Voice Over IP - Security and SPIT

Swiss Army, FU Br 41, KryptDet Report

Rainer Baumann* Stéphane Cavin† Stefan Schmid‡

baumann@hypert.net, stephane.cavin@gmail.com, schmiste@ethz.ch

University of Berne, August 24 - September 8, 2006

Abstract

Voice-over-IP (VoIP) has become an attractive alternative to the plain old telephone system, especially due to the much lower communication costs. However, there are several threats. In this document, several such threats are discussed. Moreover, we ask whether these systems could be used in environments with very high security demands as they for example exist in the Swiss army.

*Computer Systems Group, ETH Zurich
†Microsoft Switzerland
‡Distributed Computing Group, ETH Zurich

Contents

1 Introduction **4**

2 Security **5**
2.1 Confidentiality 5
2.2 Integrity 5
2.3 Availability 6
2.4 VoIP Security Threats 6

3 Spam over Internet Telephony (SPIT) **9**
3.1 Properties of VoIP Spam 9
3.2 Existing Solutions 10
3.2.1 Content Filtering 10
3.2.2 Turing Tests and Cryptographic Puzzles 10
3.2.3 Payments 11
3.2.4 White and Black Lists 11
3.2.5 Greylisting 12
3.2.6 Reputation Systems 12
3.2.7 Volume Based Models 12
3.2.8 Authentication 12
3.2.9 Statistical Analysis of VoIP Signaling 12
3.2.10 Aggressive Spam Prevention 13
3.3 Some Own Approaches for the SPIT Problem 13
3.3.1 Knowledge about Callee 13
3.3.2 Availability of Caller 13
3.3.3 Greylists with Tokens 13
3.3.4 Asking for Identity 14

4 A Biometric Framework for SPIT Prevention **15**
4.1 Concept and Architecture 15
4.2 Implementation 16
4.2.1 Using a PKI 16
4.2.2 Using Kerberos 16
4.2.3 PKI versus Kerberos 16
4.3 Conclusion 17

5 Protocols and Applications **19**
5.1 Session Initiation Protocol 19
5.1.1 Introduction 19
5.1.2 SIP Security 20
5.1.3 Threats and Mitigation 21
5.1.4 Discussion 23
5.2 Skype 23
5.3 H.323 24
5.3.1 H.323 Security 25
5.3.2 Attack examples on H.323 25

6 A Sample Attack on SIP: Man in the Middle **26**

7 Conclusion **30**
7.1 Recommendations 30
7.2 Challenges for the Future 31

1 Introduction

Internet telephony is becoming more and more important. No matter in which country you are, a look through the windows of an internet cafe reveals numerous users of Skype—a software which was only released in 2004 and has now up to 6 million users being online at any time. The so-called Voice over IP (short: VoIP) technology offers cheap calls all over the world. Besides the popular Skype solution, there exist various other open protocols such as SIP or H.323. Moreover, VoIP functionality has been integrated into many instant messaging tools such as *ICQ* or *Google Talk*.

VoIP systems are an attractive alternative compared to traditional telephony for various reasons: use of existing internet infrastructure, cheap connections, no need for expensive hardware, and so on. However, so far, it is not clear whether these solutions can be used in security-critical environments.

This document studies VoIP from a security perspective. We are interested in questions such as: Can Alice communicate securely with Bob over today's VoIP systems, that is, such that an attacker cannot follow their conversation (e.g., by decrypting the traffic)? Is it possible for the attacker to pretend being Alice such that Bob provides her with confidential information? Are man-in-the-middle attacks possible? And so on. We will also look at threats which may reduce the availability of a service, for instance denial-of-service (DoS) attacks. We study the security of state-of-the-art systems such as Skype, SIP, and H.323. We show that attacks are sometimes very easy, and give an example of a man-in-the-middle attack for SIP.

VoIP also introduces threats which have not existed in traditional telephony. One such example is *spam*: Since making a call is almost free over VoIP, the distribution of unsolicited mail is attractive. We will show that VoIP spam (a.k.a. SPIT) is quite different from email spam, and it is harder to establish countermeasures, i.e., approaches such as Bayesian filters are useless. We will review and evaluate several existing solutions for the SPIT problem in detail, and then present our own approaches, for example a biometric framework which keeps away spammers by requiring them to contribute personal information.

The rest of this report is organized as follows. In the next section we give an overview of VoIP security in general. Section 3 focuses on a sample security problem in more detail, namely *spam*. We then look at the various VoIP implementations in use today and analyze their security (Section 5). Section 6 presents our SIP man-in-the-middle attack. Finally we conclude by giving recommendations on using VoIP today in security-critical environments and by stating some key challenges for VoIP security in the future.

2 Security

When dealing with modern information technology systems such as VoIP, security is omnipresent. There are mainly three key aspects of information security often referred to as the *CIA triad*: *confidentiality*, *integrity* and *availability*. There are additional aspects to security which are not included in the *CIA triad*, e.g., non-repudiation or accounting. However, they are of minor interest to the users of VoIP systems, and hence we will not discuss them here.

In this section, we first give a general definition of each of the three aspects of the *CIA triad* and discuss them briefly with respect to VoIP. A more detailed list, organized with respect to the protocol layers is depicted in Table 1. The interested reader looking for more information is referred to [32, 33, 34, 48, 49]. Second, we give an overview on the most important security threats for VoIP in gerneral. Section 5 will then look at more specific protocol and application threats.

2.1 Confidentiality

Definition: Confidentiality means that no information will be disclosed to unauthorized subjects. Information meets the confidentiality criterion when disclosure or exposure to unauthorized individuals or systems is prevented; it ensures that only those with the rights and privileges to access information are able to do so.

We have to distinguish between two information sources: (i) the audio signal and (ii) the call control. (i) Threats regarding the audio signal are eavesdropping and man-in-the-middle attacks. Thus, the confidentiality between the called and the calling party can be broken. (ii) The threats regarding call control or signaling are the exposure of information about users (also names, passwords, etc.), systems (e.g., system version) and patterns. This information can be used for attacking a system or the privacy.

Defense Strategies:

- physical protection (e.g., equipment rooms)
- use of Ethernet switching instead of shared media
- use of VLANs, VPNs where applicable (just like your data network!)
- encrypting conversations and call control, secure the media stream SRTP
- ensuring that routing tables, instructions, account codes are well maintained and password protected

2.2 Integrity

Definition: Integrity captures the trust that can be placed in the information. Data integrity assures that the information has not been altered between its transmission and its reception. There are two categories of integrity (i) source integrity and (ii) data integrity. (i) Source integrity guarantees that the data comes indeed from the correct sender. (ii) Data integrity is compromised when information has been corrupted, willfully or accidentally, before it is read by its intended recipient.

Integrity in VoIP should ensure that packets get from one point to another without modification. Regarding the audio signal, the main threats are impersonation of user or injection of other audio. The consequences are hard to be estimated and can go from annoyance to severe incidents. With respect to call control or signaling the major threat is fraudulent use of telephony resources as toll fraud or impersonation.

Defense Strategies:

- use of encryption for secure communications
- changing default password, minimum length, enforce periodic change
- never exchanging passwords in clear text
- password maintenance, delete ex-employees, security codes

2.3 Availability

Definition: Availability means that information or resources are accessible when required. Most often this means that the resources are available at a rate which is fast enough for the wider system to perform its task as intended. It is certainly possible to protect confidentiality and integrity, but an attacker can for example run a Denial of Service attack (DoS) to reduce the availability of resources.

For VoIP, availability means ensuring that communication services are accessible to the users, especially avoiding any adverse effects resulting from a DoS attack or computer worm. Typical DoS attacks seek to (i) crash or (ii) overload a system. The consequences are partial or total loss of telephony or related services. (i) The teardrop attack involved sending IP fragments with overlapping oversized payloads to the target machine. A bug in the TCP/IP fragmentation re-assembly code caused the fragments to be improperly handled, crashing the operating system as a result. Similarly, VoIP stacks can also suffer from malformed packets. A ping of death involves sending a malformed or otherwise malicious ping to a computer. Sending an oversized ping often crashes the target computer. (ii) The smurf attack, named after its exploit program, is a denial-of-service attack which uses spoofed broadcast ping messages to flood a target system.

Defense Strategies:

- rigorous virus updates, OS and software patches
- intrusion detection systems
- protect access from external sources (firewall)
- limit access from internal sources (firewall)
- use of 802.1 p/q (VLAN) to isolate and protect voice domain bandwidth from data domain DoS floods

2.4 VoIP Security Threats

There are several security threats related to VoIP. Following we briefly list the most important ones.

