false positive rate.

Keywords:

VoIP, IMS, SPIT, DoS, Statistical Analysis, Trust

Acknowledgments

This dissertation would not have been possible without the support of many people.

Foremost, I am heartily thankful to my supervisor, Prof. Suguru Yamaguchi, who was abundantly helpful and offered invaluable assistance, support, and motivation. His guidance helped me in all the time of research and writing of this dissertation.

I also would like to express my deepest gratitude to my thesis committee, Prof. Minoru Ito, for the useful comments I received that greatly help to improve the overall quality of this dissertation.

I would like to thank Assoc. Prof. Youki Kadobayashi for his endless support of my work and ideas. His vision greatly impacted my work and his invaluable comments helped me to improve on my paper quality.

My sincere thanks also goes to my lovely teacher and thesis committee, Asst. Prof. Vasaka Visoottiviseth, for her kindness and support.

I cannot find words to express my gratitude to Assoc. Prof. Takeshi Okuda for his invaluable assistance and helpful advice. Without his knowledge and assistance, this study would not have been successful.

My special thanks goes to Christopher Michael Yap and Jane Louie Fresco Zamora, who always help me polishing my writing.

I wish to thank Ministry of Education, Culture, Sports, Science and Technology (MEXT) for granting me the greatest scholarship that allowed me to finish my graduate studies.

I owe my deepest gratitude to my family and my girl friend who were always there for me and cheer up every time.

Last but not least, I would like to thank all members of IPLab and Thai students in NAIST for their friendship and support.

To my parents,
for their love, endless support and encouragement

Contents

List of Figures

List of Tables

Acronyms

AS	Application Server.
CSCF	Call Session Control Function.
CUSUM	Cumulative Sum.
DoS	Denial-of-Service.
DST	Dempster-Shafer Theory.
EMA	Exponential Moving Average.
FN	False Negative.
FP	False Positive.
FPR	False Positive Rate.
HD	Hellinger Distance.
HSS	Home Subscriber Server.
I-CSCF	Interrogating Call Session Control Function.
IDS	Intrusion Detection System.
IETF	Internet Engineering Task Force.
IMS	IP Multimedia Subsystem.
IP	Internet Protocol.
MOI	Momentum Oscillation Indicator.

NGN	Next-Generation Network.
P-CSCF	Proxy Call Session Control Function.
S-CSCF	Serving Call Session Control Function.
SIP	Session Initiation Protocol.
SPIT	Spam over Internet Telephony.
SR	Social Reliability.
TD	Tanimoto Distance.
TN	True Negative.
TP	True Positive.
VoIP	Voice over Internet Protocol.

Chapter 1

Introduction

IP Telephony has been growing considerably in the past decade. Since it is running on the Internet Protocol (IP) network, it has many security challenges that do not exist in the traditional telephony system. This has lead to a demand for automated systems for detecting malicious activity in IP telephony networks.

1.1. Motivation

Since the invention of the telephone, real-time communication networks have been built using closed circuit-switched network infrastructures, e.g., the Public Switched Telephone Network (PSTN). With the advent and the increasing popularity of the packet-switched Internet data network, operators are seeking ways to combine both communication and data networks on IP network basis. Voice over Internet Protocol (VoIP) is a first step into this direction. It is the technology used to establish telephone calls and other multimedia streams over the IP networks. In a further step, International Organization for Standardization have coined the term Next Generation Networks (NGN) to define a standardized way to encapsulate telecommunication services in an IP network. The 3rd Generation Partnership Project (3GPP) has standardized a NGN called IP Multimedia Subsystem (IMS), which is a common architectural component of further NGNs. IMS is a global, access-independent, and standard-based IP connectivity and service control architecture that provides various types of multimedia services to end-users using common Internet-based protocols [1]. Telecommunication providers are changing their basic infrastructure to use VoIP and IMS as the core for their future network.

The key protocol for regular VoIP and IMS services is the Session Initiation Protocol (SIP), which is the control protocol for multimedia communication. Unlike the closed PSTN architecture, SIP networks are deployed on the open IP stack and thus, vulnerable to many of the same security threats. VoIP Security Alliance (VoIPSA) defines the security threats against VoIP deployments, services, and end users as follows [2]:

1. Social Threats
   This category includes misrepresentation, theft of service, and unwanted contact or Spam over Internet Telephony (SPIT).

2. Eavesdropping
   Eavesdropping attacks describe a method by which an attacker is able to monitor the entire signaling and/or data stream between two or more VoIP endpoints, but cannot or does not alter the data itself.

3. Interception and Modification
   These class of attacks describe a method by which an attacker can see the entire signaling and data stream between two endpoints, and can also modify the traffic as an intermediary in the conversation.

4. Intentional Interruption of Service
   Threats in this category aim at interrupting users from using VoIP services. Intentional interruption can be carried out in many ways. Denial of Service (DoS) threats, especially VoIP specific ones, count for the largest part of intentional interruption threats.

5. Service Abuse
   Service abuse covers threats regarding any kind of fraudulent activity over VoIP, e.g., premium rate service fraud, improper bypass, and adjustment to billing.

6. Other Interruptions of Service
   This category hosts other threats that interrupt VoIP service but are not

necessarily intentional, e.g., loss of power, resource exhaustion, and performance latency.

In this work, I focus and propose a security system to protect a VoIP network against SPIT and DoS attacks. Next, I will describe why these two threats are a serious problem in VoIP network.

VoIP services have gained popularity due to largely reduced cost and wider range of advanced services, as compared to traditional telephone networks. Thus, Spam over Internet Telephony (SPIT), known as unsolicited advertising calls sent via VoIP networks, is becoming a major problem that would undermine the usability of VoIP. SPIT is a much more serious problem than email spam because the callee is directly denied service by the incoming call. By way of example, a spam email arriving at an inbox at 2 a.m. will not disturb the user, but a ringing phone at 2 a.m. most likely will. Furthermore, spam calls waste a large amount of network bandwidth and would cause the delays for other network traffic. Filtering voice spam is more difficult than filtering email. To filter out email spam, we can use adaptive heuristic filters that look for text patterns. But voice calls take place in real-time and it is a bit pointless to filter a call after picking up the receiver. This places strong limitations on what operations can actually be performed on incoming voice packets in terms of the speed and complexity of the analysis. Therefore, SPIT detection is one of the greatest challenges for future large-scale deployments of VoIP telephony.

Denial-of-Service (DoS) attacks have been a major threat in the Internet world for many years. These attacks aim at denying a legitimate user's access to a service or network resource, or at bringing down the servers offering such services. Like any IP-based protocol, SIP is also vulnerable to this attack. Hence, VoIP and IMS are much more susceptible to DoS attacks, compared with any previous telecommunication infrastructure. An attacker can easily launch a DoS attack by flooding SIP servers with an enormous number of SIP messages. This service availability disruption is one of the major QoS threats. The US National Institute of Standards and Technology (NIST) determined DoS flooding to be a serious threat for SIP VoIP infrastructures [3]. In a threat analysis for ETSI TISPAN networks, DoS attacks on publicly available interfaces were considered a critical risk [4]. According to the 3GPP technical specifications [5, 6], IMS security offers features such as authentication and encryption, but it does not provide any mechanism to protect IMS networks from flooding attacks. As the world's telecom operators gradually deploy IMS to real world networks, SIP flooding attacks will become serious security threats to the telecom operators' businesses. Therefore, DoS attacks detection is becoming a necessity. There are several types of DoS attacks on SIP network. First, common IP network and transport layer DoS attacks are also available for SIP networks. These attacks have been known about for years and have been excessively studied in literature. Furthermore, there have been new attacks that have directly targeted the SIP application layer itself. These attacks include SIP message flooding, SIP message payload or SIP flow tampering attacks. This work focuses only on SIP-related flooding attacks. To counter such attacks, new algorithms tailored directly for SIP environments have to be deployed.

Last but not least, a basic concern for any flooding detection system is to identify changes in the traffic volume. Typically, such systems raise alarms when the traffic rate is higher than a threshold. A main drawback of this approach is that the detection accuracy may be degraded if the legitimate volume is dynamic or suddenly increased. It happens easily and frequently in telecommunications. For example, a call rate increases suddenly during a natural disaster or in a big important event. This causes a false positive in any flooding detection system that occurs when the detection fails to consider the legitimate sampled traffic as an attack. This false alarm is one of the most important problems because it causes a loss of confidence in the alerts of a security framework. In some systems, the task of filtering and analyzing alerts is done manually, but that constitutes an overload of work for any security administrator. Therefore, we need a way to confirm that a real attack is taking place before raising any alert. Many methods have been proposed in order to produce a more qualitative alert set. Some of them propose different configurations of detections, while most of them propose the post processing of alerts. Usually, with anomaly-based detection, the abnormality is determined by measuring the distance between the suspicious activities and the norm. Then, based on a chosen threshold, the observed behaviour is classified. Increasing this threshold leads directly to induce more false alarms, while many of them are actually not true. Reducing the threshold can reduce the number of false alarms but such an action causes the detection to be unable to detect major attacks. This is the trade-off between reducing false alarms and maintaining system security.

1.2. Contributions

In this work, I propose two security systems that aim at protecting IP telephony against SPIT and DoS attacks.

I present a trust model, based on call duration and direction, to distinguish between legitimate users and spammers in real-time. From the observation, the call duration of a spammer is significantly shorter than that of a legitimate user, and a spammer will typically receive few or no calls from other users. This pattern is used to calculate a trust value for each individual user. One can assign trust values to friends by calling them. A trust value is calculated by comparing one's call duration with the average call duration among friends. Long call duration indicates high trust. The following scenario shows how a trust value can be constructed. Assume that Alice makes a call to Bob and Carol lasting, 5 and 15 minutes, respectively. After comparing with the average value, the trust value of Carol will be higher than Bob's because the call duration between Alice (trustor) and Carol (trustee) is longer than with Bob. Unlike other trust-based detection systems, trust value in this work is automatically assigned to each user and adjusted by human calling behavior. It avoids the biasing problems that occur when one legitimate user incorrectly rates another legitimate user as a spammer.

In case of an unknown caller, trust values inferred from other users in the callee's community are used to calculate a trust value for this caller. I use three theories to propagate a trust value among nodes in the VoIP network. First, from the definition of trust [7], trust can be inferred from one node to another node via a social relationship. For example, if Alice knows Bob, and Bob knows Dave, then Alice can use the relationship path to infer a trust rating for Dave. Second, I use the concept of seven degrees of separation to limit the relationship links when propagating a trust value. This small world hypothesis refers to the idea that everyone is, on average, approximately seven steps away from any other person [8]. This is different from other trust-based systems because in other trust-based systems, a trust value is inferred at most two hops away. This may cause an increase in false alarms. Third, I use the data fusion technique, called the Dempster-Shafer Theory, to aggregate all trust paths and then compute an inferred trust value for this caller. In addition, the social reliability, which is the evaluation of a user's behavior up to now, is considered before forwarding or dropping a call.

The detection system at the operator side calculates a trust value and social reliability value of each caller before establishing a call to a callee. If these values are lower than the predefined thresholds, the call is rejected. This method aims to detect SPIT calls before call establishment while requiring the least possible interaction with the caller and the callee. Moreover, this system does not require changing the existing VoIP protocols infrastructure. The VoIP operator can apply this system directly to any VoIP technology.

For detecting DoS attacks, I propose a multivariate statistical-based flooding attack detection system that generates alerts based on abnormal variation in traffic flows. The SIP traffic activity is captured and a profile representing its stochastic behaviour is created. This profile is based on the number of five SIP packets types, `REGISTER`, `INVITE`, `200 OK`, `ACK`, and `BYE`. The probability dissimilarity will be computed between an incoming traffic and the current profile by a Tanimoto Distance (TD) algorithm. This algorithm is useful for quantifying the correlations among chosen attributes. Moreover, it is a simple computation method and able to adapt to traffic changes. Therefore, it fits to the dynamic environment, especially the IMS networks. Next, I propose an adaptive threshold that is used for detecting a significant deviation of traffic. The adaptive threshold can exhibit good performance to keep track legitimate traffic, but it generally suffers from a special attack pattern: an attacker can completely hide an attack by gradually increasing flooding packets. Detection of this attack is particularly important. Therefore, I propose a momentum oscillation indicator to detect such changes in the traffic.

A main drawback of detecting changes in the traffic volume is that the detection accuracy may be degraded if the legitimate volume is suddenly increasing. This phenomenon happens easily and frequently in telecommunications. I address this problem by integrating a trust model to filter out a legitimate call from suspicious traffic. The trust value of each user is computed from the call activities and human behavior of a user, including call duration, call direction, an interactivity ratio, and the diversity of calls. Trust value of suspicious callers will be evaluate before raising any alarm.

This detection system is placed in front of a SIP proxy in order to monitor incoming SIP traffic. In case of an IMS system, the detection is located at the front of a Proxy Call Session Control Function (P-CSCF) server. Since P-CSCF is the first core component to be traversed for any request process, deployment of a detection system here will be very helpful in mitigating DoS attacks against IMS networks. Moreover, because this proposal is a stateless approach, the detection system does not require huge memory capacity to process incoming packets. In fact, this can avoid a bottleneck problem when a massive traffic comes to the server.

Figure 1.1 shows the overview of the whole detection system. To summarize, my proposed IP telephony protection framework has the following security features:

SPIT detection It can detect social threats like SPIT in real-time during the signaling process by using a trust value of each caller.

Detection of SIP flooding attacks It can accurately detect the attacks with a small delay.

Robust mechanism of raising alarm With my false reduction method, it raises an alarm with high accuracy.

The proposed systems are evaluated on a realistic simulation. The experimental results demonstrate that these systems can achieve a high degree of accuracy in detecting attacks, SPIT and DoS, with low false positives. Without an infrastructure modification, these system can be deployed into any VoIP and IMS network infrastructures to provide defense against such attacks.

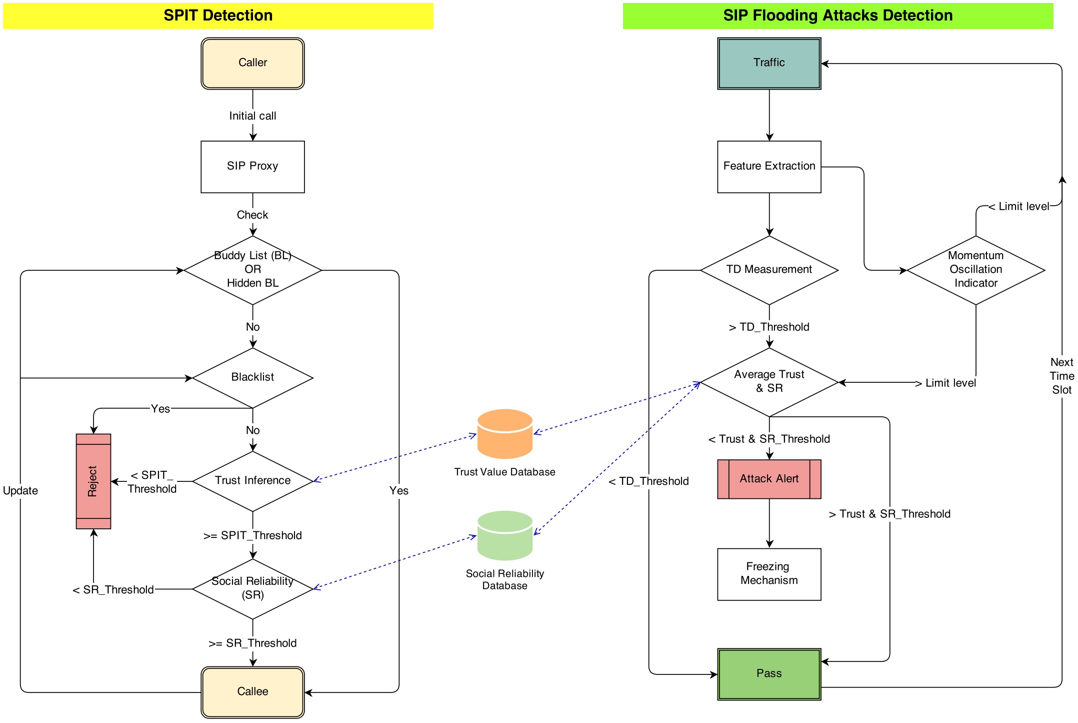

Figure 1.1. Overview of the detection architecture.

1.3. Dissertation Structure

The remainder of this dissertation is organized as follows. Chapter 2 contains an overview of VoIP and IMS technologies. I focus on these two systems because they are well-represented and widely-used. Two severe security threats, SPIT and DoS, are also described in this chapter.

Chapter 3 presents the trust-based SPIT detection mechanism.

Chapter 4 presents the proposed flooding attack detection system based on the statistical analysis. Three statistical formulas are applied to monitor SIP traffic: Tanimoto distance, adaptive threshold, and momentum oscillation indicator. False positive reduction by trust model is also described in this chapter. The trust value of each caller is estimated to filter out a legitimate call from a suspicious traffic.

Chapter 5 summarizes the contributions of this dissertation. In addition, the future challenges are also discussed in this chapter.

Chapter 2

IP Telephony and Security Threats

The telecommunications industry is undergoing a fundamental change and the catalysts for this change are the business models and technologies of the Internet. The ubiquitous use of the Internet Protocol (IP) suite for voice, data, media and entertainment purposes, is driving the convergence of industries, services, networks and business models. IP provides a common foundation by offering end-users seamless access to any service, any time, anywhere, and with any device. Voice over Internet Protocol (VoIP) is the first step to delivering voice data over IP network. In the next development, Next-Generation Network (NGN) provides the infrastructure for multimedia converging networks, called IP Multimedia Subsystem (IMS). And now it is being standardized for worldwide use. Since it is defined by Internet Engineering Task Force (IETF), the main building block of this technology is Session Initiation Protocol (SIP).

This chapter provides an overview of SIP-based VoIP network and Next-Generation Networks. I focus on SIP because of its popularity and wide use in telecommunications today. Besides SIP, there are many different alternatives for voice session handling over IP networks. The most common standard from International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) is H.323. It is implemented in various devices for voice and video conferencing. However, due to the popularity of SIP, H.323 support is declining. Skype [9] is a proprietary protocol used in its network. It provides voice, chat, and video service over an encrypted protocol between two ends. Skype's popularity was gained from its ease of use and simple configuration. The protocol specification is disclosed, hence, several attempts have been conducted to reverse-engineer the protocol.

The last two sections describe IP telephony security threats, Spam over Internet Telephony (SPIT) and Denial-of-Service (DoS), which are focused in this work.

2.1. Multimedia Communication Using SIP

VoIP defines a way to carry voice calls over an IP network including the digitization and packetization of the voice streams. VoIP has been implemented in various ways using both proprietary and open protocols and standards. However, one widely used standard nowadays is SIP.

SIP is a text-based protocol that has been standardised by the IETF, RFC3261 [10]. It is an application-layer control protocol that is widely used for controlling multimedia communication sessions such as voice and video calls over IP. It allows users to create, modify, and terminate sessions with one or more participants. It can also be used to create two-party, multi-party, or multicast sessions that include Internet telephone calls, multimedia distribution, and multimedia conferences.

The SIP protocol is an application layer protocol designed to be independent of the underlying transport layer; it can run on Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Stream Control Transmission Protocol (SCTP). It employs design elements similar to the Hypertext Transfer Protocol (HTTP) request/response transaction model and reuses the header field, encoding rules, and status codes of HTTP, e.g., From: `user@sip.org` to denote the sender of a message. The SIP request methods and response codes are described in Table 2.1 and 2.2, respectively.

SIP requests and responses contain headers following the request or status lines. Those headers are used to transport the information to the SIP entities, some of which are specific to requests and some of which to responses. A header is composed of the header name, followed by a colon and a header value. The

Table 2.1. SIP request methods.

Method	*Description*
INVITE	Initiates a call
ACK	Confirms a final response for INVITE
BYE	Terminates a call
CANCEL	Cancels searches and ringing
OPTIONS	Queries the capabilities of the other side
REGISTER	Registers with the location service

Table 2.2. SIP response code.

Class	*Description*
1xx	Provisional messages that is used by the server to indicate progress
2xx	Successful answers
3xx	Redirection, forwarding messages
4xx	Request failure (client mistakes)
5xx	Server failures
6xx	Global failures (busy, refusal, not available anywhere)

main header fields are shown below.

- **To:** Denotes the receiver of this SIP messages. This is generally the publicly available address of the user.
- **From:** Denotes the sender of the message.
- **Contact:** The actual location where a user can be reached. This location can be different from the From URI.
- **Cseq:** An integer and a method name, where the integer part of this header is used to detect the non-delivery of the message or out-of-order delivery messages. At the beginning of a transaction, it is randomized and upon arrival of a new message, it is incremented by one.
- **Record-Route:** Indicates that an intermediate proxy wants to receive further signalling traffic.
- **Route:** Indicates a route that a new request is going to take.
- **Via:** A list of all intermediate SIP entities that these messages have passed so far.

SIP communication network as defined by RFC3261 is at least composed of four general types of logical SIP entities.

User Agents A user agent (UA) is the endpoint entity. User agents initiate and terminate sessions by exchanging requests and responses. RFC3261 defines the user agent as an application which contains both an user agent client (UAC) and user agent server (UAS). UAC is a logical component generating SIP request messages. UAS is a logical component generating response messages corresponding to SIP request message sent by UAC.

Registrar Server A registrar server accepts requests from a user who wants to make himself available to the network. It processes its registration information and stores it into a location service for further reference. It gives UAC specific information about UAS's connection address.

Redirect Server A redirect server receives SIP requests and responds with redirection responses, thus directing the client to contact an alternate set of SIP addresses.

Proxy Server SIP proxies perform general routing operation in the network by forwarding SIP requests to UAS and SIP responses to UAC. The SIP standard allows proxies to perform actions such as request validation, user authentication, request forking, address resolving, or to cancel pending sessions.

SIP's purpose is to manage sessions between users. These include the management of user location, availability and capabilities as well as session setup, handling, and termination. Figure 2.1 shows the call set-up between two UAs via two SIP proxy servers that act on behalf of Alice and Bob to facilitate the session establishment.

1. When Alice (`alice@mahidol.ac.th`) calls Bob (`bob@naist.jp`), the `INVITE` request message is sent to the proxy server responsible for the `mahidol.ac.th` domain.
2. The proxy server immediately responds with a `100 Trying` provisional response.

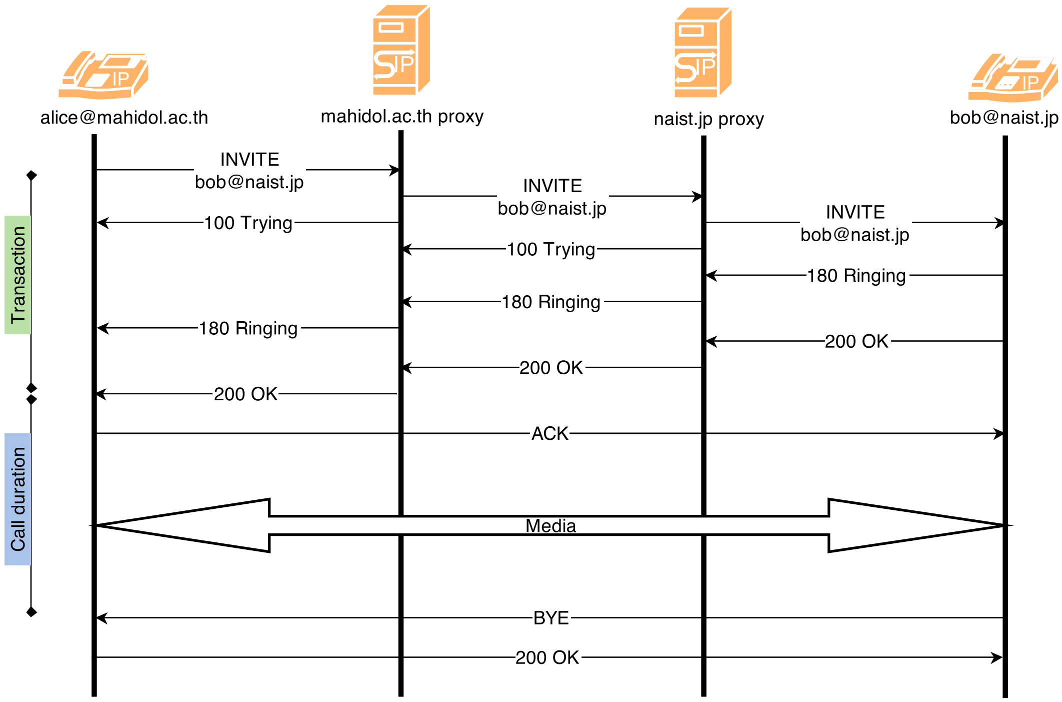

Figure 2.1. SIP session setup example.