Layer	Attack Vector	Confidentiality	Integrity	Availability
Network Interface	Physical Attacks	x		x
	ARP cache	x	x	x
	ARP flood			x
	MAC spoofing	x	x	x
Internet	IP spoofing			
	Device	x	x	x
	Redirect via IP spoof	x	x	x
	Malformed packets	x	x	x
	IP frag	x	x	x
	Jolt			x
Transport	TCP / UDP flood			x
	TCP / UDP replay	x	x	
Application	TFTP server insertion		x	
	DHCP server insertion		x	
	DHCP starvation			x
	ICMP flood			x
	SIP			
	Registration Hijacking	x	x	x
	MGCP Hijack	x	x	x
	Message modification	x	x	
	RTP insertion			
	Spoof via header	x	x	x
	Cancel / bye attack			x
	Malformed method			x
	Redirect method	x		x
	RTP			
	SDP redirect			x
	RTP payload			x
	RTP tampering	x	x	x
	Encryption	x	x	x
	Default configuration	x	x	x
	Unnecessary services	x	x	x
	Buffer overflow	x	x	x
	Legacy Network	x	x	x
	DNS Availability			x

Table 1: VoIP vulnerabilities based on protocol layers and CIA triad [34].

- `Reconnaissance attacks:` intelligent gathering or probing for assessing the vulnerabilities of a VoIP system
- `Floods and Distributed Floods:` overloading a system resulting in a denial of service attacks
- `Protocol Fuzzing:` using semi-valid input to crash or confuse a system
- `Spoofing:` misuse of someone other's address or identity
- `Session Anomalies:` confusing signaling and call control for session hijacking or denial of service
- `Stealth Attacks:` frequent requests (calls) for annoying users
- `VoIP Spam:` transmitting unsolicited and unwanted bulk messages (see Section 3)

3 Spam over Internet Telephony (SPIT)

Although there exists only little VoIP spam today, it may become a big threat in the near future. As has been demonstrated by email spam (e.g., Nigeria scam industry [38]), people are often taken in by these kind of advertisements. In this section, we first look at the characteristics of SPIT and argue why traditional solutions for email spam filtering fail. Afterwards, several possible solutions are discussed and compared. For a good overview on the topic, note the NEC documents [36], the IETF draft on SIP [43], and the thesis by Radermacher [38].

3.1 Properties of VoIP Spam

At the heart of the SPIT problem lies the fact that sending advertisements comes (almost) for free, is often anonymous and not illegal, making VoIP an attractive medium for spammers [38]. Unlike traditional telephone systems where the telemarketer had to pay for each call, advertisements can be sent in parallel to thousands of potential customers at no transmission cost.

In this respect, SPIT is similar to the email spam problem which many Internet users face today: As companies did no longer have to pay postmen to carry their advertisements to the people's mailboxes, but could send unsolicited email to virtually all inboxes for free, the amount of advertisement mail exploded. However, although email spam will still be a big challenge in the future, the numerous solutions proposed over the last years have helped to mitigate the problem significantly. For example in *Spamato* [4], users collaboratively filter spam with respect to suspicious text contents, suspicious sender domains, etc.

Unfortunately, many great mechanism which work for email spam fail completely in the context of VoIP. There are many reasons. First, an email usually arrives at a server before it is finally downloaded by the user. Such a mail server can therefore apply many filtering strategies, for instance, it can check whether the text body of the email mentioned pharmaceutic products. In contrast, in VoIP, human voices are transmitted rather than text. To recognize voices and to determine whether the message is spam or not is still a very difficult task for a computer. What is more, a recipient of a call only learns about the subject of the message when she or he is actually listening to it.

Also from a user's perspective, SPIT is quite different from spam. Although a spam email is a nuisance, it is typically easy to delete such an email. But it can be really bad if a regular email from a friend is considered spam and not delivered to a user's inbox. That is, it may be tolerable if an email spam filter yields a large ratio of false negatives, but the filter should avoid false positives completely.

The situation looks different for SPIT. Receiving a spam email often means that the telephone rings, possibly waking users up in the middle of the night. On the other hand, if a call by a friend does not get through, the friend immediately recognizes that she or he has been filtered, and can try again—possibly using a different communication channel.

3.2 Existing Solutions

Having motivated the SPIT problem, in this section, we look at some potential solutions. We will see that there is no panacea for the spam problem, as all approaches come with some drawbacks. However, there are certain design mistakes that can easily be avoided. For example, the VoIP phone numbers should not be as densely populated as regular phone numbers in order to avoid phone number guessing. Generally, we believe that SPIT will continue being a threat in future.

An overview and classification of SPIT prevention methods is depicted in Figure 1.

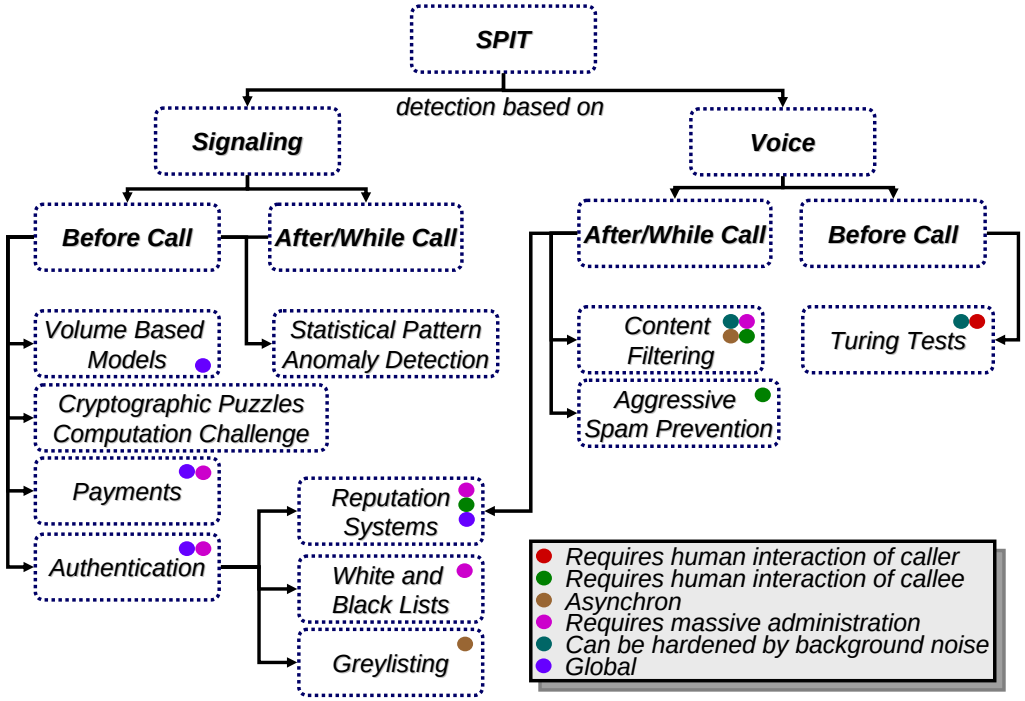


Figure 1: Overview and Classification of SPIT Prevention Methods

3.2.1 Content Filtering

As already mentioned, at the time a user learns about the contents of a call, the connection has already been established—the spam cannot be analyzed before it is actually delivered. Therefore, classic spam filtering techniques such as Bayesian spam filters or URL spam filters are useless. Moreover, even if the content is stored on a voice mail box, it is still difficult today for speech recognition technologies to decide whether it is spam or not.

3.2.2 Turing Tests and Cryptographic Puzzles

Fully automated SPIT of so called *bots* is one of the cheapest and most annoying thing. To fight them, they can be challenged in several ways (so-called *Turing tests*).

1. Voice Menu: Before a call is put through, a computer asks the caller to press a certain key combination, for example “press #54”.

2. Challenge Models: Before a call is put through, a computer asks the caller to solve a simple equation and to type in the answer, for example "divide 10 by 2".
3. Alternative Number: Under the main number a computer announces an alternative number. This number may even be changed permanently by a call management server. All of theses methods can even be enforced by enriching the audio signal with noise or music. This prevents SPIT bots from using speech recognition.

Such Turing tests are attractive, as it is often hard for computers to decode audio questions. However, these puzzles can not be made too difficult as human beings must always be able to solve them. Therefore, there are concerns about this approach in the long run.

Cryptographic puzzles [29] may also help to detain spammers. Whenever a caller tries to establish a connection, he has to solve a small puzzle consuming computational resources (CPU and bandwidth [2]). Clearly, as the computational power is limited, the number of parallel connection requests remains small. The drawback of this solution is that a regular caller with a slow machine may also experience unacceptable delays due to the puzzle challenges. Finally, as spammers sometimes use virus-infected machines (so called zombies), their computational power can be large.

3.2.3 Payments

The main reason for the spam problem is the fact that the cost of sending spam is almost zero. A straight-forward solution would therefore be to charge the caller a small amount of money for each connection attempt. This amount should be so small that VoIP calls remain virtually free for regular users, but prohibitively high for spammers. This is of course a difficult trade off. What is more, the implementation of such a payment infrastructure may be an ambitious endeavor.

Another idea is to charge back the cost (*payments at risk* [1]) if the receiver decides that the call is not spam. Unfortunately, today, Internet transactions always cost a minimum amount of money, e.g., 25 cents [43].

3.2.4 White and Black Lists

A very effective solution to the SPIT problem are white lists. Thereby, a user explicitly states which persons are allowed to contact her. A similar technique is also used in Skype: If Alice wants to call Bob, she first has to add Bob to her contact list and send a contact request to Bob. Only when Bob has accepted this request, Alice can make calls to Bob. Given that there are authentication mechanisms which prevent some attacker from pretending being Bob's friend Alice (address spoofing), unsolicited calls can be prevented. In general, white lists have an introduction problem, as it is not possible to receive calls by someone who has not yet been put on the white list explicitly.

Black lists maintain addresses that identify spammers and can be used in addition to white lists. The drawback of black lists however is that addresses can often be spoofed or changed easily by spammers unless there are inter-domain authentication mechanisms.

White and black lists have been studied intensively also in the context of VoIP, and the interested reader may refer to [10, 42, 46, 51]. Also note that there are always two approaches to create white and black lists: these lists can either be generated manually, or

they can be generated automatically using some statistical analysis of traffic or volume patterns.

3.2.5 Greylisting

Greylisting is a useful technique to filter spam emails, and it can also be applied to VoIP [38, 47]. Thereby, each call is blocked unless the same sender (w.r.t. IP address) tries to establish the call again within a certain time period. However, there are many concerns about this approach: First, it seems easy to circumvent the filter by just making second attempts. In addition, greylisting may block emergency calls from friends.

3.2.6 Reputation Systems

The idea of a reputation system [45, 27] is to give Alice a hint about the reputation of a caller before she answers the call. If the reputation is poor, she can decide not to accept the call. Unfortunately, reputation systems are often complex in distributed environments and susceptible to false praise. Moreover, if new identities are easy to acquire, a user with a negative reputation can just open a new account.

3.2.7 Volume Based Models

The idea here is that ISPs should restrict the number of VoIP connection requests their customers can execute over time. Of course, it is unlikely that ISPs will really collaborate in this respect, as they have incentives to be slightly less restrictive than their concurrence.

3.2.8 Authentication

Both SIP and H.323 support a vast variety of models for user authentication. Such authentication methods can be hardened for preventing anonymous VoIP traffic [28, 9, 15, 8] However, a global authorization model is not realistic today.

3.2.9 Statistical Analysis of VoIP Signaling

The idea of static analysis of VoIP signaling is to monitor the signaling traffic on the recipients' access domain gateway [31]. For each *external* identity observed in the signaling routing data, counters may be maintained for the number of times call setup and call termination requests went in or out of the access domain. These counters can then be statistically evaluated, for example by assuming that they have characteristic distributions. If this assumption is violated, various actions can be taken such as:

- Warning: Display the text warning on the phone, use special ringing tone.
- Call delay: Switch the caller to the recipients voice mail, reject the request and report the callerID and the missed call at a later time.
- Call cancellation: Drop the call setup on behalf of the recipient.

3.2.10 Aggressive Spam Prevention

Aggressive spam prevention mechanisms fall into two categories: (i) active publishing of incorrect information and (ii) counter attack on spammers [39]. Proactive publishing of incorrect information, namely SIP addresses, is a possible way to fill up the databases of the spammers with existing contacts. This increases the cost for a successful delivery of spam. Counter attack on the infrastructure of spammers is a way to bring them out of business. This is most effective if many victims use this technique. But this method is quiet expensive and dangerous since it could be misused for distributed denial of service attacks.

3.3 Some Own Approaches for the SPIT Problem

VoIP spam has several properties which can be used in a SPIT filtering system. In this section, we seek to exploit some of these properties, and we present novel mechanisms and extensions to the filters presented in the previous section. We believe that some of our suggestions are worth being studied in more detail in future. In the subsequent section, we will then focus on an idea for a biometric framework for the SPIT problem.

3.3.1 Knowledge about Callee

A crucial difference between a "regular" caller and a spammer is that the former typically has a certain knowledge about the person he is calling. Therefore, it seems to be natural to search for SPIT filters which ask the caller to provide specific information about the callee, for example his name, his address, his favorite food, etc. Of course, this information should be easy to obtain for regular callers, but not for spammers. Whether this is a feasible approach, and how it can efficiently implemented, is subject of future work.

3.3.2 Availability of Caller

Another property which distinguishes regular callers from spammers is the fact that it is typically impossible to call spammers back—spammers are not available. A solution could therefore be to perform some handshake protocol in the beginning, by which a caller is always called back. Note that even if the spammer tries to be available, the huge number of call backs during a massive spam attack will work similar to a DoS attack.

3.3.3 Greylists with Tokens

As we have mentioned earlier, greylists inherently engender a delay, as the caller is required to call again after a certain time period. However, we can imagine this being the case only for the first contact between caller and callee. Concretely, one solution would be to agree after each successful call on a certain shared secret. The next time Alice calls Bob, she can just present this secret which enables her to bypass the greylisting system and contact Bob instantaneously.

3.3.4 Asking for Identity

A general theme in the quest for spam filters is to require contributions from the spammer, for instance in terms of computational power, network resources, etc.: While such contributions are cheap for sporadic callers, they are prohibitively expensive for spammers. Also interesting would be to ask the caller to provide some sort of identity. In an extreme case, just for illustration, a caller can be required to present a valid credit card number whenever he tries to establish a connection.

4 A Biometric Framework for SPIT Prevention

A major difficulty in coping with the SPIT problem is the fact that spammers can change their identity frequently. Methods such as blacklists fight an uphill battle in the presence of continuously altering identities. Therefore, it is vital in any spam filtering system to inhibit these so-called *Sybil attacks* [12].

Binding identities to persons can help to prevent Sybil attacks. In this section, we present a generic framework which tackles the SPIT problem. In this solution, global servers bind the users' identities to personal data; in our case, to biometric such as a voice. Consequently, unlike in other solutions, spammers cannot obtain new identities even if they change the ISP.

4.1 Concept and Architecture

The general architecture of our system is shown in Figure 2. We use a set of trusted authentication servers (A). Before a client (C) uses VoIP for the first time, he has to register with an authentication server. The goal of this procedure is to record the user's voice and bind it to his VoIP ID. This is done as follows. First, the client calls the server. The server then asks the client to repeat a sentence, for example, a phrase from Goethe's Faust. In order to enhance the security of this procedure, the phrases should be different for each registration request. Moreover, several different languages should be offered such that each client can use his mother tongue. After completing Step 1 (cf Figure 2), the server stores the client's voice file and sends back credentials; in case of a PKI infrastructure, this is a server signed public key (Step 2). The client can now make arbitrary calls to other clients, authenticating himself using his credentials (Step 3). Anyone receiving such a call verifies the identity by checking the credentials; this may involve contacting the authentication servers (Step 4).

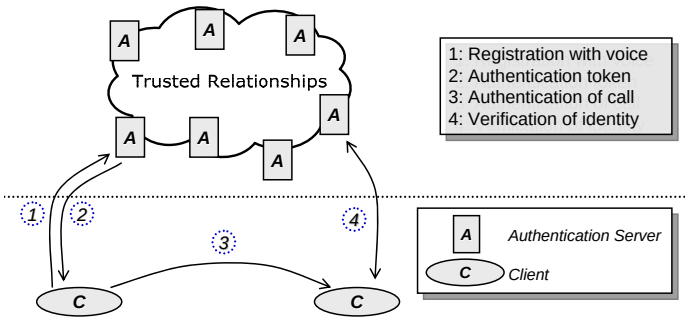


Figure 2: System architecture.

The key idea here is that it is impossible for a client to run a Sybil attack: A client who wishes to obtain additional identities is unmasked by the authentication server: The servers run a voice recognition software to reject duplicated registrations.

Observe that this approach has desirable properties. First, the solution is independent of a specific VoIP protocol and inter-operable: Authentication can be done centrally for all sorts of clients, e.g., SIP, Skype, etc. Moreover, an attacker cannot obtain new identities by switching to another provider either, as our approach is also ISP-independent. Also note that this solution is slightly different from many biometry-based systems in the sense that

we do not use biometric data for the authentication, but only as a reference data to which we can compare future registration requests.

Step 1 is time consuming, but registration is executed very seldom (e.g., once a year). Step 3 on the other hand is performed before each call, and is a quick operation: It does not involve any sentence repeating or so, but only the credential verification (e.g., checking an RSA signature).

Having described the general ideas on a high level, in the following, we will describe two sample implementations. It turns out that several options are possible, for example an implementation using a *public key infrastructure* (PKI), or an implementation using *Kerberos*.