3. Then `mahidol.ac.th` proxy determines how to route the call to the proxy responsible for Bob's domain, `naist.jp`.
4. Once `naist.jp` receives the request, it looks up user Bob and then routes it to the appropriate endpoint.
5. At the `INVITE` message recipient, Bob's UA starts to ring.
6. The UASs respond with a `180 Ringing` responses.
7. When Bob picks up the phone, the UA sends a `200OK` response. This initial message exchange forms the call setup transaction.
8. When Alice or Bob hangs up, the respective UA sends a `BYE` message and this initiates the call tear-down transaction.

SIP's target is session control and not the transmission of actual session content. Description and transport of content are managed by other protocols, which are running in conjunction with SIP.

2.1.1 SIP Security Mechanism

According to the SIP specification, it does not specify its own security mechanisms. Instead, it utilized other well-known IP-based security techniques. Four security mechanisms, that are defined in RFC 3261, are described briefly in this section.

HTTP Authentication

SIP provides a stateless, challenge-based mechanism for a user authentication that is based on authentication in HTTP. Any time that a proxy server or UA receives a request, it may challenge the initiator of the request to provide assurance of its identity. Authentication requests from a proxy server and a registrar server are slightly different. A proxy generates a `407 Proxy Authentication Required` response with an additional Proxy-Authenticate header. A registrar sends a `401 Authentication Required` response with a WWW-Authenticate header. A SIP server manages information of users that is in the form of user name and password.

Transport Layer Security (TLS) & IP Security (IPsec)

SIP messages can be encrypted hop-by-hop by using the TLS. This cryptographic protocol is widely adopted for securing a web traffic. It runs only above TCP/IP and below higher-level protocols, e.g., SIP. Then, increasing up TCP connections will cause additional load on SIP servers. SIP provides a notation to request a secure connection with the SIPS URI, e.g., `sips:bob@naist.jp`.

Alternative to TLS, IPsec may be used to encrypt SIP messages. IPsec is a protocol suite that provides a set of services to protect IP packets from attacks. It can provide confidentiality, integrity, data origin authentication services as well as traffic analysis protection.

Secure MINE (S/MIME)

Encrypting entire SIP messages end-to-end for the purpose of confidentiality is not appropriate because network intermediaries (like proxy servers) need to view certain header fields in order to route messages correctly. If these intermediaries are excluded from security associations, then SIP messages will essentially be non-routable. However, S/MIME allows SIP UAs to encrypt MIME bodies within SIP, securing these bodies end-to-end without affecting message headers. S/MIME can provide end-to-end confidentiality and integrity for message bodies as well as mutual authentication. It is also possible to use S/MIME to provide a form of integrity and confidentiality for SIP header fields through SIP message tunneling.

2.2. IP Multimedia Subsystem (IMS)

Recently, the convergence between fixed and mobile communications has been trending and is called Fixed-mobile convergence (FMC). The idea is to deliver the Internet, mobile cellular networks, and PSTN networks on a single platform. IMS is a standardized NGN for such converging communication networks that has been widely used nowadays.

IMS is an overlay service provisioning platform through which telecommunications operators can utilise Internet technologies to their greatest advantage. It operates across fixed and mobile access technologies including WLAN, UMTS/HSPA, and DSL, along with many others. The telecommunications industry has high expectations for IMS [1]. The key features of IMS are multimedia session management, guaranteed Quality-of-Service (QoS), secure network access and service control.

IMS is based on IETF core protocols that uses SIP for session control. Other important protocols are Diameter for AAA (Authentication, Authorization, and Accounting) service and RTP (Real-time Transport Protocol) for transmitting media.

IMS consists of three main components: the Call Session Control Function (CSCF), Home Subscriber Server (HSS), and Application Server (AS). CSCF is the primary SIP signaling server that acts as the SIP rendezvous point. The CSCF duties are divided into three categories.

- Proxy Call Session Control Function (P-CSCF): The P-CSCF is the first contact point for users within the IMS. It controls incoming and outgoing messages between the IMS and end users.

- Serving Call Session Control Function (S-CSCF): The S-CSCF is the focal point of the IMS as it is responsible for handling registration processes, making routing decisions, maintaining session states and storing service profiles.

- Interrogating Call Session Control Function (I-CSCF): The I-CSCF provides the external interface to other IMS networks and plays an important role in both inter-carrier calls and roaming.

The HSS is the main data storage for all subscriber and service-related data. It provides a database of user credentials and configurations and identifies the home S-CSCF of the subscribers. Finally, the AS hosts and executes services. An example of AS is the Voice Call Continuity Function (VCC server) that guarantees a call persistent when a mobile phone moves between base stations. Many AS can be installed in an IMS network as necessary to support the users. Figure 2.2 and 2.3 show the registration and basic invitation procedures of an IMS, respectively. The high-level requirements on IMS are summarized in the technical specification 23.228 [11].

2.3. Spam over Internet Telephony (SPIT)

Spam, defined as the transmission of bulk unsolicited messages, has plagued Internet email. Unfortunately, it is not only limited to email, but also affects any system that enables user-to-user communications. Since SIP is used for multimedia communications between users, it is susceptible to spam, just as email is.

VoIP spam may occur in different forms and may exploit different weaknesses of VoIP technologies. According to RFC5039 [12], a spam on SIP protocol can be classified into 3 types:

1. Call Spam: This type of spam is defined as a bulk unsolicited calling that attempts to establish a voice communication session. If the target callee answers a call, the spammer proceeds to relay his message over the real-time media. This is the classic telemarketer spam and often called SPam over Internet Telephony, or SPIT.

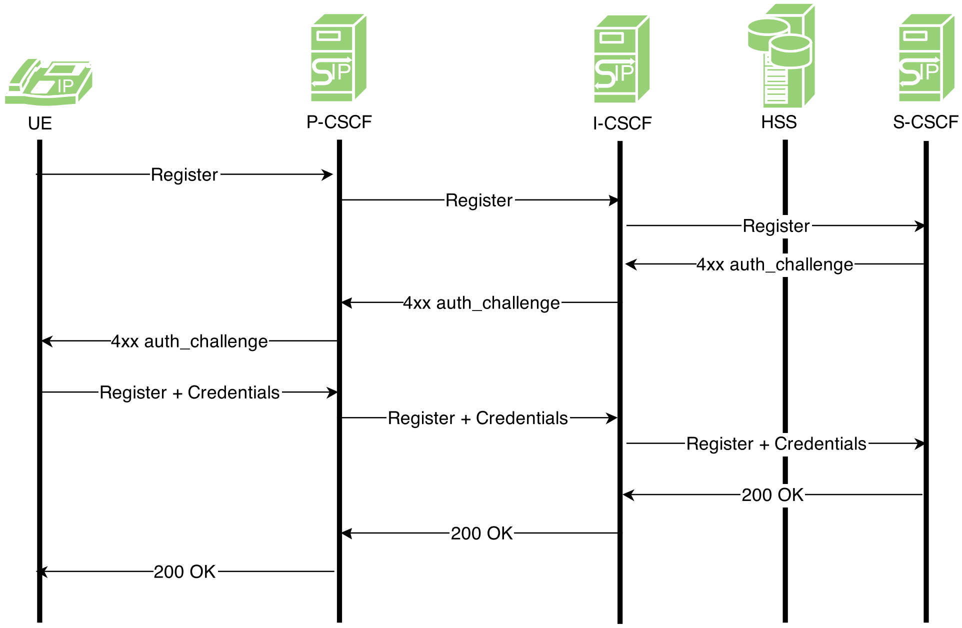

Figure 2.2. IMS registration procedure.

2. IM Spam: This type of spam is similar to email. It is an unsolicited instant message sent via the "Subject" field in a request message. This is often called SPam over Instant Messaging, or SPIM.

3. Presence Spam: This type of spam is similar to IM spam. A spammer sends bulk unsolicited set of presence requests to become a member of the white list or buddy list of a user in order to send them IM or initiate other forms of communications. This is occasionally called SPam over Presence Protocol, or SPPP.

In this dissertation, I consider only a call spam or SPIT. SPIT is a prerecorded unsolicited advertising message sent to your handset. It is a form of spam since the phone ring disrupts the work flow when at work, can wake one up at night when sleeping, and can be annoying when doing anything else. SPIT is becoming an attractive and cost-efficient marketing strategy because VoIP allows cheap calls compared with traditional PSTN networks. As we know, the traditional telephony call spam already exists in the form of telemarketer calls. But its volume is not as much as email spam because of the cost. However, the cost is

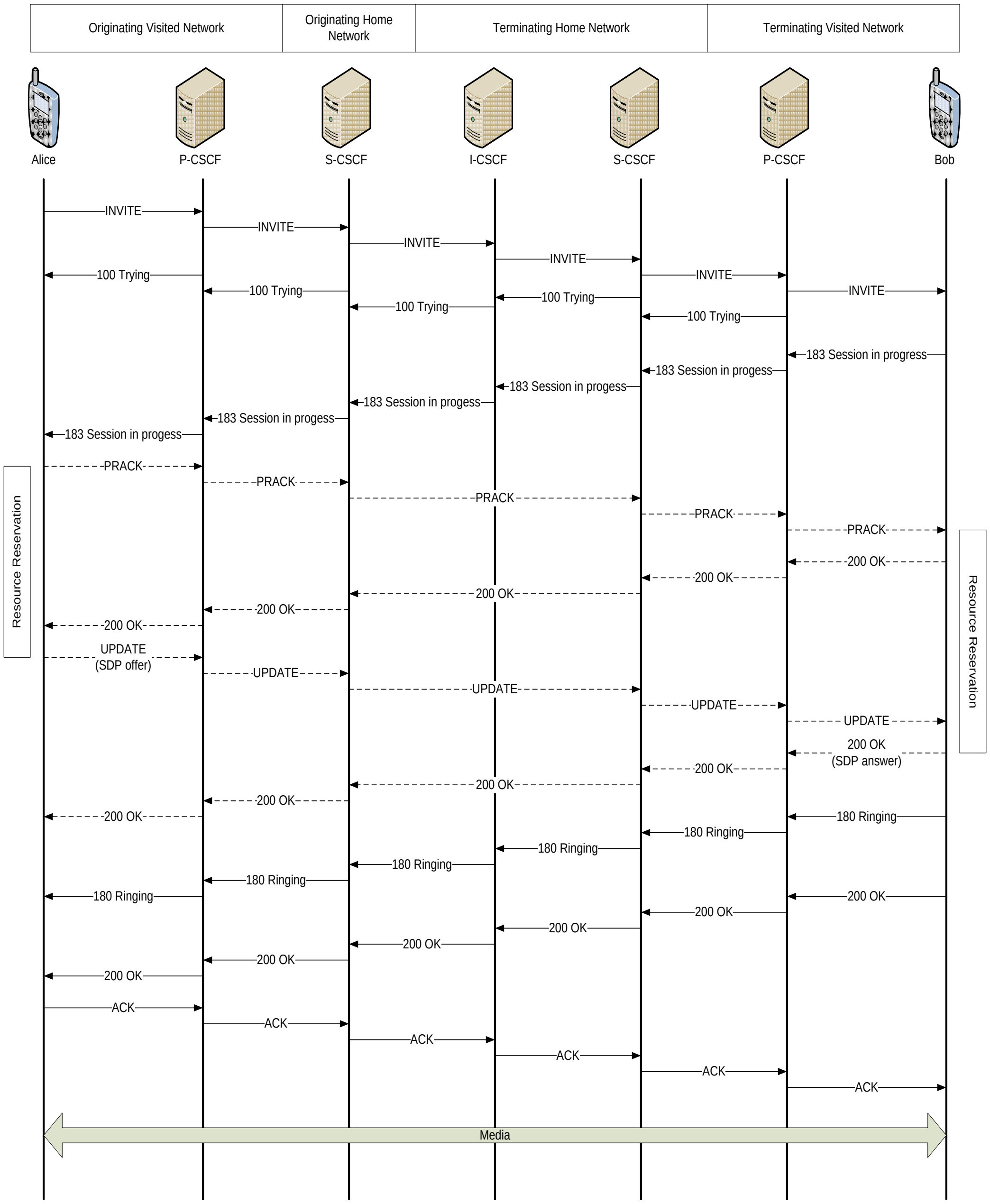

Figure 2.3. Basic IMS invitation procedure.

dramatically lower when switching to SPIT for many reasons: low call fee, low hardware cost, no boundary of international calls, etc. Launching an international spam campaign using PSTN can be rather expensive. With VoIP, however, a spammer can subscribe to a flat-rate service abroad at costs similar to a national campaign. Using IP-based technology enables companies to use automatic dialer programs for delivering thousands of unsolicited messages every minute at an incredibly lower cost. Compared with email, using voice calls offers telemarketers a wider range of use scenarios:

- Passive marketing - a prerecorded voice message presents the sales pitch. Once a recipient accepts a call, the system delivers the content as a media stream.
- Interactive marketing - these are the standard telemarketing calls in which a live caller tries to sell products or services to a callee.
- Call back - this is a fraud method. The fraudster makes a call but hangs up before the callee answers. Out of curiosity, the callee returns the call, unaware that it is a premium phone number, and incurs a hefty charge.

Unlike detection and filtering of email spam, countermeasures against SPIT face great challenges on how to identify and filter SPIT in real time. The main difference between email spam and SPIT is that an email arrives at the email server before it is accessed by the user. Hence, a structure and content of an email can be analyzed and detected at the server before reaching the recipient. But SPIT is a legitimate call request for a SIP server that forwards it to the called person. The session establishment messages are forwarded immediately to the recipients. Once the recipients picks up the telephone, only then they realize that the calls are spam. Many techniques devised for email spam filtering rely upon content analysis, but in the case of VoIP analysis, after picking up the phone is too late. Any delay due to anti-SPIT processing would degrade the quality of service.

2.3.1 SPIT Process

Basically, there are three steps of a SPIT initiation.

1. **Information Gathering**

   In order to contact a victim, a SPITer must know the SIP URI of the

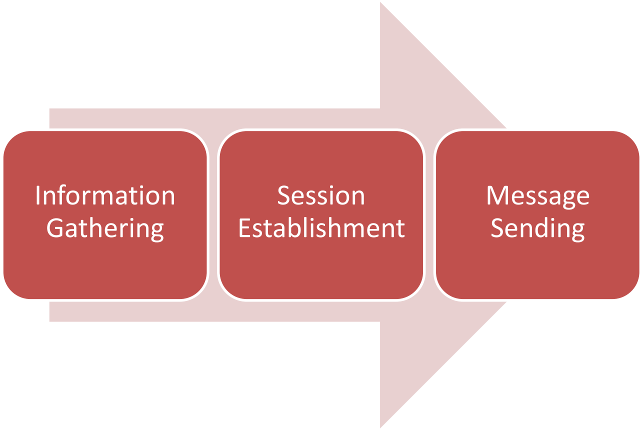

Figure 2.4. SPIT process.

victim. If he wants to reach as many victims as possible, he must catalogue valid assigned SIP URIs. The easiest way is buying valid addresses from a black market. A sophisticated SPITer can get a valid account by using a harvesting technique. Assuming a SPITer has an account at the provider. This provider distributes SIP URIs that correspond to the following scheme. The address begins with the digits '999' followed by five more random digits. If the SPITer has a knowledge about this address uniform, he can find out other assigned URIs. He will send an `INVITE` message to each SIP URI and analyze the answer of the SIP proxy. If the URI is not assigned, the proxy will send `404 Not Found` response. If the assigned URI is not registered at the moment, a `480 Temporarily Unavailable` will be returned. If the URI is assigned and the user is registered, the call will be established and answered with a `200OK` response. Now, the SPITer has a valid address list of this provider. Note that, it is not necessary to use `INVITE` message to harvest the numbers. `REGISTER` request could also be used to find out a valid address.

2. **Session Establishment**
   Generally, we can distinguish two possible ways of session establishment. The SPITer can establish a session by sending an `INVITE` message via the SIP proxy or sending directly to the endpoint without involving the proxy. In this work, I focus only on the commercial scenario, session establishment

via proxy.

3. **Message Sending**
   The last step is the message sending after the session has been established. Which type of media is sent depends on the scenario in which the SPIT attack takes place that is described previously.

2.4. Denial of Service Attacks (DoS)

Denial of Service (DoS) attacks are a class of network attack aim at preventing legitimate users from receiving service with some minimum performance. There are two common strategies to launch a DoS attack, *(1) by exploiting a software vulnerability* and *(2) by depleting resources of the target host.* To exploit the vulnerability, an attacker sends a crafted messages that takes advantage of that given vulnerability. These messages can crash a target system. The latter method aims to overwhelm a resource at the target by generating more requests than the target can handle. Such attack is generally hard to detect, as the utilised attack messages are usually valid messages and thus, not easily distinguishable from regular messages. Basically, an attacker launches multiple session initiation requests, but does not finalise the handshake after the server responds to the request. Thus open sessions that consume memory are created at the target. The server cannot free this memory immediately as it has to assume that the missing handshake messages have been lost and will eventually be re-sent by the sender. With too many concurrently open sessions, the server will run out of memory resources and will not be able to respond to further requests. In this work, I focus only on the latter case, especially *SIP Message Flooding* attacks.

2.4.1 SIP Flooding Attacks

Different SIP messages can be used for message flooding, e.g., `REGISTER` and `INVITE`. User Equipment (UE) needs to send `REGISTER` requests in order to register with the IMS server. In the case of `REGISTER` flooding, attackers send numerous bogus registration requests with invalid credentials in order to consume the processing resources of the server. The server will spend time looking into the

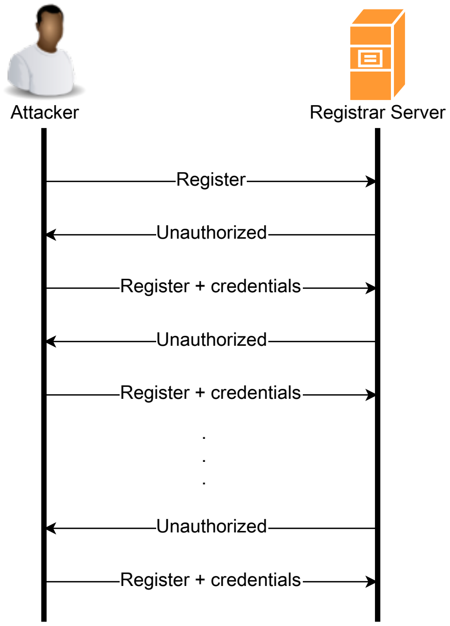

Figure 2.5. REGISTER flooding attack.

database and sending back error messages which will be ignored by the attacker. An example using `REGISTER` messages is shown in Figure 2.5.

A SIP component is most vulnerable if it has to keep state for a longer time, which is the case in the `INVITE` process. In this case, attackers send a large number of SIP `INVITE` messages to SIP proxy within a short period of time. As a transactional protocol, according to RFC 3261, SIP requires the server to maintain a state for each `INVITE` message for some time period waiting for the associated `200OK` acknowledgement message. Also, after forwarding a final non `200OK` response, i.e. 3xx - 6xx, the server needs to wait for the `ACK` message and re-transmits the response for a period of up to $64 * T1$, where $T1$ is usually set to 500 msec. In case the server has forked a request to different destinations, it needs to maintain a copy of the incoming request as well as a copy of all forked requests. The attacker generates numerous `INVITE` messages without any respond message to exploits SIP's three-way handshake procedure, and specifically its limitation in maintaining half-open connections. If the attack intensity is high enough, the resources of the server will be exhausted and then the server becomes unavailable. Figure 2.6 shows this scenario.

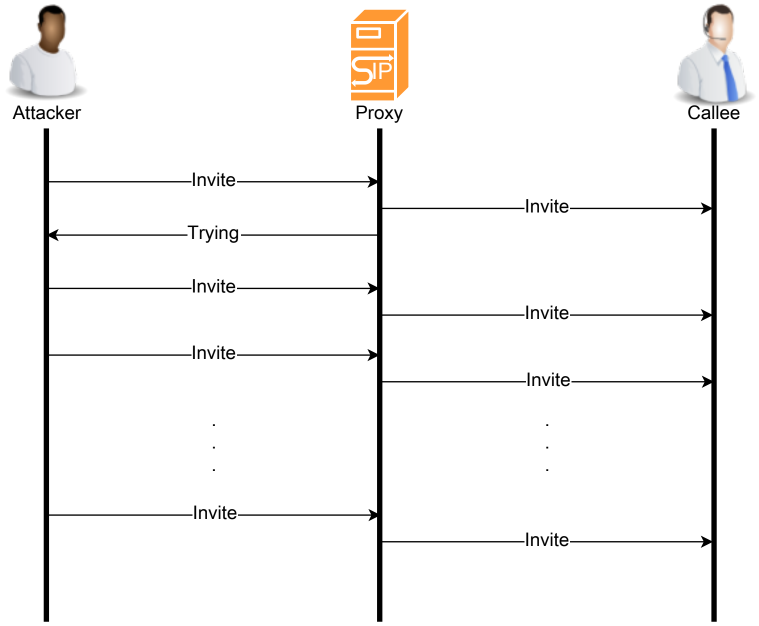

Figure 2.6. INVITE flooding attack.

In this work, I focus the evaluation on an `INVITE` flooding case first. There are two reason for choosing an `INVITE` flooding attack. First, the overhead for an incomplete `INVITE` handshake is significantly greater as compared to other SIP floods. Second, an `INVITE` attack affects two networks/servers as compared to the case of `REGISTER` floods that affects only a single network/server. However, attacks utilizing other SIP attributes can be addressed in a similar way.

2.4.2 Exploitable Resources

There are three main resources that can be targeted in a SIP flooding attack: memory, CPU, or bandwidth.

Memory

A SIP server needs to store each incoming request into its buffers for processing the message. The amount of buffer size and the timeout period vary depending on a configuration. Basically, the server needs to maintain the data while contacting another entity such as AAA and DNS servers. For example, in case of an `INVITE` message, a proxy will forward this request and wait for a reply. During this time, state memory is consumed at the proxy. If too many such messages are encountered, the proxy's memory will be exhausted.

CPU

In this case, the target is flooded with more messages than it can process at a given time. After receiving a SIP message, the SIP server needs to do many tasks such as parse the message, security check, and forward the message. Since SIP is a text-based protocol, the time taken to parse a message depends on the efficiency of the message parser and the content of the message. Furthermore, if SIP authentication information is supplied in the flooding message, it has to calculate if the user is authorised to access the service. When the target CPU cannot continue its operation, it will affect the input of other entities.

Bandwidth

The target is flooded with more messages than the network can handle. This involves overloading the access links connecting a SIP server. It causes the loss of SIP messages. Consequently, this loss causes longer session setup times or the failure of session setups. This is a general DoS flooding problem and not specific to SIP networks.

Chapter 3

Trust-Based SPIT Detection

The growing popularity of VoIP and the relatively low cost of the service make VoIP an attractive tool for spammers. Spammers or telemarketers will use VoIP to make unsolicited calls for the same purposes as email spam. This VoIP spam, a.k.a. SPIT, would be more difficult to handle because of the real-time processing requirements of voice calls. If operators are not able to combat SPIT effectively, it will affect the users' confidence.