4.2 Implementation

We use SIP as the VoIP protocol and the cryptographic authentication protocol is either a PKI system [22] or a Kerberos system [50].

4.2.1 Using a PKI

In case of a PKI authentication infrastructure, a client authenticates its calls based on an asymmetric certificate which proves his identity. The steps of authentication are the following (see Figure 3): The caller first checks if he has a valid certificate. If this is not the case it (re-)registers itself with its voice at a Certification Authority (CA). The CA verifies the callers identity based on its voice and issues a certificate for the caller. This certificate contains the caller's VoIP ID and is signed with the private key of the CA; everyone who knows the CA's public key can verify this signature and therefore the caller's ID.[1] The caller sends this certificate to the callee, who can then check for revocation and decide to accept or deny the call.

4.2.2 Using Kerberos

The authentication infrastructure can also be realized with Kerberos. Thereby, a client is bound to authenticate its calls based on an *once-ticket*.

In more detail, the steps of authentication are the following (see Figure 4). The caller first checks if he has a valid ticket-granting ticket. If this is not the case it (re-)registers itself with its voice at an Authentication-Server (AS). The AS verifies the callers identity based on its voice and issues a *ticket-granting ticket* to the caller. This ticket-granting ticket enables the caller to get a ticket for a call to a certain callee. This ticket is then used for authenticating the caller to the callee. In this case, the verification procedure is symmetric.

4.2.3 PKI versus Kerberos

Both the Kerberos and the PKI based system fulfill the requirements of our authentication protocol. But there is a crucial difference between these two realizations. The Kerberos

[1] The public key of the CA may be stored in any client from the beginning.

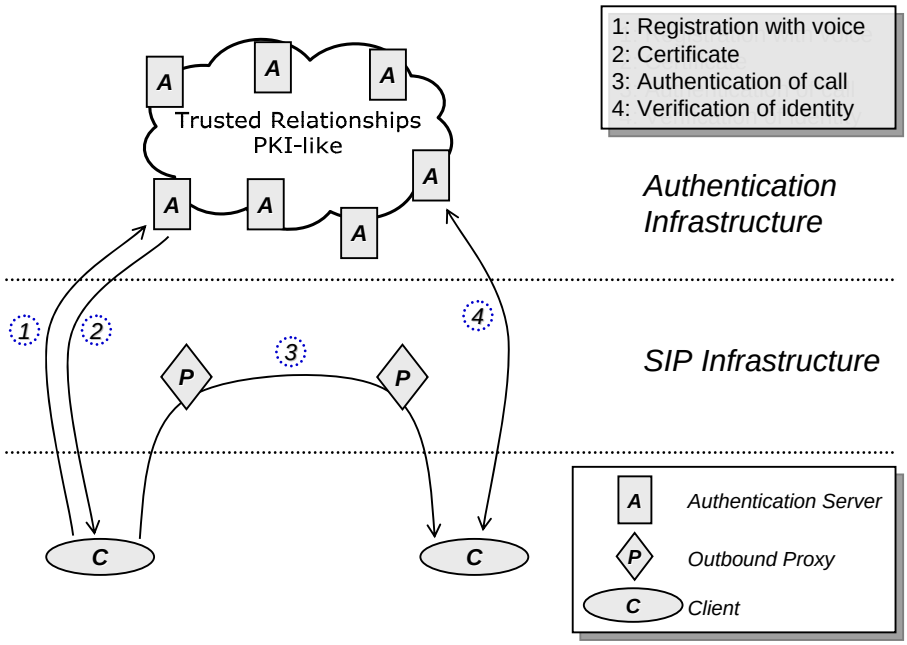


Figure 3: Biometric authentication for SIP based on PKI.

system allows for simple tracking of calls by the ticket-granting ticket server while the CA of a PKI system is unable to do so. In addition, there are also several differences between a Kerberos system and a PKI which are related to administration, communication and processing overhead. For example, while the registration step (Step 1) uses asymmetric cryptography in both cases, the Kerberos solution is typically faster in Step 3, as verification is based on classic cryptography only. For more details refer to [22, 50].

4.3 Conclusion

In this section we have proposed our own SPIT mechanism which binds clients to their identity by requiring clients to register their voice on central authentication servers. These servers ensure that the biometric data of each client is unique, and hence prevents clients from obtaining several accounts to white-wash their spamming activities.

We think that this approach may be interesting in other domains as well. However, there are several challenges in practice. For example, in truly global and inter-operable environments, the certification servers must be powerful in order to avoid bottle-necks. Moreover, we have been told that today it is still hard to distinguish voices of thousands of users, as voice patterns are sometimes close to each other, and patterns can also be changed, for instance by using some background noise. Although we believe that technological progress will mitigate these problems, and that voice will also be transmitted in higher quality in future, this solution may currently problematic for large-scale usage.

However, the general concept may be applied for different data in the meanwhile. For instance, one idea would be to ask all clients to register a unique and valid mobile phone number for each VoIP ID.

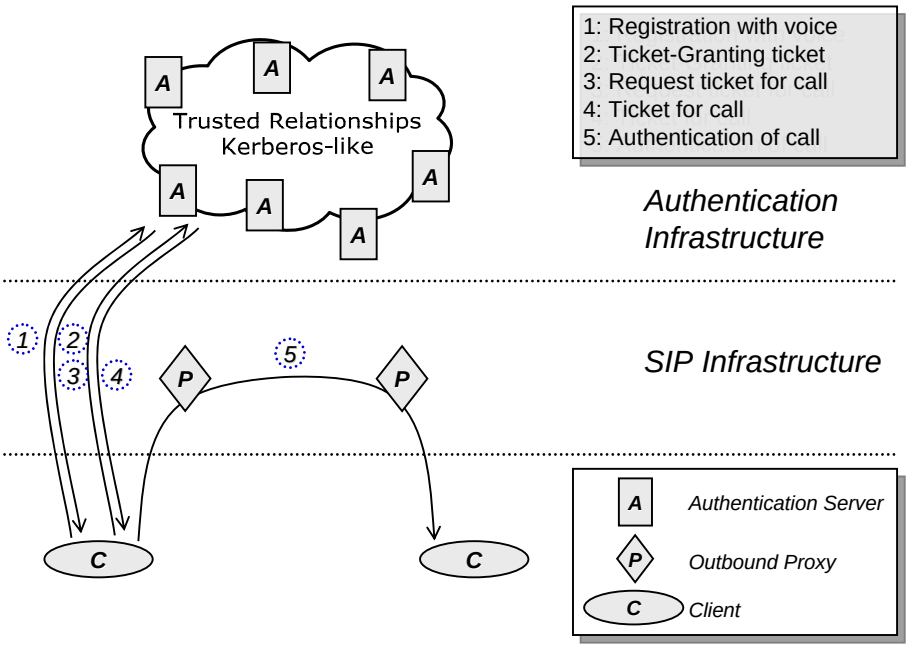


Figure 4: Biometric authentication based on Kerberos.

5 Protocols and Applications

5.1 Session Initiation Protocol

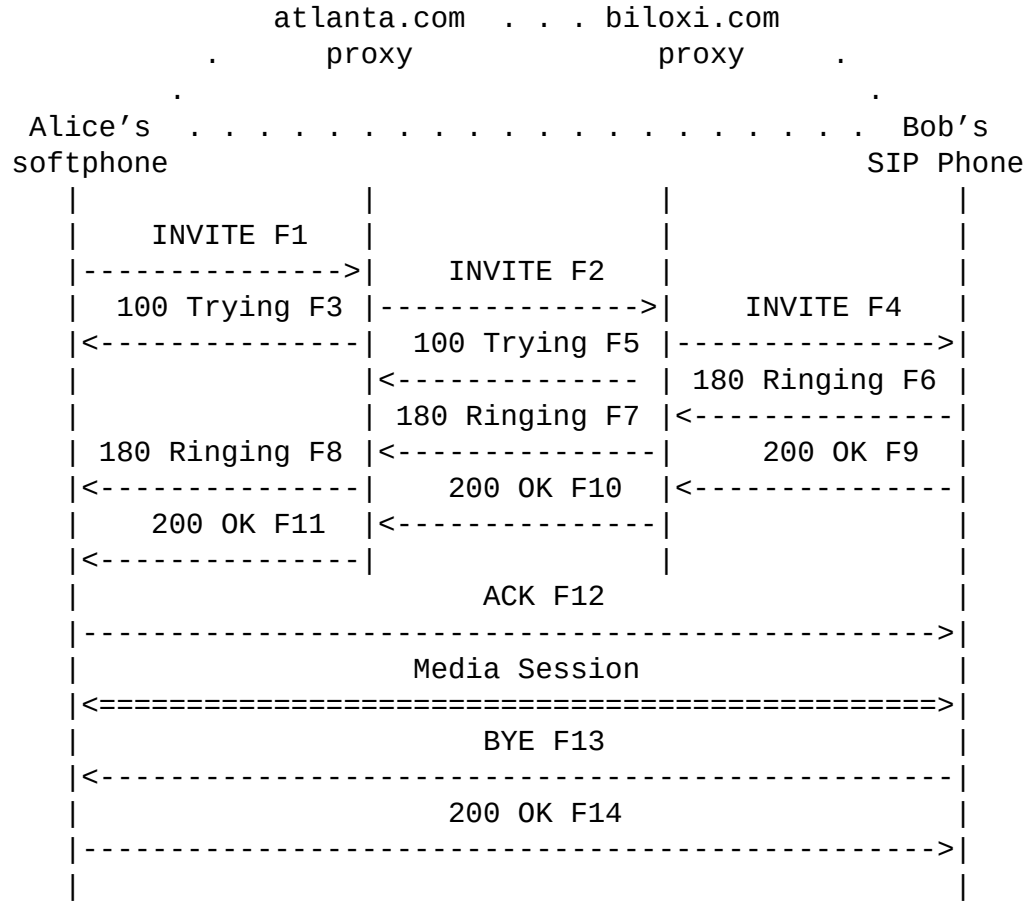
5.1.1 Introduction

The Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF) standard for IP telephony (RFC 3261 [44]). It is a text-based application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. SIP uses either UDP, TCP or TLS [11] as a transport protocol. The nature of the session established is defined by the content of the body of the request initiating the session. Most of the time, the Session Description Protocol (SDP, RFC 2327 [20]) is used to describe which kind of session is required (voice, video, instant messaging).

SIP is a request-response protocol. Therefore, every SIP entity is composed of two parts: the user agent client (UAC) which sends requests, and the user agent server (UAS) which responds to requests. In the SIP world, a user is represented by a type of Uniform Resource Identifier (URI) called SIP URI. The exact BNF can be found in [44]. A typical SIP URI is in the form *username@host* (e.g., bob@biloxi.com).

For locating purposes, SIP enables the creation of a server infrastructure (network of proxy servers) to which user agents can send registrations, invitations to sessions, and other requests. A proxy server is usually responsible for a particular domain and can be found by running a DNS query. To resolve a SIP URI to an actual endpoint IP address, another SIP server role is defined: the *registrar*. Every SIP client has to register (using a SIP REGISTER request) with the server responsible for his domain if he wants to be reachable. Basically, the registrar keeps a link between user SIP URI (called AOR, address-of-record) and user contacts (i.e., location). Note that a client can register several contacts for the same AOR. The links are stored in a database called the *location service*. It is important to note that the concept of proxy server and registrar are logical and not physical. A SIP server can play both roles.

To illustrate how SIP works, let us assume that Alice (sip:alice@atlanta.com) wants to start a voice session with Bob (sip:bob@biloxi.com). This example is taken from [44], a schema is shown below. Alice's SIP client has been configured to use the atlanta.com proxy for all her outgoing requests. Therefore, she sends a SIP INVITE (F1) to her proxy, which will forward it to Bob's proxy (F2). As Bob has registered, the biloxi proxy is able to forward the INVITE to Bob (F4). Bob will then accept the call and send a 200 OK. It will be acknowledged by Alice. Consequently, the media session can take place. The call can be terminated by Alice or Bob by sending a BYE.



5.1.2 SIP Security

Client-Side Authentication SIP provides a stateless, challenge-based mechanism for authentication based on authentication in HTTP ([14]). For any incoming request, the user agent server may challenge the expeditor of the request to provide assurance of its identity.

In the first version of SIP (RFC 2543 [21]), "Basic Authentication" was allowed. In this authentication mode, the UAC sends the user credentials in clear text. Due to its poor security, this mode was deprecated in the new standard version ([44]). The recommended authentication mechanism is the "Digest Authentication".

In digest authentication mode, for every incoming request, the UAS will send a challenge:

```
WWW-Authenticate: Digest
         realm="biloxi.com",
         qop="auth,auth-int",
         nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
         opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

The UAC of the client should then resend its invite with a special header stating its answer:

```
Authorization: Digest username="bob",
         realm="biloxi.com",
```

```
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="sip:bob@biloxi.com",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

The response is a hash of the concatenation of the user password and different info present in some headers.

This "Digest Authentication" mechanism provides message authentication and replay protection only. Nothing prevents an attacker from reading or modifying the message: there is no message integrity and confidentiality.

Server-Side Authentication SIP has no built-in mechanism to authenticate a server (proxy server, registrar, redirect server). However, it is recommended to use TLS (and check certificate validity). Servers should authenticate themselves using mutual TLS (MTLS). TLS provides a good hop-by-hop authentication.

Integrity and Confidentiality To create a session, most of the time, a SIP INVITE contains a body with session information. This body may be protected by S/MIME (RFC 1847 [16], RFC 2633 [40]). This provides integrity and confidentiality, but may pose some inter-operability problems as some firewalls might want to look at the body. Indeed, S/MIME provides end-to-end security. Moreover, to provide end-to-end integrity, a UAC may provide a copy of the whole message in the body part. This provides partial confidentiality, as the UAC may omit some headers in the "visible" version of the INVITE. Some other headers are mandatory and are visible.

Integrity may also be guaranteed by using transport or network layer security (TLS [11] and IPSec [30] respectively). Both methods encrypt the signaling traffic. In general, this is achieved by using certificates.

5.1.3 Threats and Mitigation

SIP does not provide real protection (no confidentiality, integrity or availability). Nonetheless, by using another standard, a lot of threats can be mitigated. The following threats have been described in [44].

Registration Hijacking

- The SIP registration mechanism is based on the *From* and *To* headers of the REGISTER requests. When receiving a REGISTER from a UAC, the registrar has to verify that the identity in the *From* has the permission to change the contacts of the address-of-record specified in the *To*. Without authentication, a malicious UAC could send a modified REGISTER, which de-registers valid contacts and registers new contacts. All subsequent requests will then be forwarded to the attacker endpoints.

- This threat can be mitigated by using "Digest Authentication" to authenticate the client.

Tampering with Message Bodies

- SIP message bodies are not encrypted, which means that a malicious proxy server may be able to modify the content of the body without notice. In case of a voice call, a malicious proxy may change the IP addresses in the INVITE request and OK response to make all RTP ([18]) packets transit by a malicious endpoints that can eavesdrop the conversation.
- To prevent such threats, the SIP client must use an end-to-end mechanism. A valid solution presented in Section 5.1.2 is S/MIME.

Tearing Down Sessions

- Once a dialog is established, subsequent requests can be sent to modify or terminate the dialog. It is then critical that the attacker cannot "sniff" the traffic and store important dialog info. If he is able to see the INVITE and the corresponding OK, he may be able to send a BYE to terminate the dialog, or send a re-INVITE with a different SDP to redirect all the media flow to a controlled endpoint.
- The most effective countermeasure to this threat is the authentication of the sender of the INVITE/BYE.

Denial of Service and Amplification

- Deployed SIP proxy servers often face the public internet. Hence, DoS is a probable threat, as it is easy to implement. An attacker may create bogus requests that contain a falsified source IP address (and the corresponding modification in the Via header) that identify a target host as the originator of the request. It will send this requests to a large number of SIP clients or servers, which are all going to reply to the target.

  Similarly, attackers might use falsified route header field values in a request that identify the target host and then send such messages to forking proxies that will amplify messaging sent to the target (Record-Route may also be used in a similar fashion).

  Unauthenticated REGISTER requests may lead to numerous DoS attacks. As stated before, an attacker may be able to de-register a user client, or register the same contact several time so that the user client gets flooded by requests. DoS by memory exhaustion is also possible if the attacker registers a huge number of contacts.
- Using client-side authentication is a good first step in preventing DoS. Moreover, to prevent standard DoS, the proxy server directly available from the public internet should not register any users, i.e., it should not be a registrar. Its role will be simply to forward the requests to the intern registrars. At worse, if the public proxy server is down, communication between user in the domain is still possible. Finally, the public proxy server should use mutual TLS.

Man-in-the-Middle Attack

- As described in [37], an attacker can easily set up a man-in-the-middle attack by using ARP spoofing/poisoning. He just has to spoof the MAC address of the SIP registrar. The attacker will receive all the requests and can modify them at will. Registration hijacking, Tampering with Message Bodies, Tearing Down Sessions are some attacks that are then easy to make. Moreover, client-side authentication will not help here as the attacker can just modify the packet and keep the challenge response intact.
- IPSec and TLS are the best solution to counter this kind of threat.

Eavesdropping

- As described in [37], an attacker can easily eavesdrop a conversation by launching a man-in-the-middle attack. Using ARP spoofing, the attacker will receive every packet destined to the attacked host.
- The best mitigation against eavesdropping is to encrypt the audio streams, by using SRTP for example.