The initial stage of voice communication is call setup, a handshake mechanism between a caller and a callee as shown in Figure 2.1. At this point, only the identity of both a caller and a callee are provided. The voice media will be exchanged after the callee accepts the call. Thus, a SPIT detection based on content filtering cannot prevent the phone from ringing. In addition, voice packets must be delivered to the user synchronously. Any delay caused by SPIT detection engine will result in a degraded call quality. Therefore, an effective solution to deal with SPIT must rely on the identity of the caller rather than call content. However, discriminating between a legitimate caller and a spammer is not an easy task. In this work, my focus is on developing a scheme that achieves this goal.

To be effective, a SPIT prevention system has to meet the following basic requirements.

1. It must minimize the probability of blocking legitimate calls while maximizing the probability of blocking SPIT calls.
2. It should minimize the additional effort imposed on users.
3. It should be deployed without any significant changes in the existing infras-

tructure.

4. It should be as general as possible to allow it to be applied despite barriers such as different cultures and languages.

However, most detection systems do not meet all of these general requirements. For example, challenge-response schemes require significant interaction with the caller, making them too intrusive and inconvenient for the user. There is not enough interest to adopt such measures because they are so inconvenient. To fulfill these requirements, I propose a novel trust-based mechanism using call direction and duration to distinguish between legitimate users and spammers. The motivation of my approach comes from the simple observation that a legitimate user typically makes and receives calls and many of the calls last for long durations. On the other hand, a spammer's goal is to deliver information to as many people possible, in as little time, by making a large number of short calls. Furthermore, a spammer will typically receive no calls or a much smaller number of calls. Hence, the difference in spammers' call pattern is largely unidirectional while it is bidirectional for legitimate users. I take this difference in call pattern and call duration to create a trust value for each user.

The following scenario shows how my trust-based approach can be applied to identify spammers. Assume that Alice makes a call to Bob. If Bob accepts the phone and talks to Alice, after completion of the conversation, a trust value can be generated signifying that Bob and Alice trust each other enough to talk for a certain duration of time. The longer call duration infers a highly trusted friend. As basic intuition, if a user receives calls of significant duration on a regular basis, it is likely that the caller is a legitimate user and not a spammer. This call duration will be calculated as a trust value and assigned to Bob.

The rest of the chapter is as follows. Section 3.1 briefly describes some SPIT detection solutions and their drawbacks. The trust calculation and trust inference are explained in Section 3.2 and 3.3. Section 3.4 presents a social reliability filtering which is used to check a past behavior of a caller. An evaluation and its results are discussed in Section 3.5. Finally, Section 3.6 discusses some concerns about the proposed technique.

3.1. Related Work

This section presents the state of the art of techniques to prevent and mitigate spam in VoIP networks. Rosenberg *et al.* provided a comprehensive reference for the various possible solutions that can be explored for SPIT [12].

Blacklisting is an approach whereby the spam filter maintains a list of call numbers that identify spammers. However, collecting these numbers can be tedious and users can still receive unwanted phone calls from numbers not on the list. Whitelisting is the opposite of blacklisting. It is a list of valid senders that a user is willing to accept calls from. Unlike blacklists, a spammer cannot change identities to get around the white list [12]. However, this approach greatly reduces the usability of VoIP because legitimate callers not in the list cannot contact the user at all.

Quittek *et al.* proposed a hidden Turing test technique to identify spammers [13]. This requires a SIP server or a user agent to check the Real-time Transport Protocol (RTP) before establishing a call session between a caller and a callee.

Vinokurov and MacIntosh propose a VoIP spam detection based on recognizing abnormalities in signaling message statistics [14]. A caller that sends too many call setup requests, while at the same time receiving too many or too few call termination requests in a relatively short time, is assumed to be a spammer. The difficulty of this approach is it has to maintain the signaling behavior of every caller in the system which needs to be updated for every call that a node makes or receives.

Dantu and Kolan proposed a combined filter technique based on trust and reputation for detecting spam [15]. They combined techniques such as rate limiting, Bayesian learning, and the concept of social networks for predicting the nature of a call. During the learning period, human intervention was required to identify unwanted callers. Even though this technique can isolate spammers, it suffers from two major drawbacks. First, trust and reputation in this system were assigned to an entire domain rather than individual users. Thus, if a particular domain has a lot of spammers and few legitimate users, those legitimate users would be penalized along with the rest of the spammers. Second, the system relies exclusively on the users' feedback to report spam domains. If the users do not report spam, those domains can continue sending spam. To solve these problems, trust value in this work is individually and automatically assigned to each friend in the buddy list based on the direct experience of a user. The manual feedback reports from users are not required.

Balasubramaniyan *et al.* propose to use call duration and social network graphs to establish a measure of reputation for callers, named CallRank [16]. In this filtering scheme, call duration is the main factor deciding the credibility of the caller. It is used along with the Eigentrust algorithm to develop a global view of the reputation of all users who either belong to or interact with a domain. A callee can decide to answer or reject a call based on this mechanism. However, CallRank produces false positives when a new legitimate user joins the VoIP system. Because he has no social network linkage in that system, all his calls will be classified as spam calls. Due to its centralized perspective, there are some problems. First, the users are usually reluctant to give a negative rating because of the other's negative rating. Second, if a user has a bad reputation rating, the system will discard its old identity. The third problem is that users can increase their reputation artificially by creating fake identities and using them to give themselves a high rating. Unlike this work, the proposed technique acts as a decentralized perspective to avoid these problems. Each user is responsible for evaluating the trust of other users based on their direct interactions.

Another study proposed a collaborative reputation-based voice spam filtering framework [17]. This approach used the cumulative online presence duration of a VoIP user as a reputation value. The authors automatically classified calls shorter than 20 seconds as spam calls. However, with this method, spammers can increase their reputation easily by maintaining a connection with the VoIP server. A trust value in this proposal, on the other hand, is calculated by using call duration and call direction. Assuming that user A and user B are friends, when user A calls user B, the trust value from user A will be automatically assigned to user B according to the duration of a conversation. In the case of a spam call, a spammer calls user A. The trust value will never be given to a spammer because user A does not call a spammer. If a spammer wants to increase his trust value, he needs to trick user A into calling him back. Therefore, it is difficult to alter the trust value in our proposed technique.

3.2. Trust Calculation

When dealing with strangers in electronic interactions, establishing trust for authorization is the core challenge. Trust has been traditionally proposed as a method to enhance security in many systems. The basic idea is to let parties rate each other and use the aggregated rating about a given party to derive a trust score, which can assist other parties in deciding whether or not to interact with that party in the future. In voice communication, trust represents a model of past interactions between the calling party. In this work, I attempt to formalize a structure similar to human intuitive behavior for detecting SPIT based on a trust relationship with the caller and calculate trust automatically from an individual aspect. To begin, I define trust using three properties:

1. The trust represents a callee's belief in a caller's reliability based on his/her own direct experiences.

2. The trust of a caller can be increased or diminished over a period of time based on the interaction with the callee (trustor).

3. The trust can be derived from outgoing calls. This means we assign a trust value to a user when we call him.

In IP telephony system, there are many factors that can be used to identify a malicious node, as shown in Table 3.1. However, some of these factors are quite ineffective. A sophisticated spammer can observe a filtering system and then adjust his/her spam behavior in order to break the detection criteria. For example, the error of calling occurs when the destination address does not exist. Because spammers randomly generate the target callee IDs, there is a possibility that some of them do not exist. Then, the call error rate of a spammer will be high. However, a spammer can collect the existing numbers from Yellow Pages or buy them from the black market to reduce this rate. For a call rate and a call interval time, they are the parameters that can be adjusted easily.

Call duration and direction are selected to calculate a trust value because they are reliable. Call duration is an important information used to distinguish a spammer because, in general, people do not like to talk with a spam caller for

Table 3.1. Calling characteristics

Factor	**Legitimate**	**SPIT**
Call duration	Irregular	Usually very short
Call direction	Bidirectional	Unidirectional
Call error rate	Low	High
Call rate	Low	High
Call interval time	Irregular	Regular

long. Also, a pre-recorded massage is not too long to save on call fee. Therefore, the call duration of a spam call will be significantly shorter than a legitimate call. Moreover, most spam calls are a unidirectional type of communication because a spammer calls target users only. There is a rare case that a normal user may call back to a spammer. Therefore, if a caller has a high rate of outgoing call, he can be classified as a spammer. Moreover, these two variables cannot be altered by a spammer except when using social engineering techniques. The spammer has to trick the user into calling him back and maintaining a long call in order to increase his trust value. In addition, since trust is derived from direct interaction between two people, they can express the extent of trust reliably.

Additionally, using call duration has the following advantages: it is implicit, quantifiable, easily verifiable, and easily understood. In VoIP/IMS networks, their servers keep track of call duration for billing purposes. The proposed detection system does not require any alteration of the existing infrastructure. As shown in a SIP session setup example, Figure 2.1, call duration represents the time between the end of call setup (`200OK`) to the start of call tear-down (`BYE`).

The total duration of all outgoing calls of each friend in a buddy list is calculated after a period of time, t, e.g., a billing period. The raw trust value of a friend i (R_i) can be computed by comparing this cumulative value (C_i) with the average call duration of all friends as shown in Eq. (3.1); where n is a number of friends in the buddy list.

$$R_i = \frac{C_i}{\sqrt[n]{\prod_{j=1}^{n} C_j}} \tag{3.1}$$

In this work, I use the range of numbers between 0 and 1 to represent the trust value, where 0 indicates a spammer and 1 indicates a normal user. If R_i is

Table 3.2. Data structure of a buddy list.

Attribute	**Data Type**
Friend	*char*
Cumulative duration	*integer*
Raw trust	*integer*
Final trust	*integer*

greater than 1, it is rounded down to 1. This means that higher trust values are given to friends with a duration of calls longer than average. The geometric mean is applied as an average function. Note that, the zero value (no call duration) is ignored to avoid the computation error. The advantage of this method will be described in Section 3.2.1.

Because trust depends on past experiences, I combined the raw trust value and the historical trust, which is the trust value computed at the previous time, to compute the final trust. The final trust, $T_{i(t)}$, of each friend is computed as follows;

$$T_{i(t)} = \alpha R_{i(t)} + (1 - \alpha) T_{i(t-1)}, \tag{3.2}$$

where $\alpha < 0.5$. According to human reasoning, I considered past experiences before deciding on something. Therefore, I give a higher weight value for historical trust than for raw trust.

All variables used for calculating trust values are stored in a buddy list. Table 3.2 shows the data structure of this list. The Friend attribute contains a call ID of a friend. A call from a friend, who is already in a callee's buddy list, is automatically forwarded to a callee. The cumulative duration is the total outgoing call duration to this friend.

3.2.1 Geometric Mean vs. Arithmetic Mean

In mathematics, a geometric mean is a type of mean or average which indicates the central tendency of a set of numbers. It is technically defined as the n^{th} root product of n numbers: $\bar{x} = \sqrt[n]{x_1 \times x_2 \times \ldots \times x_n}$. The geometric mean of a dataset is always less than or equal to the data set's arithmetic mean (the two means are equal if and only if all members of the data set are equal). It is useful

Table 3.3. Call pattern of each user in Figure 3.1.

Time Interval	Total Call Duration			
	A	B	C	D
1	1,000	50	30	0
2	1,000	50	30	0
3	1,000	50	30	0
4	1,000	50	30	0
5	1,000	50	30	10
6	1,000	50	30	10

in case the cumulative call durations of each friend are quite different when a user makes an unusually long call. For example, assuming A, B, C, and D are my friends. I call the first three friends 1,000, 50, and 30 minutes respectively and call friend D 10 minutes only in the last two periods of the observation, as shown in Table 3.3. Figure 3.1 shows the result of the trust values computed by geometric mean (solid line) and arithmetic mean (dash line). From this result, the geometric mean produces higher results than the arithmetic mean. At time inerval 5^{th}, I start calling friend D. The trust value computed by geometric mean is increasing while continuously decreasing in another one. Therefore, with this formula, I can conclude that the trust value can be increased significantly by generating enough call duration relative to other friends.

3.3. Trust Inference Mechanism

A trust value in the previous section is only assigned to friends in the buddy list who have direct interactions. In the real world, there is a possibility that a call will come from an unknown person. To estimate the trustworthiness of an unknown caller, I apply a situation found in daily human life. Generally, when encountering an unknown person, it is common for people to ask trusted friends for opinions about how much they can trust this new person. Therefore, we gather the trust values of an unknown caller from other friends in the network who already know about that person in order to classify a call. Hence, the property of trust in this work is transitive. The trust values of friends in the buddy list will

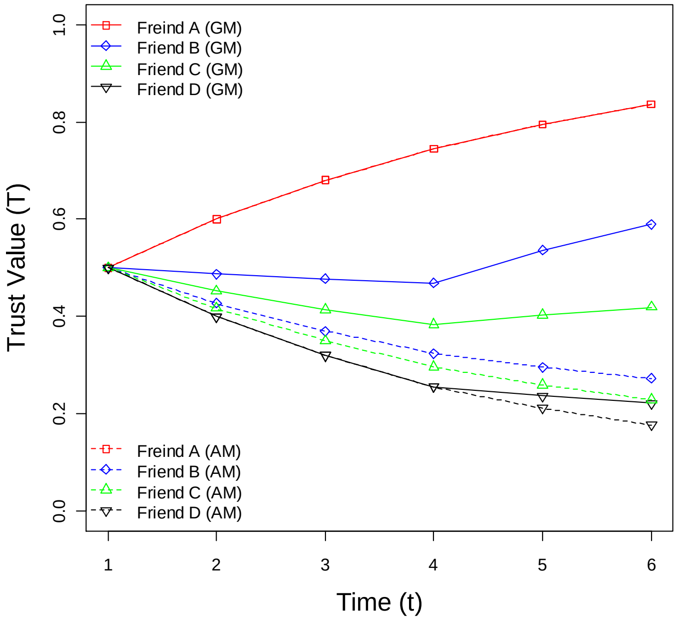

Figure 3.1. Trust values produced by geometric mean and arithmetic mean.

be shared to other nodes through a relationship path in the VoIP/IMS networks. These inferred trusts will be used when a caller and a callee do not have a direct relationship.

Assuming that VoIP system is represented as a social network. It contains nodes that are User Agent Clients (UAC). Every node maintains a buddy list. This social network is constructed by connecting a node to all the nodes in its buddy list. The buddy lists are assumed to be kept in a central database on the provider side and shared with others during the trust computing process without any privacy concerns. We also assume that any network that interconnects with others should make use of strong SIP identity as described in RFC 4474 to protect a call identity [18].

In case of a scalability improvement, the buddy lists need to be distributed throughout the global telephony network. There are already some possible techniques available for sharing the buddy lists. For instance, Trust path discovery, the IETF Internet draft, offers address book propagation within SIP messages

[19]. In general, though, we need an agreement between different providers that allow each other to exchange any information related to the users' buddy lists. However, this step is not included in the scope of this work.

According to human reasoning, a person is much more likely to believe his friends than a stranger. Furthermore, we trust to some extent the opinion of friends about their own friends as well. Thus, it is possible to find a path of friends from trustors to trustees with an appropriate discounting of level of trust [20]. For example, if Alice trusts Bob completely and Bob trusts Carol completely, then Alice may trust Carol, but not necessarily completely. Therefore, a multiplicative function is appropriate to calculate an inferred trust of an unknown caller as shown in Eq. (3.3),

$$T_{callee,caller} = \prod_{m \in path}^{caller} T_{m,m+1}, \tag{3.3}$$

where m is a user between a callee and a caller.

One interesting phenomenon in a social network is the seven degrees of separation. Everyone is connected through not more than seven intermediaries [8]. By this concept, the trust inference process limits a relationship length between a callee and an unknown caller within a count of seven hops.

For newcomers or unknown callers of whom trust cannot be computed, the system will assign an initial trust value. The initial trust value is set to be slightly higher than a trust threshold. It is adjustable automatically after a user has calling activity as describe in the previous section. This initial trust assignment can eliminate the barrier for new users who do not have a trust value assigned by other users.

In a real network, there is a high possibility to have many trust paths between a caller and a callee. The previous work selects only one trust path that produces the highest trust value [21]. However, selecting one trust path cannot reflect all the trust information of the caller. To compute the final inferred trust, I use the data fusion technique to combine the trust values of all trust paths together. Generally, data fusion is a process performed on multi-source data towards correlation, estimation, and the combination of several data streams into one with a higher level of abstraction and greater meaningfulness. Its objective is to obtain an optimal decision or solution by combining many kinds of information from different sources. Next, the trust aggregation methodology will be explained in details.

3.3.1 Multiple Trust Paths Aggregation

In the field of statistics, the most well-known data fusion technique is the Bayesian theory:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}. \tag{3.4}$$

Bayesian theory interprets a posterior probability, $P(H|E)$, as a measure of belief about a hypothesis or proposition H updated in response to evidence E. The prior probability, $P(H)$, reflects the belief about H in the absence of evidence. Researchers often estimate prior probabilities from empirical data, or, in the absence of empirical data, they assume them to be uniform or some other distribution. The outcome reflects these assumptions, so the critics of the Bayesian approach often point out that the method is not well-equipped to handle states of ignorance [22]. Clearly, this approach requires complete knowledge of both prior and conditional probabilities, which might be difficult to determine in practice. In contrast with the Bayesian approach, the Dempster-Shafer Theory (DST) does not require the complete probabilistic model. I will now briefly introduce the key concepts of this theory. DST can be considered an extension of Bayesian inference [23]. It is a system for combining evidence from different sources and arrives at a degree of belief under uncertainty. Let a frame of discernment $\Theta = \{T, \neg T\}$ be two events under consideration; e.g. T = trust in a caller, $\neg T$ = distrust in a caller.

Definition 1

Let Θ be a frame of discernment. A function $m : 2^{\Theta} \to [0, 1]$ is defined as Basic Belief Assignment (BBA) when it satisfies the following two properties:

$$m\{\emptyset\} = 0 \text{ and} \tag{3.5}$$

$$\sum_{A \subseteq \Theta} m(A) = 1. \tag{3.6}$$

Thus, we have $m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}) = 1$.

Definition 2

The belief function (Bel) for a set A is defined as the sum of all the assignments of the subsets of A:

$$Bel(A) = \sum_{B \subseteq A} m(B). \tag{3.7}$$

For our case, we have the following:

$$\begin{aligned}
&Bel(\{T\}) = m(\{T\}) \\
&Bel(\{\neg T\}) = m(\{\neg T\}) \\
&Bel(\{T, \neg T\}) = m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}).
\end{aligned}$$

For instance, suppose Alice and Bob are our friends who have a trust path to the unknown caller. Assume Alice's trust path is trustworthy with a value of 0.8. Alice states that the caller is trustworthy. This means Alice's claim gives evidence for 0.8 degrees of belief in the caller's trustworthiness, but a zero degree of belief (not 0.2) that the caller is untrustworthy. This zero value mean that Alice's evidence gives no support to the belief that caller is untrustworthy. The 0.8 and the zero together constitute a belief function.

The combination of the evidence from different sources is done through the combination rule that is defined in the next definition.

Definition 3

Let Bel_1 and Bel_2 be belief functions over Θ, with BBA m_1 and m_2. Then the function $m : 2^{\Theta} \to [0, 1]$ that is defined by

$$\begin{aligned}
&m_{1,2} \{\emptyset\} = 0 \text{ and} && (3.8) \\
&m_{1,2}(A) = \frac{\sum_{i,j:A_i \cap B_j = A} m_1(A_i) m_2(B_j)}{1 - K}, \text{ where} && (3.9) \\
&K = \sum_{i,j:A_i \cap B_j = \emptyset} m_1(A_i) m_2(B_j) && (3.10)
\end{aligned}$$

for all non-empty A.

Suppose that Bob's trust path is trustworthy with a value of 0.9, independently of Alice. Then we have

$$m_1(\{T\}) = 0.8, m_1(\{\neg T\}) = 0, m_1(\{T, \neg T\}) = 0.2$$
$$m_2(\{T\}) = 0.9, m_2(\{\neg T\}) = 0, m_2(\{T, \neg T\}) = 0.1$$

Finally, the aggregated trust value is

$$m_{1,2}(\{T\}) = 0.72 + 0.08 + 0.18 = 0.98.$$

3.4. Social Reliability

Before forward a call to a callee, the detection system will check the past behavior of this unknown caller that interacts with other users. This feature is called the Social Reliability (SR) of a user. Two variables are considered to compute this feature: the degree of activity and the unique call.

Degree of Activity

This is the ratio between incoming and outgoing calls during an observation period. Users who have a low degree of activity are those that make more calls than they receive. This can be an indicator for detecting a spammer. If the ratio is greater than 1, it is rounded down to 1. A high level of incoming calls might indicate a call center, which is not classified as a malicious user. Although a group of attackers could mimic call activity by calling each other, on the VoIP system this would mean an extra cost.

$$\text{degree of activity} = \frac{\text{incoming calls}}{\text{outgoing calls}} \tag{3.11}$$

Unique Call

This feature allows us to identify an anomalous caller who makes a significant number of calls to different callees. Generally, a normal user will call a set of destination numbers, i.e. callees that have already been contacted before.

$$\text{unique call} = \frac{\text{unique calls}}{\text{outgoing calls}} \tag{3.12}$$

Since the two variables Degree of Activity and Unique Call are not the direct interaction between a caller and a callee, we will not use them as trust values. Due to the imprecision of these values, we use fuzzy logic inference rules to calculate the social reliability of users. In this case, the Degree of Activity and the Unique Call are used as fuzzy descriptors. The membership function of these two variables is shown in Figure 3.2. An example of fuzzy inference rules is shown as follows:

- If *degree of activity* is low and *unique call* is low then *social reliability* is verylow
- If *degree of activity* is med and *unique call* is low then *social reliability* is low
- If *degree of activity* is med and *unique call* is med then *social reliability* is med
- If *degree of activity* is med and *unique call* is high then *social reliability* is high
- If *degree of activity* is high and *unique call* is high then *social reliability* is veryhigh

The SR of a user will be strong if he balances his incoming and outgoing calls and if most outgoing calls go directly to a set of friends.

3.5. SPIT Detection Performance

The sensitivity, specificity, and false positive (FP) are computed to represent the ratio of correctly classified calls. Sensitivity (3.13) is the proportion of correctly detected spam calls to all actual spam calls. With higher sensitivity, fewer actual cases of SPIT go undetected. Specificity (3.14) is the proportion of correctly detected legitimate user calls to all actual legitimate calls. Higher specificity indicates that the system detects legitimate calls more accurately. False Positive (FP) is defined as the number of legitimate calls classified as SPIT.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.13}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{3.14}$$

3.5.1 Datasets

An evaluation of a SPIT detection in the real world would require call logs from a VoIP system along with actual cases of SPIT. Unfortunately, call logs are not easy to come by due to privacy concerns and SPIT is still not sufficiently widespread. Instead, a synthetic call workload is simulated to evaluate the effectiveness for

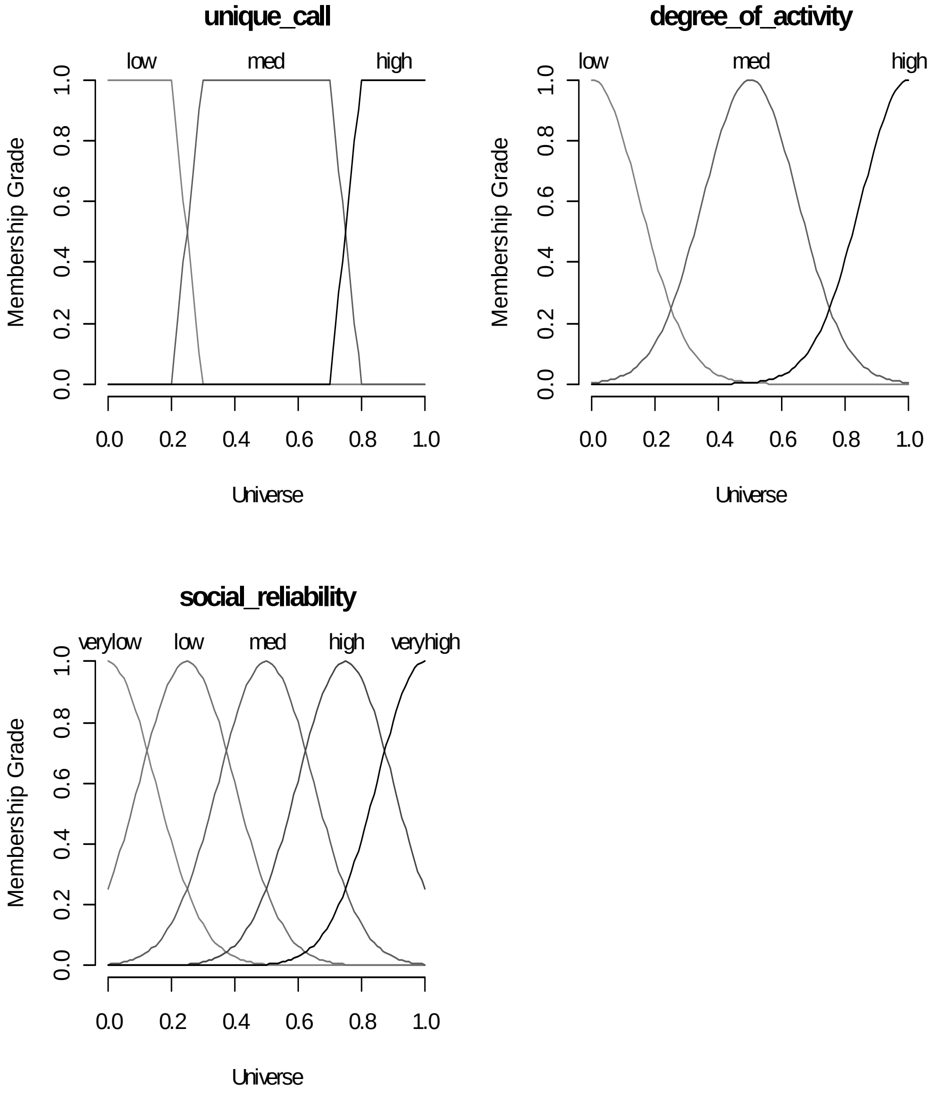

Figure 3.2. Fuzzy membership functions.

ensuring that the simulations model real world call characteristics as closely as possible. The objective of the simulations is to study the performance of the proposed technique in terms of SPIT detection accuracy.