5.1.4 Discussion

Per say, SIP has no built-in security and is a very flexible protocol. But using other secure protocols (TLS, IPSec, SRTP, S/MIME), you can enforce integrity, authentication and some confidentiality. The biggest threats SIP faces today are DoS and SPIT (Section 3). Even if some solutions have been proposed, they are still too cumbersome to deploy and are not perfect.

Here are some recommendations to deploy a SIP infrastructure. By default, you must use digest authentication to authenticate users. As seen before, this works only if your system can provide transport or network layer security (message integrity is required). The best solution is to run IPSec and use TLS to authenticate all the server components (to avoid an intern user to install a malicious server component). This will protect the signaling in your infrastructure. The session itself should also be protected. You should use SRTP instead of RTP to avoid eavesdropping and provide privacy. S/MIME is also recommended to protect the SDP part of invite requests.

If you want to support a public access, you should deploy a DMZ and install redirect servers to avoid the risk of an extern attacker hijacking registration. Moreover, these servers will protect the intern network from DoS coming from the internet. Note that DoS from authenticated clients is still possible and network traffic should be monitored to spot traffic irregularities as soon as possible. We may also point out that DoS risk highly depends on the quality of the SIP implementation (buffer overflow, parsing issues, etc.), it is thus very important to make an extensive survey of the different SIP implementation before choosing the ideal candidate.

5.2 Skype

Skype is a very popular peer-to-peer internet telephony software with more than 50 million users. Unfortunately, Skype is not open-source, and although there have been attempts to reverse-engineer certain parts of Skype [19], many algorithms remain unknown.

However, it has been conjectured that Skype has similarities with the file sharing tool KaZaA as the two projects share some developers. [17]

Communication typically happens directly between the participants in Skype. However, for name look-up operations and sometimes also for NAT-problems, peer-to-peer solutions are required. Concretely, it is possible to search the Skype network for other users, and hence to gather many user names. SPLIT attacks however are difficult: Bob can call Alice only after she has accepted his contact request and has added Bob to her friends' list.

Berson [5] has performed a security evaluation of Skype Version 1.3. He found that Skype uses standard cryptographic primitives only, e.g., AES block cyphers, RSA public key cryptosystems, SHA-1 hash functions, RC4 stream cyphers, and so on. Berson concludes that Skype is robust against identity spoofing, traffic sniffing, replay attacks or man-in-the-middle attacks, and does not seem to contain any back doors or Trojans.

Skype operates a certificate authority, and every Skype client stores the central server's public key. A user authenticates itself with a unique username and password. The traffic of each session is encrypted by a 256-bit Rijndael cypher (AES). Primality testing is done with 25 iterations of the Miller-Rabin test including all necessary test conditions. The decryption exponent (private key) is a sound Montgomery method variant of modular inversion. To protect against playback, peers challenge each other using 64-bit nonces.

However, Berson also points out some weaknesses. The CRC-type checksums are linear and may not be well-suited for detecting intentional modification of data, as has already been discovered in WEP. Moreover, a malicious program on the same machine could deduce some bits of the key by monitoring the shared resources such as CPU time and power, or storage. Finally, Berson mentions a parsing error which may lead to unpredictable behavior under malicious inputs.

5.3 H.323

H.323 [24] is a binary-based protocol standard approved by the International Telecommunication Union (ITU) which supports real-time point-to-point multimedia data communications over non-guaranteed bandwidth, packet-based networks, such as the Internet. H.323 is an umbrella specification as it encompasses various other ITU standards where the latest version (v5) was released in 2003.

In general, H.323 implementations includes four logical entities , namely:

1. H.323 Terminals
2. Gateways (GW)
3. Gatekeepers (GK)
4. Multipoint Control Units (MCU)

A H.323 Terminal provides real-time two way communication with another H.323 terminal, gateway or MCU sending multimedia messages. H.323 terminals support audio codecs for example the G.711 [25] codec and signalling using Q.931, H.245 [23] and Registration, Administration and Status (RAS) protocols. Gateways are optional components

and are only required when communicating between different networks for example between an IP-based network and Public Switched Telephone Networks (PSTNs). A Gateway provides data format translation, control-signaling translation, call setup and termination functionality as well as compression and packetization of voice. Gatekeepers are responsible for translating between telephone number and IP address and routing of calls. They also manage bandwidth and provide mechanisms for registration and authentication by terminals. All H.323 endpoints register with a single GK and build a H.323 zone. In order to support multi-terminal conferences, all terminals must establish a direct connection to an MCU.

Judging the security aspects of H.323 is difficult, as there is a plethora of associated protocols and vendor implementations. Per se, H.323 does not specify any cryptographic protocols, and several attacks have been reported, e.g. [7, 6, 35]. However, H.235 [26] gives security recommendations for the H.3xx series; its scope is on authentication, privacy and integrity. H.235 also includes the ability to negotiate services and functionality in a generic manner.

5.3.1 H.323 Security

[41] provides a concise summary of the security mechanism described by H.235. We restate here the key points.

Media Security H.235 recommend to secure the media streams by encrypting the audio stream with symmetric encryption. The encrypted stream is then encapsulated into a standard RTP packet. The encryption capabilities of the systems can be negotiated during signaling. DES, Triple DES and RC2 are intended as encryption algorithms.

Signaling Security TLS is recommended to authenticate the server components. Authentication of users is done during call control. It is done either during the initial call connection in the process of securing the signaling-channel (H.245) by support of challenge-response mechanisms or by exchanging certificates on the H.245 channel. Note that end-to-end authentication is not provided. Moreover, H.245 describes how to exchange certificates and how to use the Diffie-Hellman protocol to exchange keys. Verifying the certificate is left open.

5.3.2 Attack examples on H.323

DoS-Attack using signaling H.323 is a complex protocol suite and is therefore particularly exposed to implementation flaws. [3] reported that a DoS attack can be performed by sending unexpected or incorrect signaling PDUs.

Eavesdropping If RTP is used to transport the media, an attacker can easily eavesdrop the conversation by using ARP poisoning (man-in-the-middle attack).

Gatekeeper registration attack Registration and deregistration requests can be faked if the gatekeeper does not enforce any authentication.

6 A Sample Attack on SIP: Man in the Middle

In this section we present a sample attack on an Internet telephony application. Concretely, we show how to become a man-in-the-middle (*man-in-the-middle attack*) in the SIP VoIP system. The setting considered is as follows (cf Figure 5). Alice and Bob want to make a phone call. The attacker's aim is to become the man-in-the-middle in this connection, that is, all traffic does not flow directly between Alice and Bob, but *indirectly* via the man-in-the-middle. The attacker can therefore not only listen to the conversation and simply forward the data, but may also decide to cut important words, or replay some old sentences. For example, after having followed the conversation for some time, the attacker may have recorded a sufficiently large number of words and phrases (e.g., "yes", "no", numbers, etc.) in order to tell Bob—using Alice's voice—to pay a certain amount of money on the attacker's bank account. As SIP does not explicitly require encryption, such an attack can be achieved using *ARP spoofing* when all three clients are situated in the same local area network.

Figure 5: Setting of Man-in-the-Middle Attack. All clients are assumed to be in the same local network.

The attack works as follows. Most machines in the Internet have a hardware or MAC address (Ethernet address). However, this address is typically only visible within a local network; for global, Internet wide addressing and routing, IP addresses are used. The SIP telephony client also stores such an IP address for each contact, that is, Alice stores Bob's IP address 192.168.0.3, and Bob stores Alice's IP address 192.168.0.2. In our experiment, the attacker's IP address is 192.168.0.1. However, delivering packets to the hosts in a local network requires MAC addresses. Therefore, whenever Alice wants to communicate with Bob, she has to find out the MAC address which corresponds to Bob's IP address. This mapping from IP addresses to MAC addresses is done by the so-called *Address Resolution Protocol* (ARP). Basically, the ARP protocol is a distributed algorithm in which the query "Who has IP address X?" is broadcast in the local network, and the corresponding client with IP address X responds with the message "Hello, I have IP address X! My MAC address is Y!". This protocol can be cheated, and the attacker can become the man in the middle. In order to do so, the attacker applies ARP spoofing: When Alice broadcasts a query for Bob's hardware address, the attacker answers with his own MAC address, and similarly when Bob looks for Alice' IP address. As a consequence, both Alice and Bob bind the attacker's MAC address to the IP address of Bob and Alice, respectively.