Many researches simulated their testbed by only randomly choosing call parties, such as [15] and [16]. But in order to simulate the realistic conditions of the

Table 3.4. VoIP Network Datasets

Dataset	**Nodes**	**Edges**	**CC**
Random Graph	1,000	Varied	0.1–0.5
Epinions	75,879	508,837	0.23

VoIP user network, we used two kinds of datasets: directed random graphs and the Epinions social network. A random graph is a graph that is generated by some random process. It is obtained by starting with a set of n vertices and adding directed edges between them at random. The nodes in the graph are considered to be users and edges are designated as call directions. We adjust the clustering coefficient (CC) of the graphs to evaluate our detection accuracy among different network characteristics. The Epinions dataset is the who-trusts-whom online social network of a general consumer review site Epinions.com [24]. Members can decide whether or not to trust each other. All the trust relationships interact and form a web of trust, which is then combined with review ratings to determine which reviews are to be shown to the user. The details of these datasets are shown in Table 3.4. Table 3.5 shows all parameters used in the simulation. We referred to the published reports by NTT East Corporation [25],[26] to decide the average call duration and number of calls of legitimate users. These reports show the IP phone usage statistics during April 2011 to March 2012. During the simulation, both new legitimate users and spammers are randomly added into the network.

3.5.2 SPIT Detection Architecture

Figure 3.3 shows the overview of the detection system. At the initial state, trust values of all friends in the buddy list are set to be any value that is greater than the predefined SPIT threshold depending on the operator policy. This trust value is updated automatically every predefined period, e.g. a billing period. A call from friends present in the buddy list is accepted automatically. For an unknown caller who is not in the blacklist, the inferred trust is computed by the algorithm described in Section 3.3. If this trust is lower than a predefined SPIT threshold, this call will be rejected. Otherwise, the system finally checks the SR of the caller. This call will be connected to the callee if this value exceeds the SR

Table 3.5. Simulation Parameters

Parameter	**Value**	**Description**
Legitimate call duration	104–215 sec.	Generated by using a Pareto distribution [27]
Legitimate calls per unit of time	≈2 calls/day/user	Based on operator statistics; generated by using a Poisson distribution [27]
Spam call duration	<10 sec.	Generated by using a normal distribution
Called recipient	Random	Randomly select a callee both from a buddy list and outside the buddy list by using a Zipfian distribution [16]
Trust of unknown caller	0.4	This value is assigned to an unknown caller in the case of no inferred trust
SPIT threshold	0.25	The optimized value of our proposed technique
No. of ad subscribers	Varied	Some legitimate users are randomly selected to add spammers to their buddy list by using a normal distribution

threshold. The callee can decide to save this caller in the buddy list or not. If not, the system will automatically store this contact and its trust value in a hidden buddy list for future use. This hidden list can reduce the computation task when the same caller, who is in neither the buddy list nor the blacklist, contacts the callee the next time. For a newcomer, or an unknown caller of whom trust cannot be computed, the system will assign an initial trust value and then connect the call to the callee. This can eliminate the barrier for new users who do not have a trust value assigned by other users. If a callee finds that a call is spam, he can put this caller number in the blacklist. The trust value of every node in the

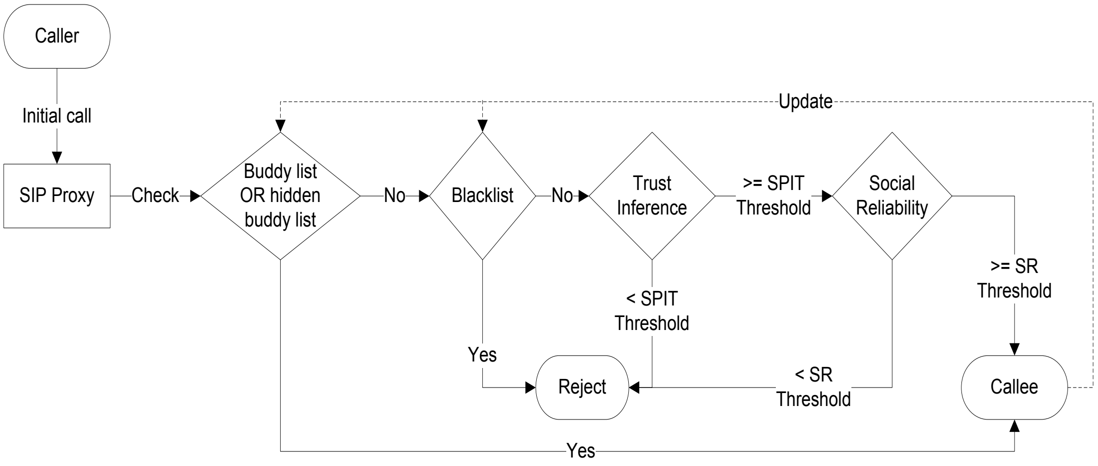

Figure 3.3. The malicious caller detection system overview.

blacklist is zero and is also used in the trust inference process.

In the following two experiments, I will compare the detection performance of three trust inference methods. First, the highest trust path between an unknown caller and a callee is selected [21]. The second method uses the DST to aggregate all trust paths. The last method uses the DST approach with the SR to filter a call.

3.5.3 Different Numbers of Spammers

I used the real social network dataset, Epinions, as the initial VoIP network in this experiment. The vertex represents a VoIP user which the directed edge represents the direction of the call from one user to another. At the initial state, assuming the pointed node is a friend in a buddy list. I ran the simulation several times and observed the detection performance with different numbers of spammers. Figure 3.4, 3.5, and 3.6 show the average sensitivity, specificity, and FP of three methods with different number of spammers: 0.1%, 0.3%, and 0.5% respectively. The right side of each figure shows the box-and-whisker plots of the results. These results indicate that the performance of the combined trust paths was better than the selecting only highest trust path of the previous work. Moreover, the SR filtering technique improved the detection performance. The average sensitivity rate and

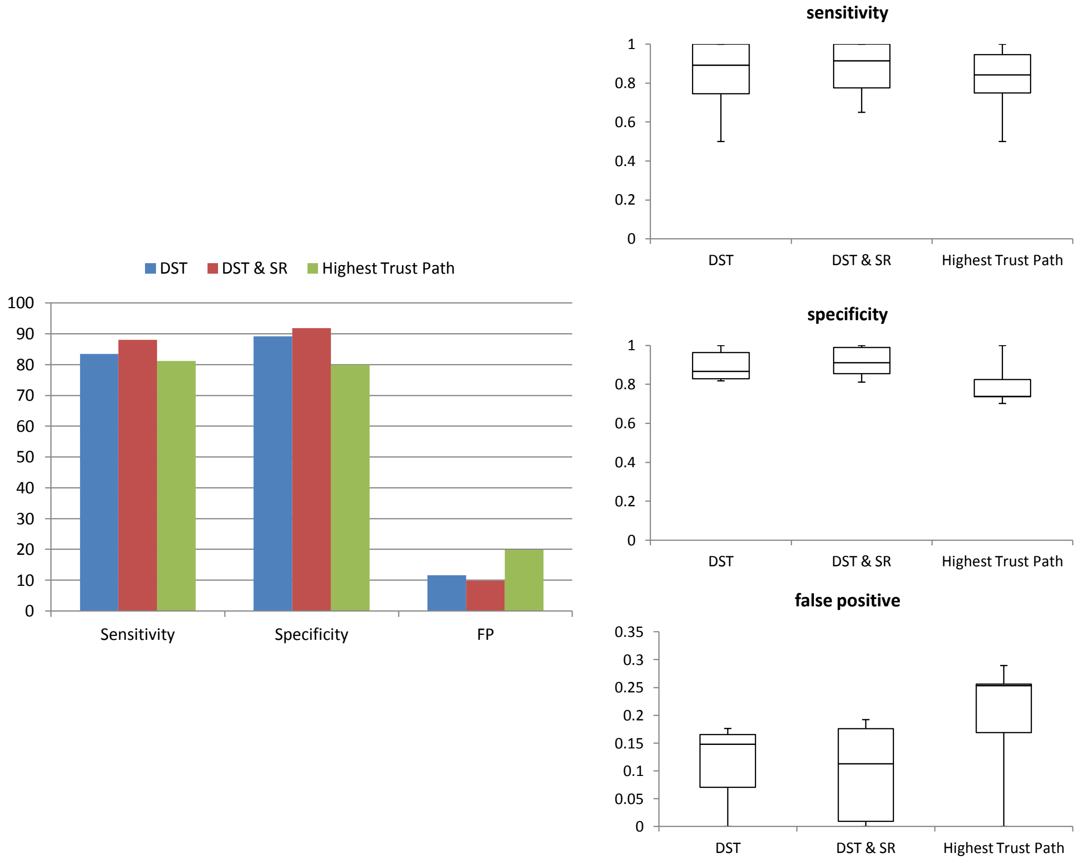

Figure 3.4. Sensitivity, specificity, and FP of 0.1% of spammers.

specificity rate of the DST with the SR approach were above 80%. From the sensitivity graphs, more spammers affected the spam classification only in the first time attack. Since we accept the first call of an unknown caller who does not have an inferred trust, when a new spammer makes a call, sensitivity dropped because every first call of the new spammers was a false negative. However, the sensitivity increased significantly in the next few periods because some nodes in the network had enough information about the spammers. After this period, the actual trust values of the spammers were inferred accurately. Of particular interest, the FP was improved significantly compared with the selecting one trust path method. With the data fusion technique, if there is a lot of SPIT in the network, we will have more evidence of it. This can enhance the detection accuracy of our system.

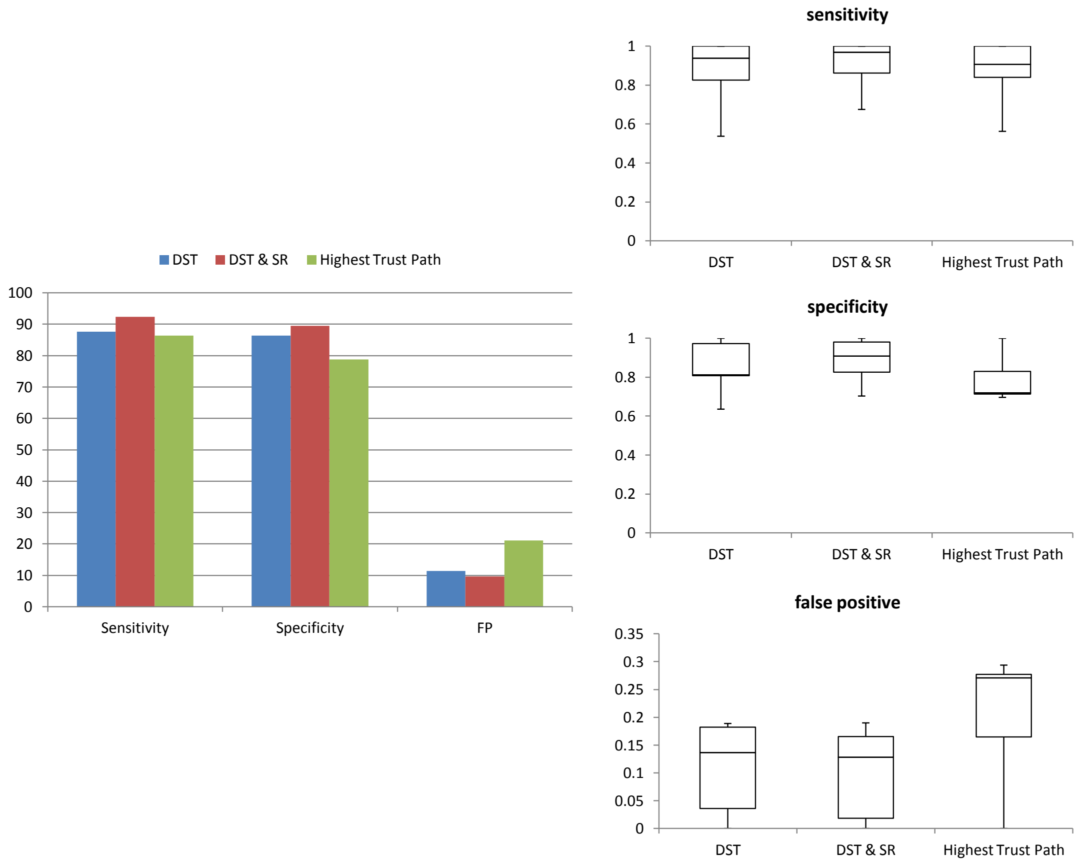

Figure 3.5. Sensitivity, specificity, and FP of 0.3% of spammers.

3.5.4 Different Network Characteristics

The objective of this experiment is to evaluate spam detection efficiency with different network distributions. Due to a lack of a real VoIP network dataset, we have to ensure that our detection can be deployed in any network. We observed the performance with five different CC random graphs: 0.1 - 0.5. In graph theory, a CC is a measure of the degree to which nodes in a graph tend to cluster together. If the neighborhood is fully connected, the CC is 1. Note that the number of spammers in this experiment is 10%. Figures 3.7 to 3.11 compare the results over the different datasets. From these results, DST with SR filtering produced the best detection performance. The average sensitivity and specificity was higher than others in every case. The FP was also the lowest. Both sensitivity and

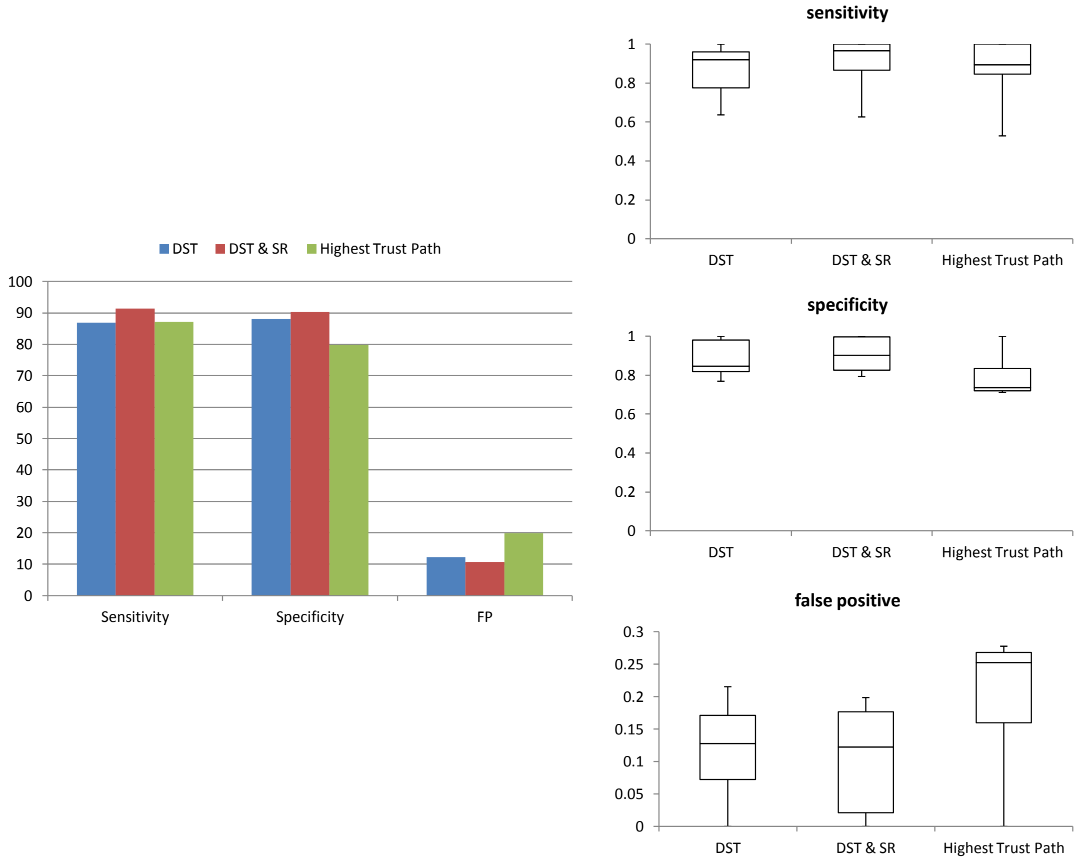

Figure 3.6. Sensitivity, specificity, and FP of 0.5% of spammers.

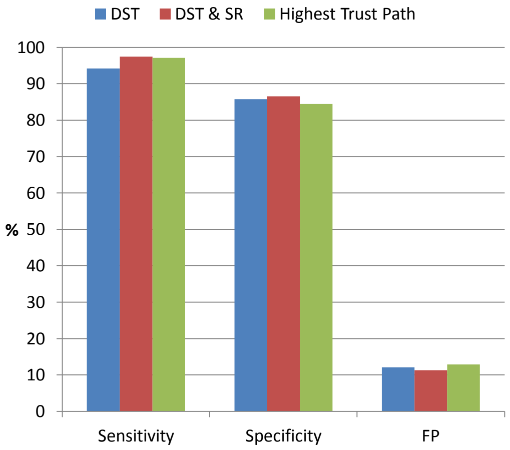

Figure 3.7. Average sensitivity, specificity, and FP of network CC = 0.1.

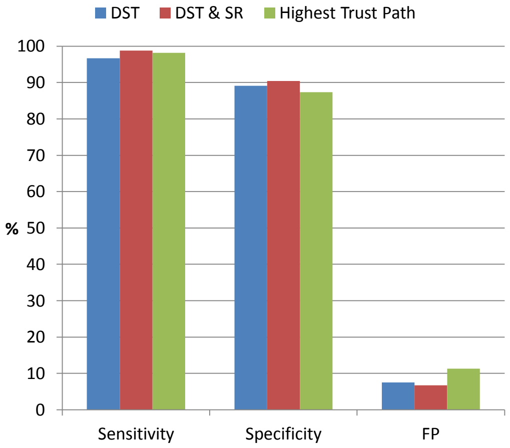

Figure 3.8. Average sensitivity, specificity, and FP of network CC = 0.2.

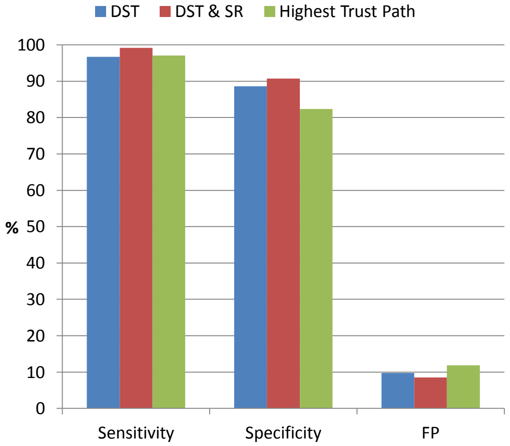

Figure 3.9. Average sensitivity, specificity, and FP of network CC = 0.3.

specificity remained high even when the network distribution was changed. The detection accuracy increased in the higher CC network. Because nodes are located closely together, they can pass the correct information of the other nodes. With the seven hops limitation, there are some misdetections in the lower CC network because it has a high possibility that two nodes are far more than seven hops. Since we accept a call that comes from a person who does not have a inferred trust. If that person is a spammer, this is a false negative.

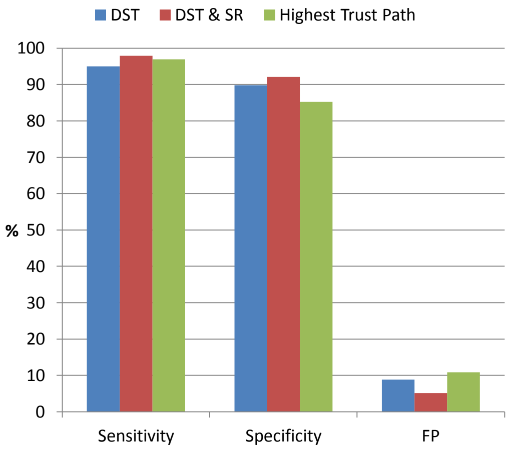

Figure 3.10. Average sensitivity, specificity, and FP of network CC = 0.4.

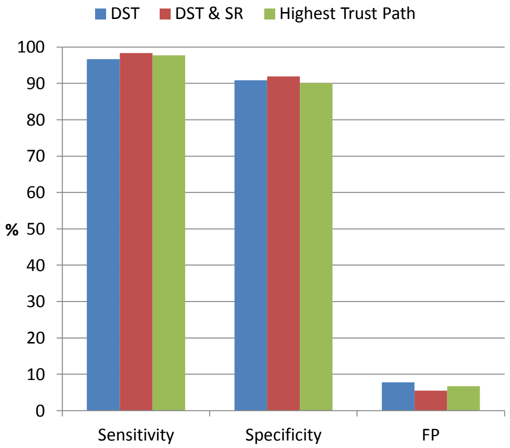

Figure 3.11. Average sensitivity, specificity, and FP of network CC = 0.5.

3.5.5 Computational Overhead

Though, filtering at the operator has to be done in real-time, the pre-computation of the trust values has to be made at regular intervals. For example, the trust value of each friend in the buddy list is updated at weekly, monthly, or at every billing period depending on the operator's policy. To evaluate the performance, I show the computation time and a memory consumption when a callee needs to calculate a trust value of unknown callers who need the inferred trust value from other nodes in the network. This CPU time also includes finding the trust path

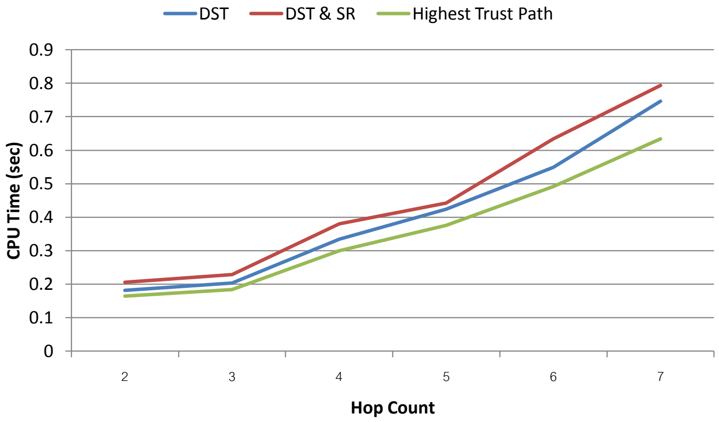

Figure 3.12. The average CPU time of different trust path limitation lengths.

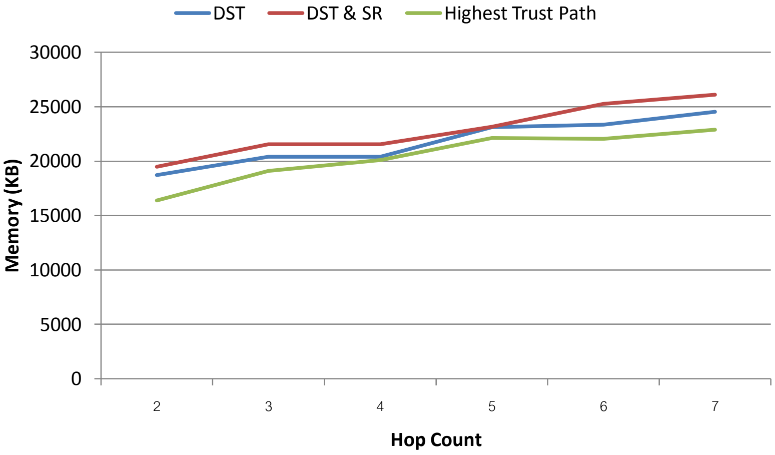

Figure 3.13. The memory consumption used in the trust calculation.

between two nodes with seven hops limitation. This experiment was conducted on the Epinion social network dataset. Figure 3.12 and Figure 3.13 show the average computation time and memory consumption of three methods with different trust path lengths. The results show that the more distant the relationship, the more computation resources were needed in all methods. When the relationship limitation length is increased, more inferred trust paths are found. The trust path combination method spends more time to aggregate all inferred trust paths together compared with the highest trust path selection method. If the number of trust paths increases, the time for the trust aggregation also increases. In addition, the social reliability filtering requires more computing resources to analyze the recent behavior of a caller. Even though the DST together with SR filtering method slightly requires more computing resources than other methods, it produces the best detection performance in terms of sensitivity, specificity, and false positive rate, as described in Sections 3.5.3 and 3.5.4.

3.6. Discussion

In this section, I discuss attacks against a trust-based system that might affect my proposed technique. I also introduce solutions to mitigate such attacks and explain the resistant ability of my SPIT detection system.

3.6.1 Sybil Attack

Generally, a trust-based system will be subverted by a Sybil attack. A Sybil attack is a type of security threat when a node in a network claims multiple identities. The attack targets the reputation system and allows the attacker to have an unfair advantage in influencing the reputation of entities on the network. In this case, a spammer creates a large number of entities and uses them to gain a disproportionately large influence. A system's vulnerability to this attack depends on how cheaply trust establishment can be generated. However, with my proposed technique, this attack would be difficult to construct for two main reasons. First, the cost of a VoIP service usage makes it difficult to keep high trust value among spammers. Trust in my system is calculated based on the call duration of outgoing calls. If a spammer S1 wanted to maintain a high trust value for a neighbor spammer S2, S1 would have to call S2 frequently. Thus, if the spammers wanted to construct a large strong spammer community, they would have to call each other frequently. At the same time, they would have to maintain a balance in their in-out calling degrees in order to keep an appropriate SR level. As a result, the cost would make it counterproductive for their business. Second, in case a legitimate user subscribes to a spammer, it does not affect my trust evaluation even though the trust of spammer from this user is inferred to other nodes. According to Eq. (3.2), a trust value of a subscribed advertisement service will be continuously decreasing. Thus, the inferred trust is possibly lower than the SPIT threshold.

3.6.2 Whitewashing

A whitewashing attack occurs when a node having a poor trust value changes its identity to start a fresh and escape from the consequences of its bad actions. Since a new spammer has a trust value equals to the unknown caller trust (higher than a threshold), its first call can bypass the detection system. From those results, even though the system can completely detect a spam call in the next computing period, we still suffer from the attack when a spammer reenters the VoIP system with new a caller id. Mitigating this attack, a VoIP operator should decide a proper contract agreement. For example, the registration fee can include a dissuasive deposit but it can be refunded at the end of the contract. Following this contract, a spammer can generate spam calls only within a short period. After that, its call cannot be established anymore. The spammer has to wait until the end of his contract to get his deposit back. Consequently, this scenario will affect the spammers business. In addition, this solution is also applied to prevent Sybil attacks because it can reduce a number of new caller ID registrations.

3.6.3 Malicious Spies

Unlike the previously described attacks that employ primarily one strategy, this attack utilizes multiple strategies, where spammers employ different attack vectors, change their behavior over time, and divide up identities to target. This scenario may be complex in the area of spam but it has a possibility to occur in the real world. One example of this attack is where colluders divide themselves into teams and each team plays a different role. Some teams will exhibit honest behavior while the other teams exhibit dishonest behavior. Spammers exhibiting an honest behavior in my system is impossible because one characteristic of my trust evaluation is based on human relationship but a spammer is an automated system that does not have a real human relationship. However, there are alternative ways to get honest colluders that can be classified into two types: short-term and long-term spying. For short-term spying, a spammer uses a malware installed in legitimate user devices that is responsible for making call to a spammer. This technique is effective only during a short time period because a user can notice an unusual calling behavior easily in a bill. So hiring and purchasing the existing legitimate callers are the most effective method for a long-term attack. The honest teams serve to build their own trusts as well as reduce the declining speed of the trust of the dishonest teams by often calling to them while the dishonest teams attempt to conduct SPIT activity.

This attack is most effective when there are several colluders for each role. Larger numbers allow each colluder to be linked less tightly to other colluders, which make detection much more difficult. Therefore, the sophisticated colluders can balance between spam behavior and avoiding detection. In addition, they can subvert the detection system perfectly if some nodes in honest teams are supernodes who have a high degree of centrality in the network. Due to a lot of bidirectional communication to many other legitimate users, it is a high possibility that the inferred trust of a spammer (in dishonest team) will be propagated to other nodes from this colluder. Figure 3.14 is the one example of this attack. Assume that the trust value among spammers including colluder C is one ($T_{c,s} = 1$) and the trust values among legitimate friends are greater than 0.9 ($T_{L1,L2}$, $T_{L1,L3}$, $T_{L1,L4} > 0.9$). At time $t = 1$, a spammer in a dishonest team calls three legitimate users ($L2, L3, L4$) who do not have any relationship with the spammer. They accept the call as an unknown caller. After they realize that this call is a spam, they put this caller in their blacklist because they do not want to subscribe to it (normally, a user is not required to put a spammer in the blacklist because the trust value of the spammer will be reduced by default as described in Section 3.2). At time $t = 2$, a spammer calls legitimate user $L1$ who has $L2, L3, L4$, and colluder C in his buddy list. $L1$ searches a trust path and finds that the path from C produces the highest one. Then, it uses this trust as the inferred trust for the calling spammer. Even though the trust values from other legitimate users' blacklists are also estimated (a trust value for a blacklisted caller is zero), they are definitely useless in this case because the system selects only the highest trust path.

Identifying this attack is difficult. Instead of trying to identify cliques in a graph representing identities and their relationships, a detection systems need to

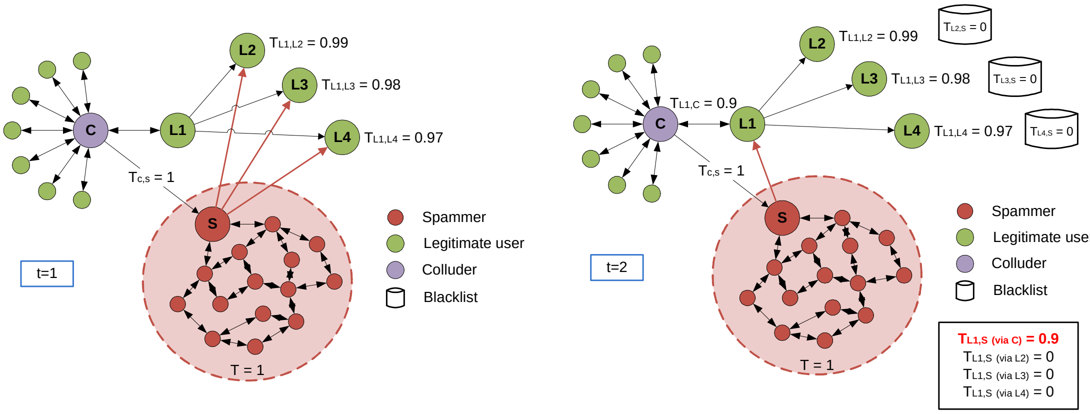

Figure 3.14. Malicious spies attack

identify partially connected clusters where each colluder may not appear to be connected to every other colluder due to the differing behaviors of the different roles in the strategy. In fact, it is possible that two colluders may have no direct interaction that is observable by the system, and thus appear completely separated when they are actually colluding indirectly. For example, the two colluders may report spam feedback to each other in order to hide their relationship. This is the most dangerous attack and the most difficult to tackle regarding the proposed system. However, like previous attacks, to conduct this attack requires an extra cost for generating legitimate colluders and a strong spam community.

3.6.4 General Discussion Regarding the Algorithm

There are two possible ways that a user will receive a SPIT: (1) a trust value of a spammer cannot be computed, and (2) a trust value of a spammer is incorrect. The former situation occurs when a spammer just enters the network. We do not have information about him at the first few periods of the attack. The latter situation can possibly occur due to the changes of calling behaviors. The operator has to investigate which trust element in the graph needs to be adjusted. Unfortunately, in reality, there is a huge relationship graph in the network. Therefore, the appropriate solution for the trust adjustment could be a collective effort. For instance, some customers received SPIT from the same spammer. This information is helpful to identify a common segment that needs to be recalculated.

Therefore, the operator needs to provide a method for a customer to report the feedback of the classification. After receiving a report and investigating a root cause of the misclassification, the operator has to reconfigure the detection system and recalculate trust values to avoid problems arising in the future. There are two methods that can be followed to adjust a trust value in this system. One is by adjusting thresholds: SPIT threshold and Social reliability threshold. The other is by resetting a trust value to the initial value as described in Section 3.3.

Another discussion is about the memory full scheme. Assuming one is marked as a spammer by many other nodes and there is no any previous calling activity, no one calls to this user. This user may not be able to make a call because he is classified as a spammer. It is a serious problem that affects the telecommunication service. Therefore, to avoid this problem, an operator should provide the ability for an affected user to challenge the justification on why the communication was identified as SPIT. For instance, this user will be enforced to verify himself before making a call through an audio CAPTCHA for a time period. After passing a justification period, his trust score will be reset.

3.7. Summary

The low cost and the flexibility of VoIP is rapidly attracting new subscribers and enabling innovative services. It also provides, however, a powerful tool for conducting unwanted communication, such as SPIT. I propose a trust-based system that would detect SPIT based on call duration, call direction, and social reliability. Since selecting only one trust path does not reflect all the trust information of a caller, I apply the Dempster-Shafer theory to aggregate the trust paths from a callee to an unknown caller in order to compute an inferred trust. The experimental results show that detection accuracy of the DST with the SR approach can be maintained at a high level even if the number of spammers increases. The false alarms are reduced significantly comparing with the another study. Also, this approach can be deployed in a real VoIP network because the results show that the characteristics of a network do not affect the detection performance. In addition, this system can detect SPIT before a call is received and does not require any interaction between caller and callee. These results show that my approach meets all the basic SPIT detection requirements.

Chapter 4

Anomaly-based SIP Flooding Attacks Detection

The IP telephony is constantly threatened by malicious software, fraud, or privacy violations. In this chapter, I focus on Denial-of-Service (DoS) attacks as a specific security problem which can force a service provider to reduce its capabilities or to go out of service totally. DoS attacks aim at denying or degrading a legitimate user's access to a service, or at bringing down the servers offering such services. Several possibilities exist for an attacker to cause DoS in a VoIP infrastructure, as described in Chapter 2. A SIP flooding attack is a severe DoS attack type that is the focus of this work. An attacker launches brute force attacks by generating a large number of SIP messages, trypically useless calls, in order to exhaust the resources of a server. Due to the time-sensitive nature of the Voice over Internet Protocol (VoIP) and IP Multimedia Subsystem (IMS) applications, a defense mechanism must not introduce noticeable timing delays.

As a solution, to protect a system against this attack, I present a statistical anomaly-based SIP flooding attacks detection system. The proposed system quantifies the correlations between SIP attributes in order to detect an anomalous transaction. In general, it learns and quantitatively tracks such relationships among chosen attributes of SIP packet, and raises an alarm for observed significant deviations, which alert an onset of a flooding attack.

Three statistical algorithms will be presented to monitor a significant deviation of SIP traffic and identify an anomalous event. The first is an application of Tanimoto Distance (TD), which is used to measure a dissimilarity of selected SIP attributes. The reasons of TD selection are as follows:

1. it is a simple computation method and can adapt to traffic changes, and therefore fits the dynamic environment of IMS, and
2. it is based on the proportion of the protocol attributes.

Second, an adaptive threshold is introduced to correctly keep track normal traffic. In fact, a dynamic setting of threshold will make an attack harder to evade. I modify an Exponential Moving Average (EMA) to compute the dynamic threshold based on the TD observed during the previous time intervals. Unfortunately, an attacker may hide an attack from an adaptive threshold by gradually increasing flooding packets. Therefore, I introduce a Momentum Oscillation Indicator (MOI) in order to detect such changes in the traffic.

The proposed method has a number of virtues as follows:

1. It does not require prior knowledge about the normal activity of the target system; instead, it has the ability to learn the expected behavior of the system from observations.
2. Statistical methods can provide accurate notification of malicious activities occurring over long periods of time.
3. Since services in VoIP and IMS are real-time applications, the detector should be efficient enough to perform in real-time. Based on a strong theoretical foundation, these algorithms can exhibit satisfactory performance over various attack types, without necessarily being complex or costly to implement.

Intrusion Detection System (IDS) researches have been evolving to create robust and effective technologies that will be able to classify activity in a system at an acceptable success rate. Having detected signs of security violations, IDSs trigger alerts to report them. These alerts are presented to a human analyst, who evaluates them and initiates an adequate response. However, a common issue of almost IDSs is the huge number of false alerts they produce. Generally, IDSs produce too many alerts compared to the size of the system they protect. It produces thousands of alerts per day, whose analysis may require excessive effort by a single network administrator [28]. Besides, many of these alerts are usually false ones; most of which are mistakenly triggered by benign events, i.e., False Positive (FP). This makes it extremely difficult for the analyst to correctly identify alerts related to attacks, i.e., True Positive (TP).

According to National Institute of Standards and Technology (NIST)'s IDS security issues report, anomaly detection systems are vulnerable to false positives [29]. DoS detection mechanisms, which aim at detecting floods, mainly look for sudden changes in the traffic and subsequently mark them as anomalous. However, they may produce false positives easily. The rationale behind detection methods is an assumption that the proportion between certain parameters remains roughly uniform as long as traffic is normal. A main drawback of detecting changes in the traffic volume is that the detection accuracy may be degraded if the legitimate volume is suddenly increasing. This phenomenon happens easily and frequently in telecommunications, such as during some flash events. For instance, cellphone networks were overwhelmed after the terror attacks in Boston [30]. In such event, the system cannot respond to all incoming requests. This traffic leads to significant changes in the distance between the current and previous traffic measurement which causes FPs in any anomaly-based attack detection system. These false alarms also occur in the proposed system as shown in the evaluation section of Chapter 4. The false alarm is one of the most important problems because it causes a loss of confidence in the alerts of a security framework. In some systems, the task of filtering and analyzing alerts is done manually, but that constitutes an overload of work for any security administrator. Therefore, we need a way to confirm that a real attack is taking place before raising any alert.

Many methods have been proposed in order to produce a more qualitative alert set. Some of them propose different configurations of detections, while most of them propose the post-processing of alerts. Usually, with anomaly-based detection, the abnormality is determined by measuring the difference between the suspicious activities and the norm. It is also based on a chosen threshold the observed behaviour is classified. In general, a threshold tuning is the most widely-used method for false detection reduction. Reducing the threshold directly induces more false alarms, while many of them are actually not true. Increasing the threshold can reduce the number of false alarms, but such an action causes the detection to be unable to detect major attacks. This is, in fact, the trade-off between reducing false alarms and maintaining system security.

In this work, I first address the problem of false positives by integrating the trust model to the DoS attacks detection algorithms. A trust model is applied to filter out a legitimate call after a probability of incoming traffic is significantly higher than its adaptive threshold. To calculate a reliable trust score, I use the call duration and its direction of each user to distinguish a legitimate user from a malicious user, as described in Chapter 3. This trust value can be used to construct the reliable social linkage with other users in the network through the trust inference mechanism. The social reliability, which is the evaluation of a user's behavior up to now, is also considered. A caller who conducts calling activities like a human will have a high trust value and a social reliability value. The system classifies this call as a legitimate call. If the average of trust score and social reliability of all callers in the testing phase is greater than the thresholds, the system will not raise the alarm even though the distance between the training and testing phases is high.

The rest of the chapter is as follows. Section 4.1 briefly discusses two main type of detection techniques and some SIP flooding attacks detection works. This section also includes several works that deal with false reduction in IDS. Three main statistical algorithms will be presented in Section 4.2 - 4.4. Section 4.5 demonstrates the detection architecture. An evaluation and its results are discussed in Section 4.6. The false reduction methodology and its evaluation are explained in Section 4.7 and 4.8, respectively. Finally, Section 4.9 discusses some issues about the proposed technique.

4.1. Related Work

4.1.1 Signature-based vs Anomaly-based Detections

The literature has described various forms of DoS handling strategies. There is no unique solution that is able to cover all types of attacks [31]. Depending on the type of analysis carried out, IDS are classified as either signature-based or anomaly-based. Signature-based (or misuse-based) systems rely on pattern recognition techniques where they maintain the database of signatures of previously known attacks and compare them with analyzed data. An alarm is raised when the signatures are matched. Therefore, these systems provide very good detection results for specified well-known attacks. However, they are not capable of detecting new, unfamiliar intrusions, even if they are built as minimum variants of already known attacks. On the other hand, if normal traffic has a similar signature to an attack, it will be falsely identified as an attack, and thus raises a false alarm. Also, a signature-based detection system needs regular updates to keep up with the latest attack patterns [32]. Therefore, a signature based IDS can be said to be reactive, because the occurrence of an attack will predate the systems' detection and following action. New ways of exploiting the computer networks are being invented every day. Moreover, there are obviously many ways of circumventing this signature. Since several protocols are used in an IMS, there is a possibility that flooding attacks may be generated by any combination of protocols. An intelligent attacker can always develop attacks that remain undetected by the signature-based systems. A number of researchers have argued that it is not difficult for an attacker to evade a signature [33]. SIP is also vulnerable to network anomalies that can be easily mounted by utilizing various SIP traffic generators openly available on the Internet.

On the contrary, anomaly-based systems attempt to estimate the normal behaviour of the system to be protected. Basically, the anomaly-based detection method consists of two parts: a training phase and a testing phase. During the training phase, traffic is monitored with no attacks assumed. During the testing phase, observed traffic is compared with what was measured during the training phase. It generates an anomaly alarm whenever the deviation between a training phase and a testing phase exceeds a predefined threshold. The key value of an anomaly-based detection system is that it can automatically infer attacks which are yet unknown, such as the polymorphic packet flooding attacks, and therefore undetectable by signature-based methods. Then, novel attacks can be detected as soon as they take place. This will alert the network administrator early and potentially reduce the damage caused by the new attack. However, anomaly-based systems require a training phase to construct the database of a normal behavior and a careful setting of threshold level of detection makes it complex. A model that specifies legitimate traffic too tightly will generate many false positives whenever traffic fluctuates. On the other hand, a loose model will let many attacks go undetected. So the challenge for an anomaly-based detection system is finding the right set of features and a model approach that strikes a balance between mistakenly identifying legitimate traffic as attacks, or false positives, and failing to detect real attacks, or false negatives.

The Internet and its applications are exposed to an increasing number of security threats. With new types of attacks appearing continually, developing flexible and adaptive security oriented approaches is a severe challenge. In this context, anomaly-based detection techniques are a valuable technology to protect IP telephony systems and networks against malicious activities. Therefore, my detection system developed in this work adopts the anomaly-based approach.

4.1.2 SIP-based DoS Attacks Detection Review

In this section, statistic anomaly-based IDS techniques are reviewed. In these techniques, the network activity is captured and a profile representing its stochastic behaviour is created. Two datasets of network traffic are considered during the anomaly detection process: one corresponds to the currently observed profile over time, and the other is for the previously trained statistical profile. As the events occur, the current profile is determined and an anomaly score is estimated by comparison of the two behaviours. The score normally indicates the degree of irregularity for a specific event, such that the intrusion detection system will flag the occurrence of an anomaly when the score surpasses a certain threshold.

The detection algorithms based on a non-parametric Cumulative Sum (CUSUM) have been applied in [34] and [35]. The CUSUM algorithm belongs to the family of change point detection algorithms that is used for detecting changes in a statistical distribution between two hypotheses. They observe the difference between the number of call setup requests (`INVITE` messages) and successfully completed handshakes (`200 OK` reply messages). In normal traffic, these two types of messages should be equal at any given time. So when this ratio unexpectedly changes, it indicates a flooding attack.

Reynolds and Ghosal describe a multi-layer detection scheme against DoS attack in VoIP [36]. They use a combination of sensors located across the network, continuously estimating the deviation from the long-term average of the number of call setup requests and successfully completed handshakes. However, the target of this method is only to protect the end user devices. They do not consider the core entities in the VoIP infrastructure in their scheme.

Sengar *et al.* present the VoIP Flooding Detection System (vFDS) for detecting anomalies in Session Initiation Protocol (SIP) traffic [37]. The Hellinger Distance (HD) is used to measure abnormal deviation in VoIP packet streams. Traffic is divided into two sets and the dissimilarity between these sets is measured. The HD scheme has shown a strong flooding detection ability because low-rate flooding is likely to have a probability distribution that is different from that of normal traffic. However, an attacker can subvert this approach by only slightly increasing the attack traffic. Furthermore, it does not address how to maintain an accurate threshold during attacks as described in Hecht's work [38]. This problem increases the likelihood of other attacks remaining undetected. Tang *et al* propose a similar approach to overcome the limitations of the previous schemes by using a sketch-based algorithm [39].

M. A. Akbar *et al.* compares three well-known flooding attacks detection algorithms: Adaptive threshold, CUSUM, and HD [40]. Their results show that HD algorithm outperforms the others. HD has a better detection accuracy and robust to variations in benign and attack traffic patterns. Therefore, in the evaluation section, I will compare the proposed method with the HD-based method.

4.1.3 False Positive Reduction Solutions

Several methodologies have been applied to reduce false alarm problem in IDS. Abimbola *et al.* [41] reconfigured the IDS in order to produce less false alerts. However, the static configuration is not enough to be applied for reducing false in a dynamic traffic of IMS network.

Pietraszek *et al.* [42] applied data mining techniques to train classifiers. This method requires a human intervention. The human analyst must either recognize root causes of alert, or label past alerts in order to train a classifier. The similar work proposed by Tian *et al.* [43], which an adaptive classifier was used. Human expertise is required to format frequent item sets and association rules to generate a knowledge-base of filtering rules, that can discard false alerts. This human interaction makes it difficult for an IDS platform to be effective.

Neural networks are used in some works such as [44]. This method requires a training on labeled alerts in order to be able to reduce FPs. This affects the Quality of Service (QoS) of the real-time application because of the delay.

4.2. Traffic Deviation Monitoring

4.2.1 Distance Formulas

At first, I focus on the use of the similarity or the distance measurement formulas to detect a deviation in a traffic. The basic idea is that they measure the dissimilarity between two probability distributions. However, in the probability theory, there are several distance measurement formulas. Many researches, [37, 38, 39], apply the Hellinger distance algorithm in their detection, but they did not provide any reason why they selected this formula. Therefore, I begin the research by evaluating some formulas, as shown in Table 4.1. My evaluation covers six distance measurement families as indicated in the table below.