For ARP spoofing, we have used the tool *Cain & Abel* v. 2.9.[2] The tool is shown in Figure 6. Using the network sniffer, it is possible to explore the current IP and MAC addresses in the network. The result is shown in Figure 7: The attacker has found that Alice (IP

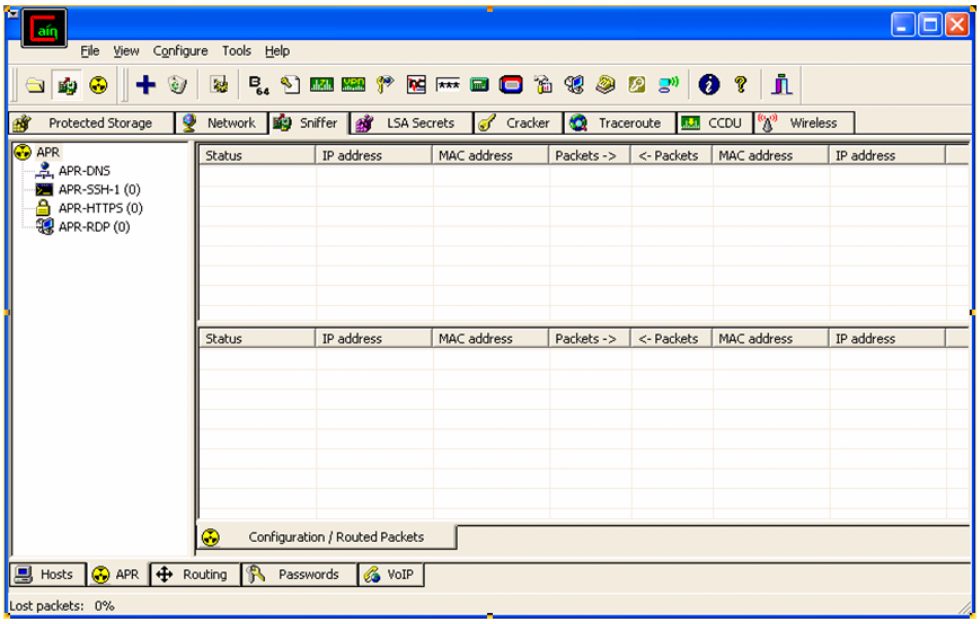


Figure 6: Cain & Abel.

address 192.168.0.2) has MAC address 00:0d:60:b0:5f:43, and Bob (IP address 192.168.0.3) has 00:0d:60:79:cb:13. The attacker can then execute the ARP spoofing/poisoning in order to become the man in the middle, see Figure 8: Cain & Abel then applies techniques to modify the ARP cache of Alice and Bob in order to become the man in the middle. Consequently, the traffic is routed through the attacker. We have used Cain & Abel to record the conversation on the attacker's machine.

The following two figures (Figures 9 and 10) show the state of Bob's machine *before* and *after* the ARP spoofing. While before the ARP spoofing took place, Bob correctly believes that Alice has the MAC address 00:0d:60:b0:5f:43, he wrongfully assumes that the MAC address is 00:08:02:e5:7e:f5 after the attack.

[2]See http://www.oxid.it.

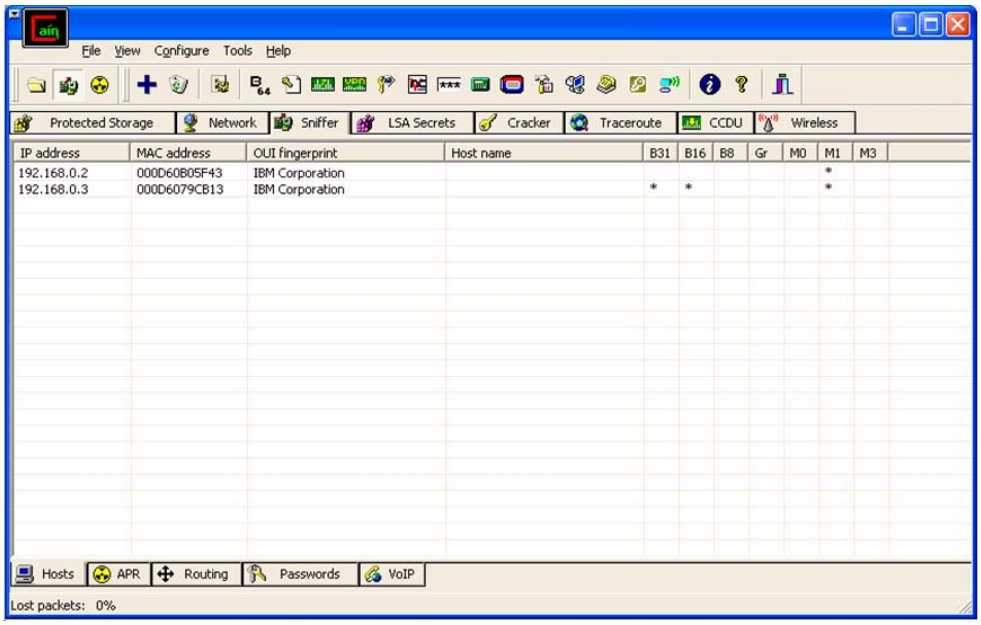


Figure 7: Exploring IP and MAC addresses with Cain & Abel.

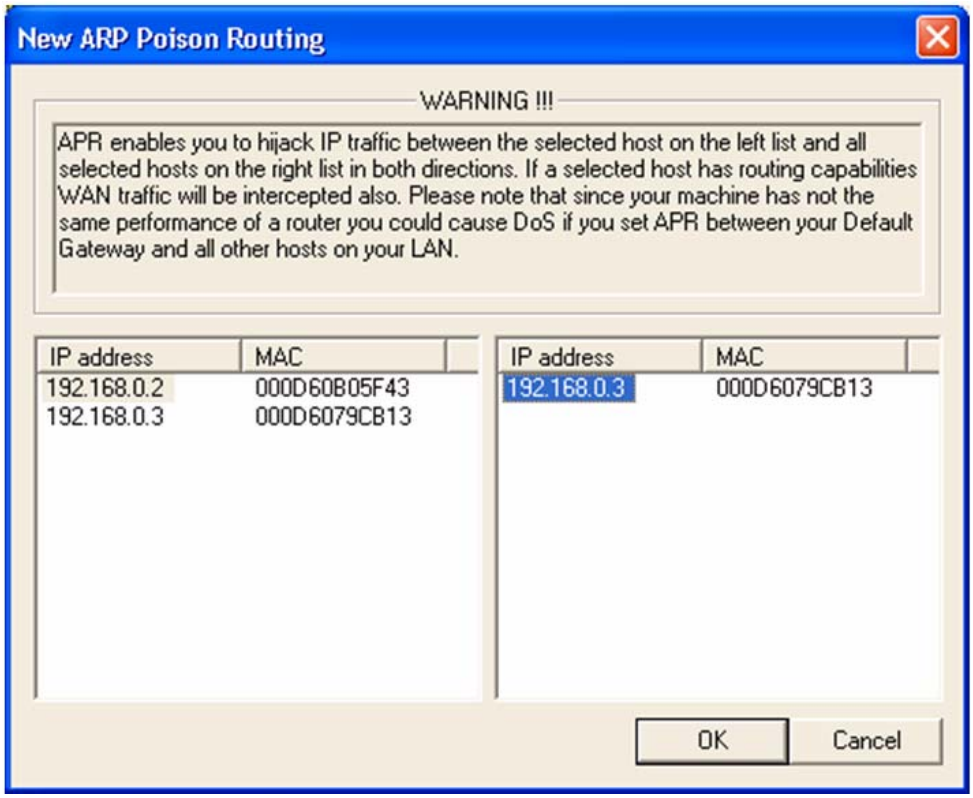


Figure 8: ARP Spoofing/Poisoning.

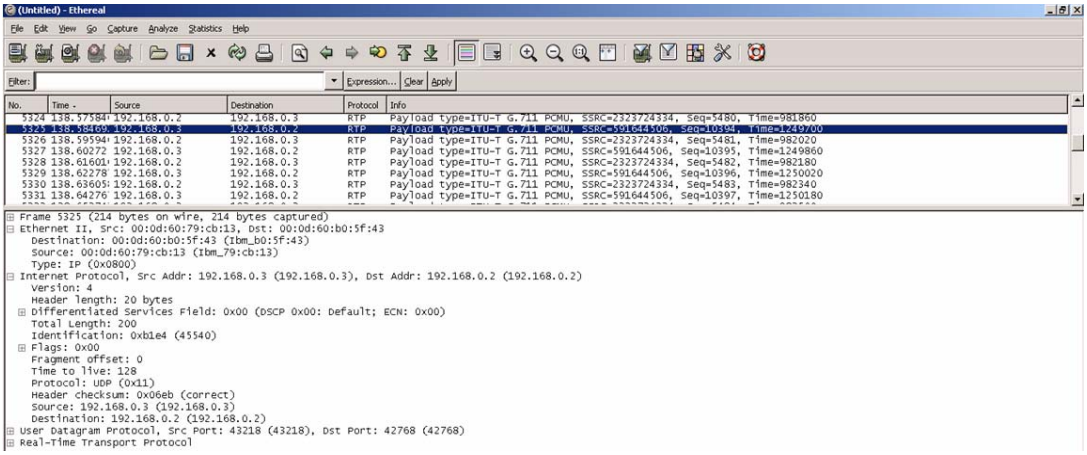


Figure 9: Ethereal Network Sniffing Tool at Bob: *Before* ARP Spoofing.