Table 4.1: Distance Measurement Family

Minkowski family	
City block	$d_{CD} = \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$
Chebyshev	$d_{cheb} = \max_i \lvert P_i - Q_i \rvert$
L_1 family	
Sorensen	$d_{sor} = \dfrac{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{d} (P_i + Q_i)}$
Kulczynski	$d_{kul} = \dfrac{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{d} \min(P_i, Q_i)}$
Canberra	$d_{can} = \sum_{i=1}^{d} \dfrac{\lvert P_i - Q_i \rvert}{P_i + Q_i}$
Intersection family	
Intersection	$d_{is} = \sum_{i=1}^{d} \min(P_i, Q_i)$

Tanimoto	$d_{tani} = \frac{\sum_{i=1}^{d}(\max(P_i, Q_i) - \min(P_i, Q_i))}{\sum_{i=1}^{d} \max(P_i, Q_i)}$
Fidelity family or Squared-chord family	
Bhattacharyya	$d_{bhat} = -\ln \sum_{i=1}^{d} \sqrt{P_i Q_i}$
Hellinger	$d_{hel} = \sqrt{2 \sum_{i=1}^{d} (\sqrt{P_i} - \sqrt{Q_i})^2}$
Squared family or χ^2 family	
Squared Euclidean	$d_{se} = \sum_{i=1}^{d} (P_i - Q_i)^2$
Squared χ^2	$d_{SqChi} = \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{P_i + Q_i}$
Divergence	$d_{div} = 2 \sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{(P_i + Q_i)^2}$
Shannon's entropy family	
Kullback-Leibler	$d_{KL} = \sum_{i=1}^{d} P_i \ln \frac{P_i}{Q_i}$
K divergence χ^2	$d_{Kdiv} = \sum_{i=1}^{d} P_i \ln \frac{2P_i}{P_i + Q_i}$
Jensen-Shannon	$d_{JS} = \frac{1}{2} \left[\sum_{i=1}^{d} P_i \ln \left(\frac{2P_i}{P_i + Q_i} \right) + \sum_{i=1}^{d} Q_i \ln \left(\frac{2Q_i}{P_i + Q_i} \right) \right]$

I conduct the preliminary evaluation by using the simple legitimate and attack traffic. (More details of the realistic evaluation will be described in Section 4.6) Figure 4.1 shows the average FP of each formula. From this result, Tanimoto distance produced the lowest FP. Hence, I apply this formula in my traffic monitoring algorithm.

4.2.2 Tanimoto Distance

In probability theory, a TD is used to measure the difference between two probability distributions [45]. To compute the TD, let P and Q be two probability distributions in the same sample space where P and Q are N-tuples $(p_1, p_2, \ldots, p_n)$

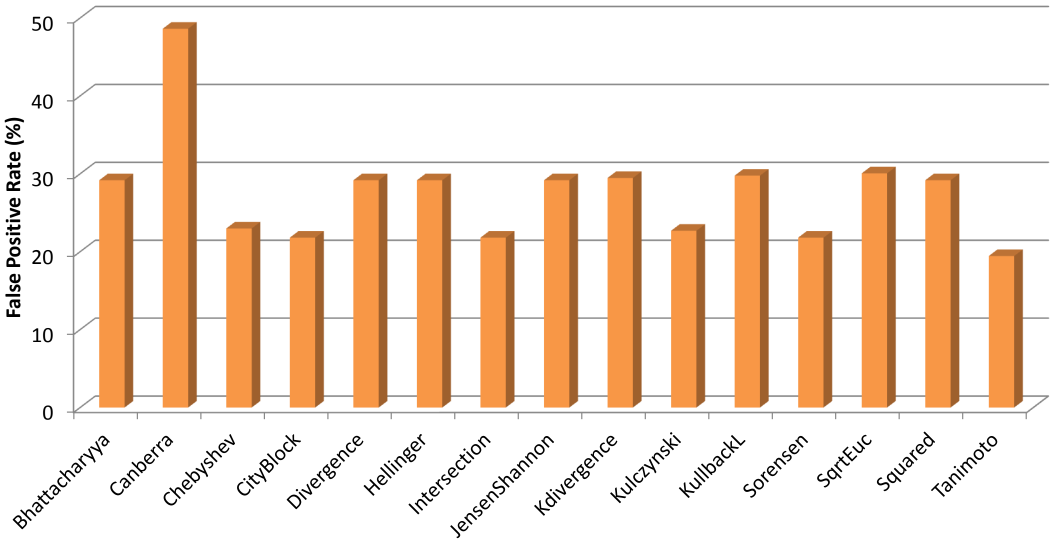

Figure 4.1. The average false positive of each distance formula.

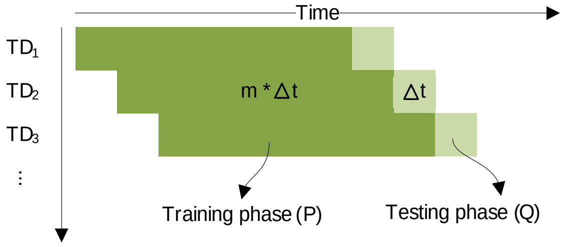

Figure 4.2. Training phase and testing phase in the data stream.

and $(q_1, q_2, \ldots, q_n)$. Then, the TD between P and Q is defined as

$$TD(P,Q) = \frac{\sum_{i=1}^{k} [\max(p_i, q_i) - \min(p_i, q_i)]}{\sum_{i=1}^{k} \max(p_i, q_i)}. \tag{4.1}$$

If the two probability distributions are totally different, TD approaches 1. This property provides a good approach to quantify the similarity of two data sets.

For the proposed flooding attack detection method, the traffic is divided into two portions: training and testing phases. The training phase, P, is assumed to be the probability distribution of a set of five SIP message types (`REGISTER`, `INVITE`, `200 OK`, `ACK`, and `BYE`) as determined during a training phase of length $m * \Delta t$ timeslots. With the testing phase, Q, corresponds to the probability distribution measured during a Δt timeslot, as shown in 4.2. Therefore, the 5-tuples of P and Q are $(p_{reg}, p_{int}, p_{ack}, p_{ok}, p_{bye})$ and $(q_{reg}, q_{int}, q_{ack}, q_{ok}, q_{bye})$ respectively where p_{int} is the number of `INVITE` messages divided by the total number of five SIP message types in the training phase. The initial training phase is assumed to be free of any attacks and acts as a basis for comparison with the testing phase. Next, I measure the distance between these two phases, i.e., TD_1. A low TD value means that there is no significant deviation between the two probability distributions. Meanwhile, high TD indicates that there are numerous open connections, which are not closed in proper time. Then, it is implied that anomalies occurred in the traffic and altered the distributions. After measuring the distance, if the distance does not exceed a threshold, a portion of the training phase will be merged with the testing phase to construct the next training phase, i.e., TD_2. This sliding window function helps the training phases to adapt to the dynamics of network traffic during a real-time analysis.

4.3. Adaptive Threshold

Most flooding attack detection methods face difficulty when determining the threshold for detection. It is hard to set an appropriate threshold value for a real-time communication scenario. In particular, the IMS service traffic pattern will change over time. For example, calling traffic during the night may be less than during the day. A sudden increase in traffic can also occur, e.g., hot breaking news can cause a rapid increase in communication. These conditions are not necessarily caused by a DoS attack and need to be taken into consideration when setting a threshold. Therefore, a static threshold is neither practical nor responsive to expected normal traffic in this case. To accurately track normal traffic, I use an adaptive threshold in this system.

To deal with the fluctuation of IMS traffic, an adaptive threshold is based on an estimate of the mean deviation of the selected SIP packets computed from recent traffic measurements. In statistics, a moving average is widely used in time series analysis for predicting a future data set by using current and previous data sets. I preliminarily apply the Exponential Moving Average (EMA) for computing a distance threshold for the next time interval. Unlike the Simple

Moving Average (SMA), EMA gives more weight to the latest data, which is suitable for the IMS traffic environment. From Eq. (4.2), let D_t and D_{t-1} be estimated averages of the current and previous distances between two probability distributions: the training phase P and testing phase Q and where d_{t-1} is the previous distance. The coefficient α is a smoothing factor where $0 \leq \alpha \leq 1$. Using a small α, it can detect small changes, and a larger value for detecting larger changes. Alternatively, α may be expressed in terms of n time periods, where $\alpha = \frac{2}{n+1}$. For example, if one wants to calculate the EMA for the last 14 periods, n is equal to 14.

$$D_t = \alpha d_{t-1} + (1-\alpha) D_{t-1} \tag{4.2}$$

Many anomaly traffic detection systems, such as [34], apply EMA as a threshold. However, because it utilizes only one single coefficient, EMA is not effective if there is a trend in the time series data [46]. Therefore, I add a trend forecast in the EMA as shown in Eq. (4.3) - Eq. (4.5). γ is the trend smoothing factor where $0 \leq \gamma \leq 1$. Equation (4.3) adjusts D_t directly for the trend of the previous period, b_{t-1}, by adding it to the last estimated distance value D_{t-1}. This helps to eliminate the lag and brings D_t to the appropriate base of the current value. Equation (4.4) updates the trend, which is expressed as the difference between the last two values. This equation is similar to the basic form of EMA. Note that there are several methods to choose the initial value of b_1, e.g., $b_1 = D_2 - D_1$. Finally, the adaptive threshold can be calculated by Eq. (4.5). I add k times N-period standard deviation, σ, of the forecast values $(D_t + b_t)$ to reduce the false alarms. This parameter influences the threshold value. Hence, an operator can tune this value to achieve desirable detection accuracy.

$$D_t = \alpha d_{t-1} + (1-\alpha)(D_{t-1} + b_{t-1}) \tag{4.3}$$

$$b_t = \gamma (D_t - D_{t-1}) + (1-\gamma) b_{t-1} \tag{4.4}$$

$$TD_t^{\text{threshold}} = (D_t + b_t) + k\sigma \tag{4.5}$$

4.4. Momentum Oscillation Indicator

An attacker can potentially subvert an adaptive threshold if he knows the legitimate traffic intensity. In this scenario, during the first few periods, the attacker sends a very low intensity attack to the target server. This malicious traffic does not impact the threshold because there is no significant deviation in the traffic. Next, he increases the attack rate slightly, which still does not affect the deviation of the overall traffic significantly. As the traffic increases, the adaptive threshold is updated. Finally, the attacker can send large malicious traffic that can block the IMS server without ever being detected because the traffic is below the current threshold. Thus, to detect such attack patterns, I propose a MOI that is used to measure the change of signal movements.

To simplify the explanation of the calculation, this indicator divides the traffic intensity into two sides: upside and downside. They can be classified by the median of the signal over n periods. Upside means that the traffic intensity at that time is above the median while downside means it is lower the median. The very first calculations for average upside and downside are simple n period averages, as computed by Eq. (4.6) and Eq. (4.7). The second and subsequent calculations are based on the prior and the current upside and downside, as computed by Eq. (4.8) and Eq. (4.9). This is the same concept that EMA uses for comparing the prior value with the current value. This also means that MOI values become more accurate as the calculation period extends. Finally, in Eq. (4.10), the result is normalized and turned into an oscillator value that fluctuates between 0 and 100. The normalization step makes it easier to identify extremes because MOI is range bound. MOI is 0 when the *avgUp* equals zero. Assuming $n = 20$, a zero MOI value means the number of `INVITE` packets moved lower in all 20 time periods. The MOI is 100 when the *avgDown* equals zero. This means that the number of packets moved higher during all 20 time periods. Lastly, there were no downsides to measure.

$$\mathrm{Up}_1 = \frac{\sum_{i=t}^{t-n} \mathrm{Up}_i}{n} \tag{4.6}$$

$$\mathrm{Down}_1 = \frac{\sum_{i=t}^{t-n} \mathrm{Down}_i}{n} \tag{4.7}$$

$$\mathrm{avgUp}_t = \frac{(\mathrm{Up}_{t-1} \times (n-1)) + \mathrm{Up}_t}{n} \tag{4.8}$$

$$\mathrm{avgDown}_t = \frac{(\mathrm{Down}_{t-1} \times (n-1)) + \mathrm{Down}_t}{n} \tag{4.9}$$

$$\mathrm{MOI}_t = 100 - \frac{100}{1 - \frac{\mathrm{avgUp}_t}{\mathrm{avgDown}_t}} \tag{4.10}$$

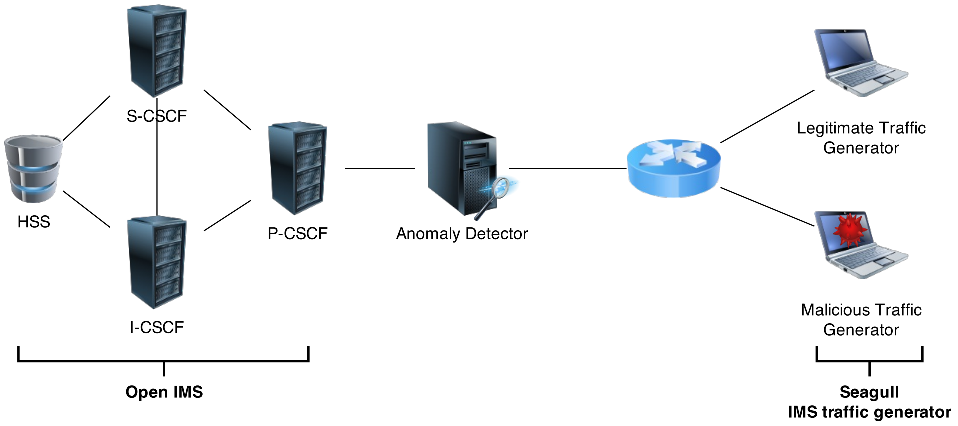
S-CSCF
HSS
P-CSCF
Anomaly Detector
I-CSCF
Legitimate Traffic Generator
Malicious Traffic Generator
Open IMS
Seagull IMS traffic generator

Figure 4.3. Testbed topology.

In this work, I consider the traffic as behaving in an anomalous way when MOI is above 80%. This level can be adjusted to better fit the traffic environment. The previous median before an attack is stored and used as the median for the current state. This median value is kept until MOI falls below the limit level again. The alarm is raised when the momentum is greater than the desired level over the predefined time period. It means that the current traffic is continuously increasing during the observation periods.

4.5. DoS Attacks Detection System Architecture

Under the assumption that most of the attacks come from outside the operator network, to analyze all incoming traffic, therefore, the detection system is located at the perimeter of the IMS as shown in Figure 4.3. The Open IMS server and attack detector are running on 1.86 GHz Intel Xeon processor with 8 GB of memory. Figure 4.4 shows the overview of the detection system. Before starting the detection system, the number of timeslots m and the duration ΔT of the training and testing phases must be defined. Increasing the time span of the training and testing phases will increase the number of packets in the analysis. Since our proposed system quantifies the correlations among SIP attributes, the number of packets during a learning process does not highly affect the detection

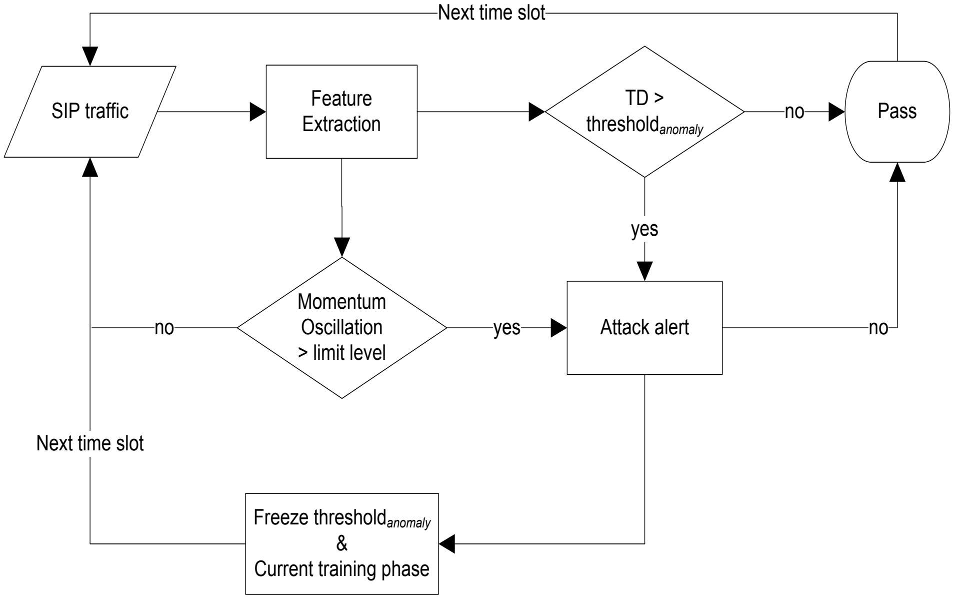
Next time slot
SIP traffic
Feature Extraction
TD > $\text{threshold}_{anomaly}$
no
Pass
yes
Momentum Oscillation > limit level
no
yes
Attack alert
no
Next time slot
Freeze $\text{threshold}_{anomaly}$ & Current training phase

Figure 4.4. System overview.

performance. However, there is a trade-off between a longer and a shorter time span because of the response time of each packet. A longer time span can accurately return distance values, while shorter time spans can detect small changes. We assume that the first few training phases are free from any malicious traffic. The incoming SIP traffic is extracted into five SIP message types. The distance between the training and testing phases are computed by TD algorithm and then compared with the adaptive threshold. If the distance does not exceed the threshold, the next training phase and testing phase are continuously evaluated, and so on. However, when a flooding attack comes, it will disturb the probability distribution obtained from the testing phase. Thus, the distance will exceed its threshold. When this happens, the system raises the attack alert. Then, the system will keep the current training phase and anomaly threshold and only move the testing phases to the next time interval. As a result, the distance in the next cycle is evaluated between the stored training phase and the current testing phase. This freezing process will continue until the distance drops below the anomaly threshold. This can protect the threshold from being impacted by the flooding attacks and makes it stable during attack. The system will resume the normal process when the detected anomaly is no longer present. Consequently, the traffic during attacking periods will never be included in the training phase. Concurrently, the momentum oscillation of the traffic is also monitored. If it is higher than the desired level over the predefined time period, the alarm will be raised.

The traffic sampling technique is not applied in this work because I aim at analyzing all incoming traffic to the server and proposing a near real-time detection system. Since the lightweight statistical calculation is applied to analyze the data, the system does not require high computing resources. In contrast with other methods that require sampling the data, they need to keep the whole traffic in the memory before analyzing. This requires more computing resources and cannot be executed in real-time.

An alarm notifies the system administrator about a current attack attempt and anomalous activity. The system will generate a report that provides key information which can be used to identify the attack's origin such as caller id and attack time. This is helpful for the administrators to take the next action regarding this suspicious event. To protect the entire system, we need to incorporate our proposed system with other protecting systems or countermeasure mechanisms. In a practical deployment, it can be transparently interposed at a firewall and implemented as a loadable module of the firewall. This topic is not included in the scope of this work.

4.6. Detection Efficiency Evaluation

The accuracy of the flooding attack detection system is the ratio of correctly classified instances over the total number of instances. We focus on binary classification because it simplifies the analysis of a system's performance. According to the relation between the result of the detection for an analyzed event (normal and intrusion) and its actual nature (innocuous and malicious), a classification may fall into one of the following four categories:

- True Positive (TP) - an actual attack triggers a detection system to produce an alarm

Table 4.2. Wikipedia talk network characteristics.

Node	2,394,385
Edges	5,021,410
Average clustering coefficient	0.1958
Diameter (longest shortest path)	9

- False Positive (FP) - an event signaling a detection system to produce an alarm when no attack has taken place
- True Negative (TN) - no attack has taken place and no alarm is raised
- False Negative (FN) - a failure of a detection system to detect an actual attack

It is clear that low FP and FN rates, together with high TP and TN rates, will result in good efficiency values.

In this chapter, I calculate sensitivity and specificity rates to measure the performance of the proposed system. Sensitivity, Eq. (4.11), is the ratio of correctly detected flooding attacks to all actual flooding attacks. With higher sensitivity, fewer actual cases of attack go undetected. Specificity, Eq. (4.12) is the proportion of correctly detected legitimate traffic to all actual legitimate traffics. Higher specificity indicates that the system detects legitimate calls more accurately.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{4.11}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.12}$$

This section includes a description of the attack scenario, the system design, simulation information, and the performance evaluation of the detection system.

4.6.1 Dataset

Due to privacy concerns, I have not found any publicly available IMS traffic dataset. Therefore, I use Seagull [47], the IMS traffic generator, to synthetically generate the traffic. Seagull is an open source multi-protocol generator that can generate SIP messages and has the ability to simulate customized SIP scenarios. It is a powerful traffic generator for functional, load, endurance, stress and performance/benchmark tests for almost any kind of protocol adopted in IMS networks. So I used Seagull to generate both legitimate traffic and attack traffic. I used the Open IMS Core [48] to emulate an IMS system. It is an open source implementation of IMS Call Session Control Function (CSCF)s and a lightweight Home Subscriber Server (HSS), which together form the core elements of all IMS/NGN architectures as specified today in 3GPP. The four components are all based upon Open Source software, e.g., the SIP Express Router (SER) and MySQL.

Many researchers simulated their testbed by only generating a set of users and randomly choosing call parties, such as [38] and [39]. In order to simulate the real conditions of the IMS user network and make our testbed close to the real environment, I use the communication between users in the Wikipedia talk network [49] as the IMS users. Table 4.2 shows the characteristics of this social network. The network contains all the users and discussion from the inception of Wikipedia until January 2008. We select this dataset because it contains the communication direction among users and large enough for the evaluation. Initially, this dataset does not have positive and negative links among them. I assume that the nodes who have direct communication are friends in a buddy list. The path length between users does not affect the trust model as proved by the previous work [21].

The SIP session initiation of a legitimate call was synthetically generated with a Poisson distribution, which was the same as in the PSTN model [27]. The call duration in VoIP are heavy-tailed distributions that can be generated with Pareto distribution [27]. The mean number of calls per unit of time and call duration are defined according to the IP phone statistics from the operator [50]. These statistics were collected from April 1, 2011 to March 31, 2012. According to this data, call duration ranged between 104 and 215 seconds. The average call frequency in this study is around 200 calls per second. The call destination is selected either from a buddy list or another person who is not in a buddy list. The choice of this recipient is a Zipfian distribution [16].

The attacks are synthetically generated. This allowed the author to control the characteristics of the attacks, and hence be able to investigate the performance of the detection algorithms for different attack types. The experiment in this work considered both high and low intensity attacks, whose rates varied from 100 to 400 calls per second. This range was chosen to evaluate the effectiveness of our detection approach under both low rate and high rate attacks. The duration of the attacks was normally distributed with a mean of 60 seconds. I considered attacks whose intensity increases both abruptly and gradually.

The window size of a training phase and a testing phase are set by Δt. According to the SIP specification [10], an `INVITE` transaction timeout is 32 seconds. Consequently, to correlate an `INVITE` message with a response message, the sampling window size should be 32 seconds. However, our proposed method is not sensitive to per-flow information. Therefore, we set sampling size (Δt) equal to 10 seconds in order to achieve a high detection accuracy. Moreover, the distance measurement algorithm depends on the training phase window size, $m * \Delta t$. A longer training phase accurately returns the distance value, while shorter training phases can detect a small change. Then, we set the training phase to 40 seconds ($4 * 10$) in order to balance the responsiveness and the detection accuracy. Other parameters are set as $\alpha = 0.2$, $\gamma = 0.2$, $k = 2$, and the standard deviation is calculated from the last 20 time intervals. The parameter n of the MOI is 20. This value can be lowered to increase sensitivity or can be raised to decrease sensitivity. The trust threshold and SR threshold are 0.25. These are the optimal values in our environment.

4.6.2 Multiple Attack Intensity Rates

This experiment investigates the performance of the detection system without the trust model integration under multiple attack intensities. The mean amplitude of the low intensity attack in this experiment was 50% of the legitimate traffic mean rate. The attack traffic was injected every five minutes starting from the 598^{th} second with the low rate, 100 calls per second. The attack rates of next three floodings were 200, 300, and 400 calls per second, respectively. Figure 4.5 shows the results for the TDs and their adaptive thresholds. The horizontal axis started from 50^{th} second according to the length of the first training and testing phases. The learning period ended at 250^{th} second and then threshold calculation is initiated. The MOI rate during the attacks is shown in Figure

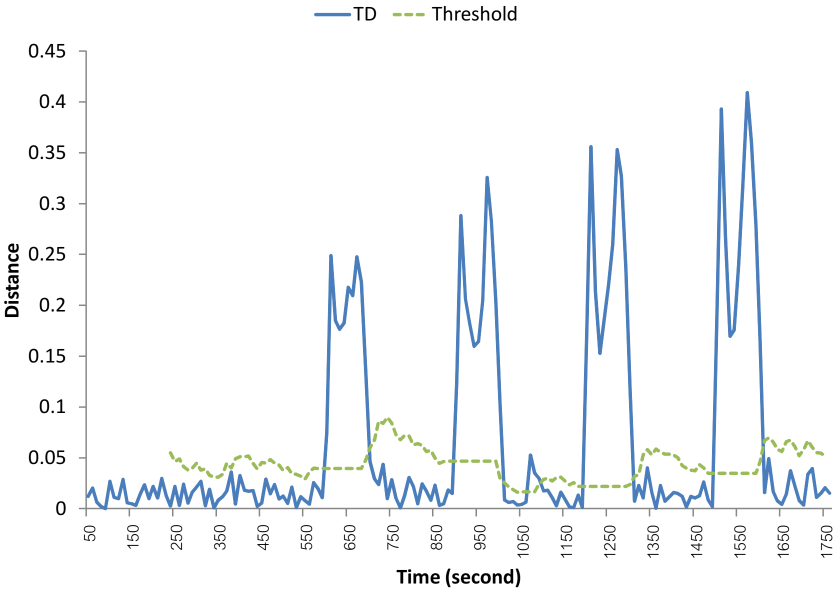

Figure 4.5. TDs and their adaptive thresholds during four flooding attack rates: 100-400 calls/sec.

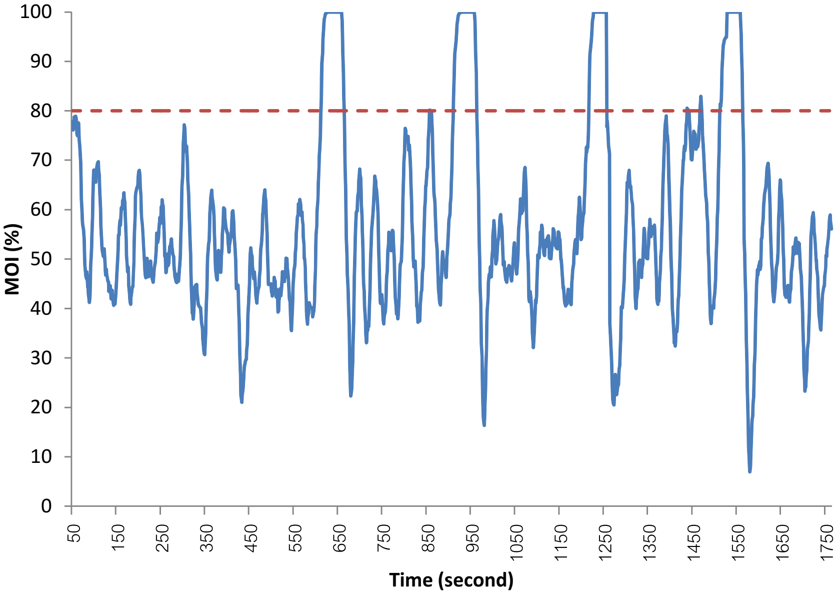

Figure 4.6. MOI during four flooding attack rates: 100-400 calls/sec.

4.6. The MOI threshold is 80% (red dash line). The first attack was detected at 600th second and at 614th second by the TD and MOI respectively. The highest attack intensity, around 400 calls per second, was injected at 1,498th second. This attack traffic is 200% of a legitimate traffic. The distance significantly deviated at 1,500th second. From these results, our system produced a very small detection delay. Moreover, according to these graphs, both low and high intensity attacks were detected accurately. The average sensitivity and specificity of the detection system in this experiment was 100% and 95.38% respectively. However, false positives occur when legitimate traffic suddenly increased, e.g., at 1060th second in the Figure 4.5. According to Eq. (4.12), this FP impacts to the specificity rate. I will introduce the false positive reduction method in Section 4.7.

4.6.3 A Gradually Increasing Attack Pattern

This experiment investigates the performance of MOI for detecting a gradually increasing attack detection. Figure 4.7 shows the SIP traffic including a gradually increasing flooding attack. The attack traffic was injected at time 601st with an attack rate of 1% of the normal traffic and then increased slightly until it reaches 500% of the normal traffic at time 1100th. This kind of attack can subvert an adaptive threshold technique, as explained in Section 4.4. However, as shown in Figure 4.8, the proposed MOI can accurately detect it. The MOI increased rapidly after time 601st and then reached the highest level at 678th. The MOI remained 100% until the SIP traffic intensity was decreased at time 1101st. This result shows that the MOI can detect such attack pattern correctly.

4.7. False Positives Reduction System Architecture

Figure 4.9 shows the overview of the detection system. This architecture is an extension from the previous system design described in Section 4.5, in order to enhance the detection performance. The similarity between a training phase and a testing phase is measured by a TD. If they are not different significantly, the system will continue the normal measuring process. If the TD value exceeds

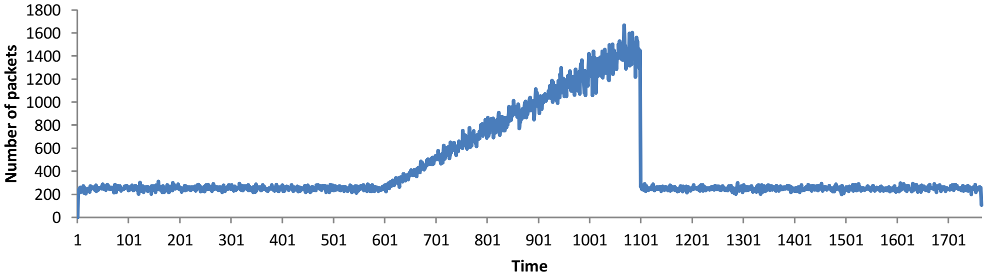

Figure 4.7. SIP traffic during a gradually increasing flooding attack.

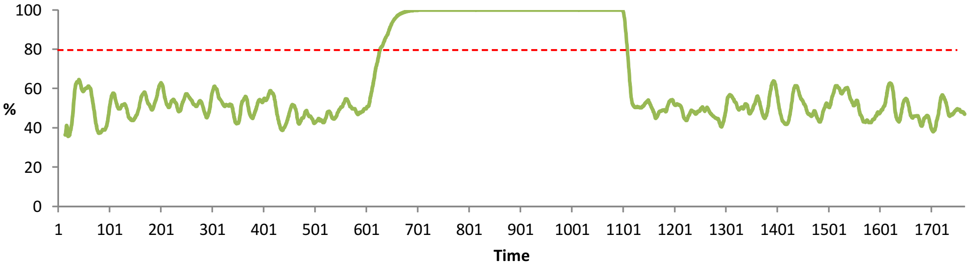

Figure 4.8. MOI of the traffic.

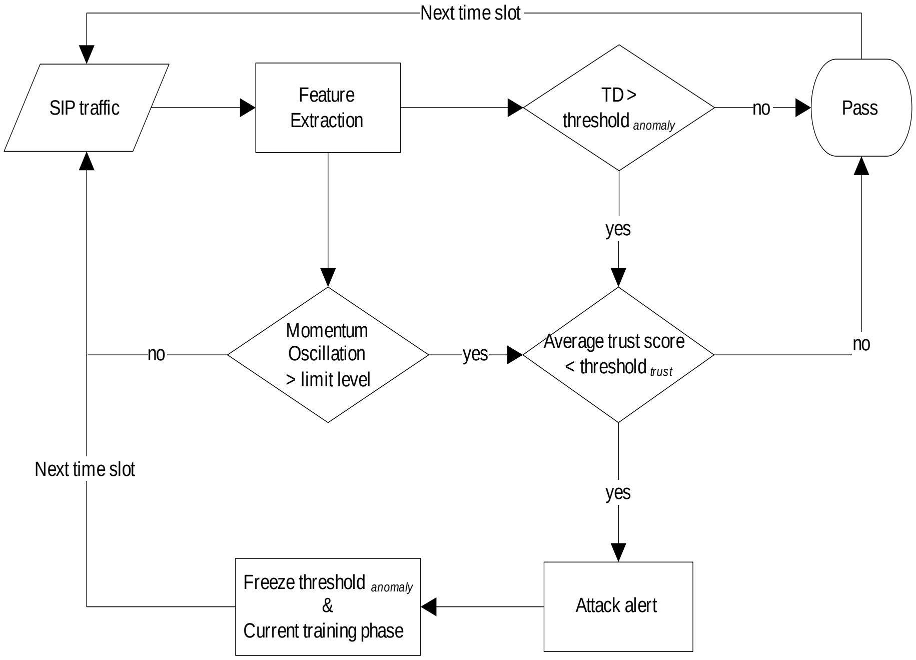
Next time slot
SIP traffic
Feature Extraction
TD > threshold anomaly
no
Pass
yes
Momentum Oscillation > limit level
no
yes
Average trust score < threshold trust
no
Next time slot
yes
Freeze threshold anomaly & Current training phase
Attack alert

Figure 4.9. System overview.

the anomaly threshold, the system will check the trust values and the Social Reliability (SR) values of all callers in the testing phase. These two variables are used to discriminate a legitimate caller from a malicious caller. From the expectation, if most of the calls are legitimate communications, the average trust score will be higher than the trust threshold. Therefore, the attack alarm will be raised when the average trust and SR value of the callers are less than the threshold. Then, the system will keep the current training phases and anomaly threshold, and only move the testing set to the next time interval. This freezing process will keep on until the distance drops below the anomaly threshold to protect the threshold from being impacted by the attacks. The traffic during attacking periods will never be included in the training phase. It is also able to protect the threshold from being impacted by the flooding attacks and makes it stable during attack. The system will resume to the normal process when the detected anomaly is over.

4.8. False Positives Reduction Performance

The performance metrics considered in this chapter include the accuracy rate and the False Positive Rate (FPR):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.13}$$

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{4.14}$$

The accuracy rate is the proportion of true results, both TP and TN. It is the degree of closeness of measurements of a quantity to its actual value. FPR is the probability of falsely detecting a legitimate event as malicious.

In order to validate the proposed system, I compare its performance with other available approaches. Therefore, this section shows the comparison results among the proposed system, the HD approach [37], and the CUSUM approach [34]. Furthermore, in the last experiment, I investigate how the testing window size affects the detection of false negatives.

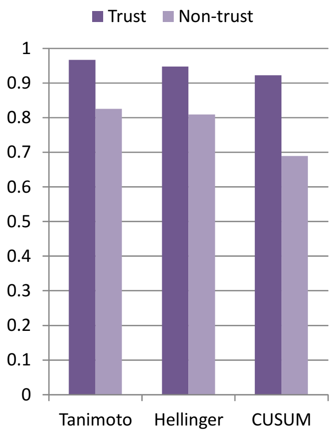

Figure 4.10. The accuracy rate between trust and non-trust approaches.

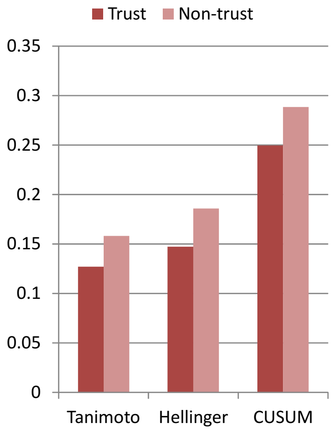

Figure 4.11. False positive rate between trust and non-trust approaches.

4.8.1 Performance Comparison with Hellinger Distance and CUSUM Algorithms

I conducted this experiment to investigate the performance of detection after integrating the trust model. I compared our proposed system with two well-known anomaly detection algorithms: HD [37] and CUSUM [34]. Figures 4.10 and 4.11 show the average accuracy rate and FPR between trust and non-trust integration approaches, respectively. After integrating the trust model, the average detection accuracy of TD, HD, and CUSUM algorithms were increased by around 14.17%,

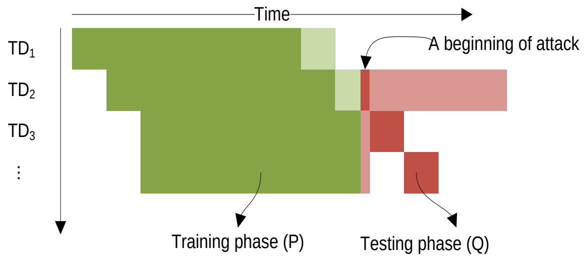

Figure 4.12. A FN may occur in a longer testing phase size.

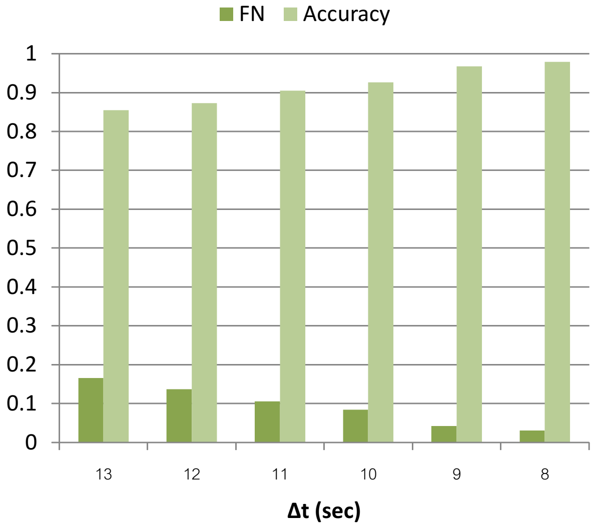

Figure 4.13. FN and accuracy rate among different testing phase sizes.

13.87%, and 23.3%, respectively. It can also be seen that the trust model integration method can reduce the FP in all flooding attack detection algorithms. According to Eq. (4.13), since false positives were reduced, the accuracy rate increases. It indicates that the trust model can classify a legitimate call correctly. Moreover, from these results, our proposed system produced the best detection performance compared to other methods in term of the highest accuracy and the lowest FPR.

4.8.2 The Impact of a Testing Phase Window Size

The above results were for specific values of the testing phase window size. Next I investigate the trade-off between the window size of testing phase and the detection accuracy. Generally, in an anomaly-based detection approach, a longer testing phase is set to obtain a stable distribution under normal conditions. However, an attack detection probability will be reduced because of an FN. This FN occurs when few portions of the testing phase contain an initial part of attack traffic, as shown in Figure 4.12. If this attack portion is not high enough to alter the distribution, then no any attack alarm is raised. If the testing phase is set to be short, many FPs will be raised. This is the general issue of a flooding attack detection system that uses training and testing phases. However, with our trust integration method, we can reduce FN while increasing detection accuracy by resizing the testing phase. Figure 4.13 shows the FN and accuracy rate among different testing window sizes of our detection system. FN were reduced when the testing phase window size was decreased. With our trust integration, FP were also reduced even though the size was decreasing. Consequently, the accuracy rate of the detection was improved. However, reducing a testing window size will consume more computing resources. Therefore, I suggest using the appropriate size that fits the operator's policies and resources.

4.8.3 Computational Overhead

This section presents the processing and memory overheads of the detection system. Figure 4.14 shows the percentage CPU consumption of the detector during many incoming traffic rates: 100 - 700 calls/sec. From these results, the CPU time of the detector depends on the number of incoming packets. If the traffic rate is increased, the detector needs more processor power to analyze them. Table 4.3 shows the average of CPU consumption of the detector and the percentage of failed calls of each traffic rate. The failed calls occur when the Open IMS server rejects calls due to a full memory. When I increased the traffic rate to 700 calls/sec, the failed calls reached 12.40% and then the Open IMS server crashed. At this point, the average of CPU time of the detector was around 76.70%. This implies that it was still capable to receive more traffic. It means that the pro-

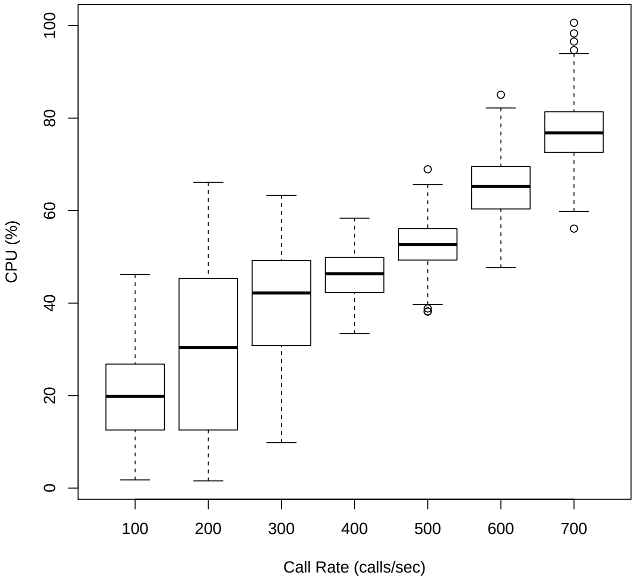

Figure 4.14. CPU consumption of different traffic rates: 100 - 700 calls/sec.

posed detector can work with the maximum call rate of the Open IMS system. However, I used only a single IMS server in this testbed. When in reality, an operator may use multiple servers to handle a huge incoming traffic. Therefore, for the deployment, the operator must place the detector in front of every server in order to monitor all traffic.

Figure 4.15 illustrates the memory consumption of the detector. The memory usage was rising when the traffic rate increased. However, due to the lightweight algorithms, the detector does not require a high memory for the computation. At the maximum call rate, 700 calls/sec, the detector spends memory around 17 MB.

4.9. Discussion

A challenge on this proposed technique is the performance comparison with a modern application layer session-based firewall. This firewall keeps track of the network connection sessions and holds significant attributes of each session in memory. Then, the CPU and memory resources are required for analyzing the session. Generally, in order to prevent the memory from filling up, sessions will

Table 4.3. CPU consumption of the detector and the rejected call rate by the Open IMS server

Call Rate (calls/sec)	Average CPU Consumption	Failed Calls (%)
100	19.52	0
200	30.28	0.37
300	39.70	3.39
400	46.21	6.33
500	52.65	7.81
600	65.01	8.42
700	76.70	12.40

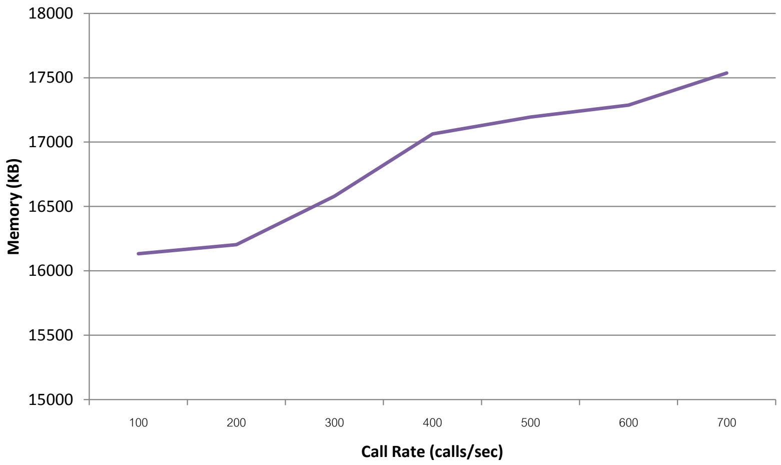

Figure 4.15. The memory consumption of the detector.

time out if no traffic has passed for a certain period. These state connections are removed from the memory. However, during an attack, more and more requests come in, which have to be processed by the firewall. As a consequence, the firewall cannot monitor every packet, so some of them are not recognized by the security system or must be dropped because of a lack of buffer capacity. Comparing with our statistical method, the detection system does not require a large memory to store a processed data. This can avoid a bottleneck problem when the detection system needs to process a massive traffic.

Generally, a trust-based system will be subverted if an attacker has a high trust value assigned by other nodes. Since the duration of an outgoing call is used to compute a trust value, an attacker needs some calls from other users frequently to maintain high trust values. At the same time, the attacker would have to maintain a balance in his in-out calling degrees in order to keep an appropriate social reliability level. In the VoIP system, this activity requires an extra cost that would make it counterproductive for the attacker's business. However, for the proposed model to be efficient, an attacker must not be able to steal the identity of a legitimate user. The trust model will be useless if an attacker can use the trust score of a legitimate user. This is the authentication issue of the IP telephony system that is out of the scope of this work. However, there are many works that have been proposed to fix this problem such as in [51], where an Identity Based Cryptography (IBC) is employed to enhance the security of the IMS authentication process.

4.10. Summary

In this chapter, I proposed an anomaly-based DoS attack detection system using three statistical algorithms: Tanimoto distance, an adaptive threshold, and a momentum oscillation indicator. The system extracts five types of SIP traffic messages and estimates the dissimilarity between them over time. The proposed system is multivariate model that considers the correlation of the chosen SIP packets. Therefore, an attacker needs to mimic the legitimate traffic with complete SIP transaction in order to subvert the proposed system. The adaptive threshold, developed from EMA, adds data trends to track the behavior of the traffic and make the system more accurate. I also proposed a MOI to detect a gradually increasing attack pattern. Performance evaluation on a testbed simulation showed that the proposed mechanism successfully detected various flooding attack patterns, including both low and high intensity attacks.

Additionally, I first present a trust-based model integration that is used to reduce false positives of a SIP flooding attack detection targeting IMS networks. I measure the accuracy rate and false positive rate after integrating the trust model to the proposed DoS attack detection system. Then, I compared the detection efficiency with two well-known anomaly detection algorithms: Hellinger Distance and Cumulative Sum. The mixed attack rates was used to evaluate the proposed method. From the result, we can see that the trust model can reduce false positives by more than 10%, as well as increase the detection accuracy in all algorithms. Moreover, the trust model can enhance the detection performance when resizing the testing phase's window size.

Chapter 5

Conclusion and Future Work

5.1. Thesis Conclusion

The low cost and the flexibility of Voice over Internet Protocol (VoIP) is rapidly attracting new subscribers and enabling innovative services. Along with the accelerated global deployment of IP telephony networks, their security problem has become increasingly serious. In this work, two security challenges are considered: Spam over Internet Telephony (SPIT) and Denial-of-Service (DoS) attacks.

To deal with SPIT, I have presented a trust-based system that would detect SPIT on the basis of call duration, call direction, and social reliability. A trust value based on call duration is calculated for each friend in the buddy list. This technique provides a simple way to use call duration as an automatically assigned trust value based on call behavior and human reasoning. Due to the reliability of this value, it is difficult for a spammer to subvert the system. To extend the detection scalability, I further proposed a trust propagation method in case a caller and a callee do not have a direct relationship. The data fusion technique, named Dempster-Shafer Theory (DST), is applied to aggregate the trust paths between an unknown caller and a callee in order to compute an inferred trust. The Social Reliability (SR), the evaluation of human behavior, is considered before connecting a call to a callee. According to these methods, the detection system is difficult to subvert by a malicious user. Based on realistic simulation results, the proposed technique can detect all SPIT completely and maintain a high level even if the number of spammers increases. The false alarms are reduced significantly compared with the highest trust path selection method.

These results show that my proposed method meets all the basic SPIT detection requirements. The sensitivity and specificity from the experiments show that the detection system can minimize the probability of blocking legitimate calls and maximize the probability of blocking spam calls. Since no modifications to VoIP clients or servers are required, we can readily deploy our voice spam filter as a pluggable security module on the any VoIP technologies. Moreover, the system does not require additional effort by a user because the trust value is automatically calculated in the background. Therefore, it is very convenient for all users. Also, this approach can be deployed in a real VoIP network because the results show that the characteristics of a network do not affect the detection performance. In addition, this system can detect SPIT before a call is received and does not require any interaction between caller and callee.

Undoubtedly, DoS attack also presents a serious threat to IP telephony networks. In this dissertation, I proposed a statistical anomaly-based DoS attack detection system, using the Tanimoto distance, an adaptive threshold, and a momentum oscillation indicator, to detect SIP flooding attack. The detection system extracts and calculates the probability of each Session Initiation Protocol (SIP) message. Next, the dissimilarity of probability distributions between a training phase and a testing phase is estimated. Because of the correlation of the chosen SIP packets, an attacker needs to mimic the legitimate traffic through complete SIP transaction in order to subvert this system. The modified moving average is computed as an adaptive threshold for tracking the behavior of the traffic and making the system more accurate. A momentum oscillator indicator is proposed to detect a special attack pattern, which is a gradually increasing attack. These statistical algorithms are stateless and require low computational overhead. Hence, the proposed system is a near real-time attack detection that is suitable for an IP telephony system.

Furthermore, I address the false alarm problem by integrating a trust model to filter out a legitimate call from suspicious traffic. The trust value of each user is computed from the call activities and human behavior of a user, including call duration, call direction, an interactivity ratio, and the diversity of calls. The SR, which is the evaluation of a users recent behavior, is also considered. A caller who conducts calling activities like a human will have a high trust value and SR value. The system classifies this call as a legitimate call. If the average of trust score and social reliability of all callers in the testing phase is greater than the thresholds, the system will not raise the alarm even though the distance between the training and testing phases is high.

Performance evaluation on the testbed simulation showed that our detection system successfully detected various flooding attack patterns. The average accuracy rate was higher than 90%. Furthermore, the false positives were reduced after using the trust model. Lastly, the experimental results showed that decreasing a testing phase's window size can improve the detection performance while simultaneously reducing the false negatives.

5.2. Future Work

5.2.1 System Enhancement

Together with the introduced related works, the method proposed in this work is a basic step for protecting an IP telephony network against current SPIT and DoS attacks. Especially, when the attacks become more sophisticated, the research work is still in its infancy. The research is needed to enhance in many directions. The following lists are the example of the outlook of this work.

Trust Path Query

Since the number of VoIP subscribers have been dramatically growing, the time for finding a relationship between a caller and a callee in a large population of users may be increasing. In this work, I store a user data in an ordinary database and use a landmark-based technique to query for a shortest path between call parties. For the future improvement, one may use the power of cloud computing technology to accelerate the query time. Moreover, many excellent query methods have been proposed recently, e.g., Apache Hive or Facebook's Presto. One can apply these techniques to enhance the performance of a detection system.

Collaborative Attacks

In the case of a sophisticated flooding attack, attackers may compromise both end systems, sender and receiver, to send flooding traffic. The proposed statistical algorithms fail to detect such anomalous pattern because this attack has complete SIP transaction. Even though this attack pattern require high computing resources, attackers can use a cloud service to conduct this attack. So, when compromised senders and receivers organize into pairs to flood a server's resources, we need a method to correctly classify legitimate traffic from attack traffic. For the future direction, an integration with policy-based or rate limitation-based maybe a solution to mitigate such attack.

Hybrid Attacks

Another challenge are dynamic and polymorphic DoS attacks. As described in Chapter 2, many protocols are used in an IMS system. Since packet floods can be generated from any combination of protocols, a defense mechanism is required to evaluate its performance under this situation.

VoIP Dataset

The researches in this area need to address the major weakness: the lack of standards of evaluation and the scarce information on modern types of attacks. Launching real attacks against real networks with real legitimate users is impractical. Moreover, due to user privacy agreements, VoIP/IMS providers are not able to contribute their data. Hence, the main difficulty remains to obtain real-world traces of attack incidents or even normal traffic. A pragmatic solution to these problems consists in organizing a close cooperation of the research community with operators. If it is difficult to cooperate with the organization, a publicly available labeled dataset is an alternative solution. One can use the simulation information in this work to create a dataset. However, this work focuses only on SPIT and SIP flooding attacks. There are several attack types, including the polymorphic attacks, that must be covered in the public dataset.

5.2.2 The Outlook of VoIP Security Threats

Cyber threats are developed as quickly as new technologies themselves. Modern malware is effective at attacking new platforms. At the time of writing this

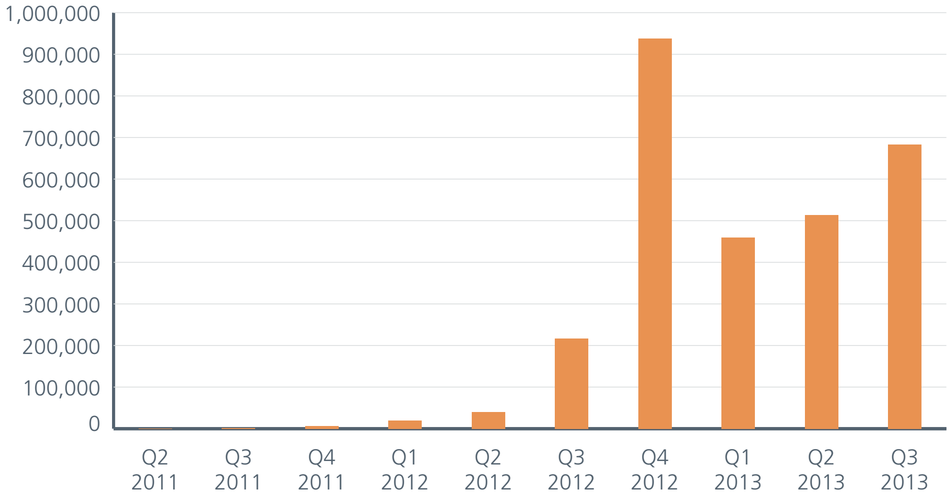

Figure 5.1. New Android malware samples.

dissertation, Figure 5.1 indicates the increasing number of new Android malware samples [1]. We can see that a rapid growth of malware targeting smartphone has become a serious threat. Once a victim's device has been infected, an attacker can steal any information in the device or recruit this device into a global botnet. Consequently, an attacker is possible to use an infected device to spread SPIT and flooding attacks.

After getting a contact list from a victim's device, an attacker delivers a spam call from the infected phone to his/her friends. The detection system classifies this call as a trusted call and then forwards this to the callee. It can bypass many SPIT prevention techniques including white/black lists because this call comes from friends. Moreover, if all calls are delivered only to the victim's friends, it may be difficult to notice any uncommon behavior in the billing statement.

Victim's devices may be used as botnets and compromised to perform DoS attacks. Now, we have not obviously seen that a DoS attack comes from compromised smartphones. However, due to the enhancement of smartphones' hardware and the increasing number of smartphone users, it has a potential to conduct the attack by using infected smartphones in the near future.

In this case, the trust filtering will not be effective because the traffic are

[1] McAfee labs threats report.

sent from legitimate users. Therefore, a suggestion for the further research is the collaborative threat measurement. It would be more advantages if the security framework can analyze the information from both end-users side and the operator side in order to create attack detection heuristics.

5.2.3 Open Discussion for the Deployment

This section discusses some aspects whether the SPIT detection should be distributed or centralized. In the following discussions, the trust score delivering equipment is regarded to be composed of two parts as follows:

- Monitoring: it is used to gather relevant information, such as call duration and call direction, that are necessary to estimate a score.
- Scoring: it processes the information, gathered by the monitoring part, and delivers a score to a callee.

Figure 5.2 illustrates the distributed approach where the SPIT functionality (monitoring and scoring) is distributed, but centralized per operator. For instance, the SPIT detection would be located in an application server communicating with all S-CSCFs. The SPIT detections in different networks would communicate their scores to SPIT detections in other networks.

The second variant of the distributed approach is that the SPIT functionality is still distributed, but they do not communicate their scores to others, i.e., each operator would operate their SPIT functions independently, and react to the locally determined score. The issues of this variant are as follows. Each operator is independent from other operators in deploying monitoring, marking and reacting functionality. This appears to be a practical approach. However, the effectiveness of trust scoring in the terminating IMS network still depends on measures of other operators' network. Therefore, if networks do not cooperate to share the trust scores, they may not exploit the full available information.

A possibility to overcome the disadvantages of a distributed approach is to centralize the scoring part, as shown in Figure 5.3. This centralization means a single trust scoring instance is located above the operator level and operated by a neutral organization. As the SPIT monitoring function has necessarily to be

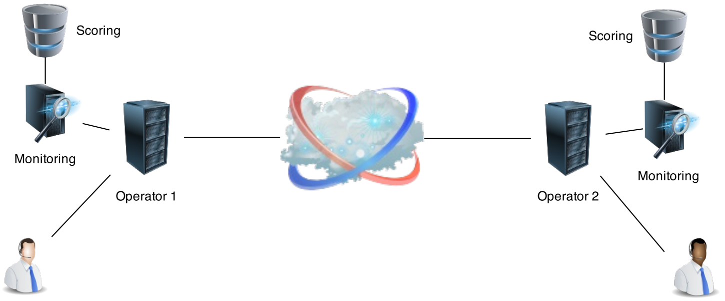

Figure 5.2. Distributed SPIT monitoring and scoring, centralized per operator.

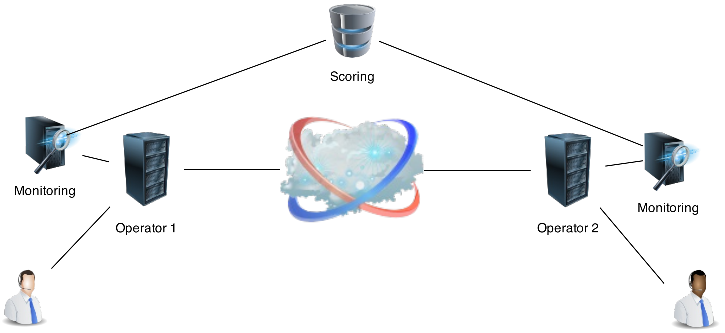

Figure 5.3. Distributed SPIT monitoring, centralized trust score.

located inside the networks to monitor the signaling traffic, this functionality is distributed across different networks, as before. A central trust scoring part guarantees always consistent scoring results, as only one score is delivered. However, additional traffic is generated to transfer the data to the central scoring instance. In addition, legal concerns may be related to a central scoring instance.

The author cannot conclude that which one is the best approach. This is an open discussion that the standard organization and operators need to find a solution together.

5.2.4 Legally-Compliant Guideline for the Deployment

As telecommunication is protected by many laws, the call filtering or blocking may face several legal consequences. Therefore, a researcher should not only concentrate on technical filtering mechanisms, but also consider implications from telecommunication laws and regulation. Below are legally-compliant guidelines for SPIT filtering deployment.

Transparency

A customer have to be explained the mechanism of identification, classification, and countermeasure of incoming calls.

Data Processing Control

For the trust inference process, a buddy list of each subscriber has to be shared to others. However, this sharing is processed within the operator side and no user can see the shared data. The proposed system also keeps track a calling behavior of each user. This feature might be an invasion of privacy. Therefore, the operator has to inform a customer about the purpose and the process of these mechanisms. They must be explicitly activated by a customer.

5.2.5 Legal Issues

SPIT legislation is a national and international issue and may differ per country. There is currently no uniform, worldwide-accepted definition of SPIT, neither in standardization nor in legislation. Laws even differ in the definition of electronic advertisement. In some countries, such as United States, electronic advertisement must additionally have a commercial background. Thus non-commercial advertisement, e.g., political, religious, or scientific advertisement, is allowed in the US while it is prohibited in the EU. Figure 5.4 highlights the problems that occur in an international religious advertisement campaign.

- Is it allowed to send religious bulk advertisement from the US (allowed) to recipients in the EU (prohibited)?

- Is it illegal to send religious bulk advertisement from the EU (prohibited) to recipients in the US (allowed)?

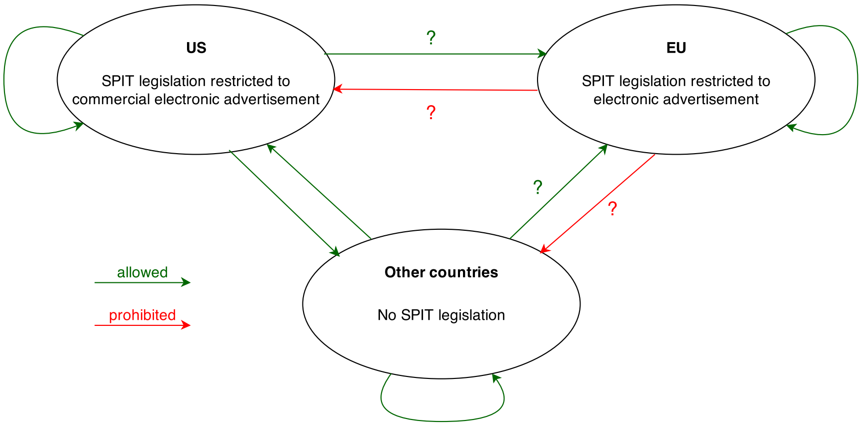
US
SPIT legislation restricted to commercial electronic advertisement
EU
SPIT legislation restricted to electronic advertisement
Other countries
No SPIT legislation
?
?
?
?
allowed
prohibited

Figure 5.4. International religious advertisement campaign.

Therefore, the difficulty for the SPIT prevention system is that it is subject to the corresponding national law. This means that besides monitoring and marking traffic technically as SPIT, the trust system has to consider the follows:

- whether the communication is international
- which the involved countries are
- which legislation is valid
- whether the technical SPIT classification corresponds to the legal situation.

The SPIT detection systems will be burdened to handle besides the technical part of SPIT monitoring and scoring and the evaluation of the legal situation of a suspicious call. This may involve the existence of a changing worldwide legislation; evaluated according to the source and the destination of a call and the location of the operator. It is quite uncomfortable for operators because it will not be easy for them to prove that their trust score complies with national or international laws. With this confusing situation, the operators are also exposed dangers like lawsuits or claims for damages.

Publications

Journal (Peer Review)

1. Noppawat Chaisamran, Takeshi Okuda, Youki Kadobayashi, and Suguru Yamaguchi, “SIP Flooding Attack Detection Using a Trust Model and Statistical Algorithms”, *Journal of Information Processing*, Vol. 22, No. 2, April 2014 (*related to Chapter 4, 5*).

2. Noppawat Chaisamran, Takeshi Okuda, and Suguru Yamaguchi, “Trust-based VoIP Spam Detection based on Calling Behaviors and Human Relationships”, *Journal of Information Processing*, Vol. 21, No. 2, pp. 188-197, April 2013 (*related to Chapter 3*).

International Conference (Peer Review)

1. Noppawat Chaisamran, Takeshi Okuda, Youki Kadobayashi, and Suguru Yamaguchi, “Trust-based SPIT Detection by Using Call Duration and Social Reliability”, in *Proceedings of the 19th Asia-Pacific Conference on Communications (APCC 2013)*, August 2013 (*related to Chapter 3, 5*).

2. Noppawat Chaisamran, Takeshi Okuda, and Suguru Yamaguchi, “Using a Trust Model to Reduce False Positives of SIP Flooding Attack Detection in IMS”, in *Proceedings of IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW 2013)*, pp. 254-259, July 2013 (*related to Chapter 4, 5*).

3. Noppawat Chaisamran, Takeshi Okuda, and Suguru Yamaguchi, “A Pro-

posal for Anomaly Traffic Detection in the IP Multimedia Subsystem using Tanimoto Distance and a Modified Moving Average", in *Proceedings of the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012)*, pp. 278-283, July 2012 (*related to Chapter 4*).

4. Noppawat Chaisamran, Takeshi Okuda, Gregory Blanc, and Suguru Yamaguchi, "Trust-based VoIP Spam Detection based on Call Duration and Human Relationships", in *Proceedings of the 11th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011)*, pp. 451-456, July 2011 (*related to Chapter 3*).

Technical Report

1. Noppawat Chaisamran, Gregory Blanc, Kazuya Okada, Takeshi Okuda, and Suguru Yamaguchi, "Basic Trust Calculation to Prevent Spam in VoIP Network based on Call Duration (Single Hop Consideration)", in *IEICE Technical Report, IA2010-51*, Vol. 110, pp. 1-6, November 2010 (*related to Chapter 3*).

Bibliography

[1] M. Poikselka and G. Mayer, *The IMS: IP Multimedia Concepts and Services.* United Kingdom: Wiley, third ed., 2009.

[2] J. Zar, "VoIP Security and Privacy Threat Taxonomy." VoIPSA, October 2005.

[3] R. Kuhn, T. Walsh, and S. Fries, "Security Considerations for Voice Over IP Systems." NIST Special Publication 800-58, January 2005.

[4] ETSI, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): NGN Functional Architecture." ETSI ES 282 001 V3.4.1, September 2009.

[5] 3GPP, "Security Architecture." TS 33.102 V10, May 2011.

[6] 3GPP, "Access Security for IP-based Services." TS 33.203 V10.2, May 2011.

[7] J. Golbeck and J. Hendler, "Inferring Binary Trust Relationships in Web-based Social Networks," *ACM Transactions on Internet Technology*, vol. 6, no. 4, pp. 497–529, 2006.

[8] J. Leskovec and E. Horvitz, "Planetary-Scale Views on a Large Instant-Messaging Network," in *Proc. of 17th international conference on World Wide Web*, pp. 915–924, 2008.

[9] "Skype." `http://www.skype.com`. Accessed: 2013-10-25.

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol." RFC3261, June 2002.

[11] 3GPP, "IP Multimedia Subsystem (IMS)." TS 23.228 V11.7, January 2013.

[12] J. Rosenberg and C. Jennings, "RFC5039: The Session Initiation Protocol (SIP) and Spam," 2008.

[13] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT Calls by Checking Human Communication Patterns," in *Proc. of IEEE International Conference on Communications*, pp. 1979–1984, 2007.

[14] D. Vinokurov and R. W. Maclntosh, "Detection and Mitigation of Unwanted Bulk Calls (Spam) in VoIP Networks." Patent WO2006000466A1, 2006.

[15] R. Dantu and P. Kolan, "Detecting Spam in VoIP Networks," in *Proc. of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, pp. 31–37, 2005.

[16] V. Balasubramaniyan, M. Ahamad, and H. Park, "CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation," in *Proc. of 4th Conference on Email and Anti-Spam*, 2007.

[17] R. Zhang and A. Gurtov, "Collaborative Reputation-based Voice Spam Filtering," in *Proc. of 20th International Workshop on Database and Expert Systems Application*, pp. 33–37, 2009.

[18] J. Peterson and C. Jennings, "RFC 4474: Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," 2006.

[19] K. Ono and H. Schulzrinne, "IETF Internet-Draft: Trust Path Discovery," 2006.

[20] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of Trust and Distrust," in *Proc. of 13th international conference on World Wide Web*, pp. 403–412, 2004.

[21] N. Chaisamran, T. Okuda, and S. Yamaguchi, "Trust-based VoIP Spam Detection based on Calling Behaviors and Human Relationships," *Journal of Information Processing*, vol. 21, no. 2, pp. 188–197, 2013.

[22] T. Chen and V. Venkataramanan, "Dempster-Shafer Theory for Intrusion Detection in Ad Hoc Networks," *IEEE Internet Computing*, vol. 9, no. 6, pp. 35–41, 2005.

[23] G. Shafer, *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[24] M. Richardson, R. Agrawal, and P. Domingos, "Trust Management for the Semantic Web," in *Proc. of 2nd International Semantic Web Conference*, pp. 351–368, 2003.

[25] "IP Phone Usage Status and Network Information 2011." `http://www.ntt-east.co.jp/info-st/network/traffic_h23/index.html`. Accessed: 2013-04-15.

[26] "NTT Telecommunication Services Report 2011." `http://www.ntt-east.co.jp/info-st/subs/ekimu/h23/index.html`. Accessed: 2013-04-15.

[27] T. Dang, B. Sonkoly, and S. Molnar, "Fractal Analysis and Modeling of VoIP Traffic," in *Proc. of 11th International Telecommunications Network Strategy and Planning Symposium*, pp. 123–130, June 2004.

[28] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in *Proc. of DARPA Information Survivability Conference and Exposition (DISCEX'00)*, 2000.

[29] P. Mell, "An Overview of Issues in Testing Intrusion Detection Systems," tech. rep., NIST, 2003.

[30] M. Farrell, "Cellphone networks overwhelmed after blasts in Boston." `http://b.globe.com/14uWliO`. Accessed: 2013-08-14.

[31] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms.* Laflin, PA: Prentice Hall, 2005.

[32] M. Garuba, C. Liu, and D. Fraites, "Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems," in *Proc. of 5th International Conference on Information Technology: New Generations (ITNG'08)*, 2008.

[33] G. Varghese, J. A. Fingerhut, and F. Bonomi, "Detecting Evasion Attacks at High Speeds Without Reassembly," *SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 327–338, 2006.

[34] V. Siris and F. Papagalou, "Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks," in *Proc. of Global Telecommunications Conference*, (Dallas, TX), pp. 2050–2054, 2004.

[35] Y. Rebahi, M. Sher, and T. Magedanz, "Detecting Flooding Attacks against IP Multimedia Subsystem (IMS) Networks," in *Proc. of IEEE/ACS International Conference on Computer Systems and Applications*, (Qatar), pp. 848–851, 2008.

[36] B. Reynolds and D. Ghosal, "Secure IP Telephony using Multi-layered Protection," in *Proc. of 10th Annual Network and Distributed System Security Symposium*, (San Diego, CA), 2003.

[37] H. Sengar, D. Wijesekera, and S. Jajodia, "Detecting VoIP Floods Using the Hellinger Distance," *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, pp. 794–805, June 2008.

[38] C. Hecht, P. Reichl, A. Berger, O. Jung, and I. Gojmerac, "Intrusion Detection in IMS: Experiences with a Hellinger Distance-Based Flooding Detector," in *Proc. of 1st International Conference on Evolving Internet*, (France), pp. 65–70, 2009.

[39] J. Tang, Y. Cheng, and C. Zhou, "Sketch-Based SIP Flooding Detection Using Hellinger Distance," in *Proc. of IEEE Global Telecommunications Conference*, pp. 1–6, 2009.

[40] M. Ali Akbar, Z. Tariq, and M. Farooq, "A Comparative Study of Anomaly Detection Algorithms for Detection of SIP Flooding in IMS," in *Proc. of 2nd International Conference on Internet Multimedia Services Architecture and Applications (IMSAA'08)*, 2008.

[41] A. Abimbola, J. Munoz, and W. Buchanan, "Investigating False Positive Reduction in HTTP via Procedure Analysis," in *Proc. of International conference on Networking and Services (ICNS'06)*, July 2006.

[42] T. Pietraszek and A. Tanner, "Data Mining and Machine learning-Towards Reducing False Positives in Intrusion Detection," *Information Security Technical Reporty*, vol. 10, no. 3, pp. 169–183, 2005.

[43] Z. Tian, W. Zhang, J. Ye, X. Yu, and H. Zhang, "Reduction of False Positives in Intrusion Detection via Adaptive Alert Classifier," in *Proc. of International conference on Information and Automation (ICIA'08)*, pp. 1599–1602, June 2008.

[44] R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan, "Using Neuro-Fuzzy Approach to Reduce False Positive Alerts," in *Proc. of 5th Annual Conference on Communication Networks and Services Research (CNSR'07)*, pp. 345–349, May 2007.

[45] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. United States: Wiley, second ed., 2001.

[46] NIST/SEMATECH, "e-Handbook of Statistical Methods." `http://www.itl.nist.gov/div898/handbook/`.

[47] "Seagull." `http://gull.sourceforge.net`. Accessed: 2013-04-25.

[48] "Open IMS Core." `http://www.openimscore.org`. Accessed: 2013-04-25.

[49] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks," in *Proc. of 19th International Conference on World Wide Web*, pp. 641–650, 2010.

[50] "IP Phone Usage Statistics and Network Information 2011." `http://www.ntt-east.co.jp/info-st/network/traffic_h23/`. Accessed: 2013-04-25.

[51] M. Abid, S. Song, H. Moustafa, and H. Afifi, "Efficient Identity-based Authentication for IMS Based Services Access," in *Proc. of 7th International Conference on Advances in Mobile Computing and Multimedia*, pp. 260–266, 2009.