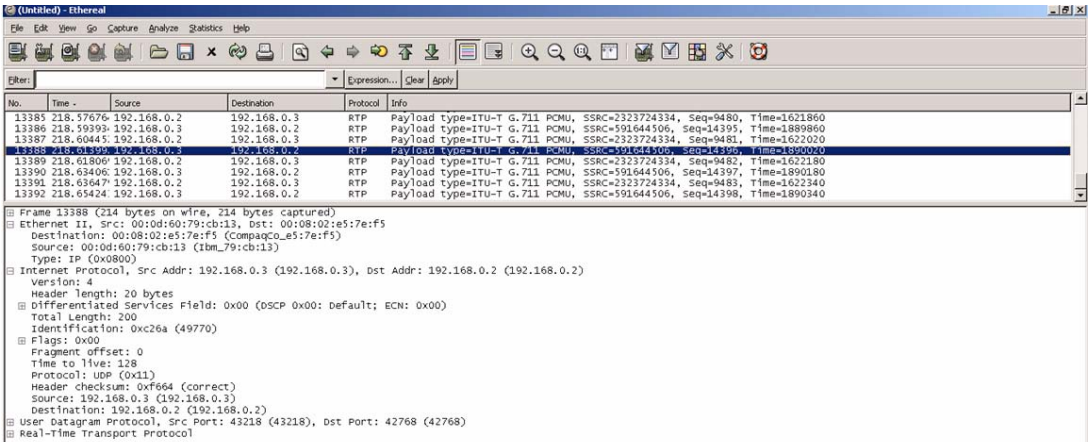


Figure 10: Ethereal Network Sniffing Tool at Bob: *After* ARP Spoofing.

7 Conclusion

The VoIP world is evolving rapidly and security issues are well discussed among researchers. However, the well-known systems (Skype, GoogleTalk, Yahoo, MSN, etc.) are closed-group systems, where a user has to be registered to interact with any other user. Efforts to interconnect these systems are currently made. This could lead to a world-wide system, where new threats such as SPIT may emerge. Yet another world-wide system may appear if ISPs are willing to provide basic VoIP functionalities for hosted domain.

This document has surveyed security issues related to internet telephony. Today's state-of-the-art VoIP applications have promising properties, and it can be expected that VoIP will more and more replace the traditional telephone system. It also seems that many products are aware of security threats and incorporate countermeasures. For instance, while in traditional telephone systems, voice was transmitted plainly, Skype uses modern cryptography to hide the contents. VoIP security is a hot topic in literature as well. In 2006 alone, two books have been published [7, 13].

7.1 Recommendations

As of today, we cannot rely only on a VoIP only solution to provide all the standard telephony functionalities we need, in particular because there is no world-wide VoIP coverage. An access to the "old" telephony network (PSTN) is still a must. Hybrid solutions exist and already offer enough security and flexibility to be deployed. In such a solution, the voice traffic is ideally routed on a different VLAN than the data traffic. Moreover, IPSec should be deployed and server components should only accept requests sent with TLS (and MTLS between two server components). To prevent unwanted communication costs, the PSTN Gateway should only be accessible via an authenticated server (MTLS) and dialing authorizations should be enforced. To prevent eavesdropping, SRTP should be used to secure the voice streams. Such a solution will prevent SPIT (or reduce SPIT to the PSTN level) and external threats. DoS attacks from the inside are possible, but the threat already exist on the data network nowadays. Traffic monitoring should be used to stop such attacks as soon as possible.

In 2-3 years, global VoIP system will emerge and SPIT might be a big issue. It is most probable that such a global VoIP network will be run by several organizations, which will be responsible for user management (registration, accounting). SPIT can be mitigated by deploying an authentication and reputation system between the organizations. This will enable the use of *Payment* schemes. Moreover, in such scenarios, back-end servers should be used to filter the traffic using *Volume based Models* or *Statistical Pattern and Anomaly Detection* methods. Following the defense-in-depth paradigm, endpoints should also run anti-SPIT methods. In the general interest of the community, *Cryptographic Puzzles and Computation Challenge* should be used to weaken the power of any malicious user. *White and Black Lists* may also be used if the authentication mechanism in place provide strong identity.

7.2 Challenges for the Future

There are still several security problems, and many VoIP hacking tools can be found online, e.g., on http://www.hackingvoip.com/sec_tools.html. On the other hand, there are products such as Skype which are not open source, making it hard to find and repair security flaws. We believe that there are inherent trade-offs, for example related to SPIT: More intensive filtering may threaten the availability of communication partners. It has also been pointed out that many VoIP components use Web servers for configuration, and that the corresponding development tools often lack security features. Finally, the fact that machines today are increasingly well protected by firewalls, demilitarized (DMZs), etc., complicates many aspects of VoIP [52].

References

[1] M. Abadi, M. Burrows, A. Birrell, F. DAbek, and T. Wobber. Bankable Postage for Network Services. In *Proc. 8th Asian Computing Science Conference*, 2003.

[2] M. Abadi, M. Burrows, M. Manasse, and T. Wobber. Moderately Hard, Memory Bound Functions. In *Proc. NDSS*, 2003.

[3] Ralf Ackermann, Markus Schumacher, Utz Roedig, and Ralf Steinmetz. Vulnerabilities and Security Limitations of current IP Telephony Systems. In *Proceedings of the Conference on Communications and Multimedia Security (CMS 2001), Darmstadt, Germany*, volume 192, pages 53–66. Kluwer Verlag, May 2001.

[4] Keno Albrecht, Nicolas Burri, and Roger Wattenhofer. Spamato - An Extendable Spam Filter System. In *Proceedings of the Second Conference on E-mail and Anti-Spam (CEAS)*, 2005.

[5] Tom Berson. Skype Security Evaluation. Technical report, ALR-2005-31, 2005.

[6] Christoph Bronold. Hacking IP-Telefonie. In *Talk at IT-Symposium*, 2005.

[7] L. Chaffin, J. Kanclirz, T. Porter, C. Shim, and A. Zmolek. *Practical VoIP Security*. Syngress, 2006.

[8] N.J Croft and M.S Olivier. A Model for Spam Prevention in IP Telephony Networks using Anonymous Verifying Authorities. University of Pretoria, 2005.

[9] E. Katz H. Tschofenig D. Schwartz, B. Sterman. SPAM for Internet Telephony (SPIT) Prevention using the Security Assertion Markup Language (SAML). Internet-Draft, 2005.

[10] Ram Dantu and Prakash Kolan. Detecting Spam in VoIP Networks. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, pages 31–37, Cambridge, MA, 2005.

[11] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), January 1999. Obsoleted by RFC 4346, updated by RFC 3546.

[12] John R. Douceur. The Sybil Attack. In *Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 251–260. Springer-Verlag, 2002.

[13] D. Endler and M. Collier. *Hacking VoIP Exposed*. McGraw-Hill, 2006.

[14] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), June 1999.

[15] H. Tschofenig G. Dawirs, T. Froment. Authorization Policies for Preventing SPIT. Internet-Draft, 2006.

[16] J. Galvin, S. Murphy, S. Crocker, and N. Freed. Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted. RFC 1847 (Proposed Standard), October 1995.

[17] Simson L. Garfinkel. VoIP and Skype Security. Technical report, Skype Security Overview Rev. 1.5, 2005.

[18] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard), January 1996. Obsoleted by RFC 3550.

[19] Saikat Guha, Neil Daswani, and Ravi Jain. An Experimantal Study of the Skype Peer-to-Peer VoIP System. In *Proc. 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2006.

[20] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327 (Proposed Standard), April 1998. Updated by RFC 3266.

[21] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. RFC 2543 (Proposed Standard), March 1999. Obsoleted by RFCs 3261, 3262, 3263, 3264, 3265.

[22] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280 (Proposed Standard), April 2002. Updated by RFC 4325.

[23] International Telecommunication Union. Control protocol for multimedia communication. Recommendation H.245, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, February 1998.

[24] International Telecommunication Union. H.323 extended for loosely coupled conferences. Recommendation H.332, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, September 1998.

[25] International Telecommunication Union. Pulse code modulation (PCM) of voice frequencies. Recommendation G.711, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, November 1998.

[26] International Telecommunication Union. Security and encryption for h-series (H.323 and other h.245-based) multimedia terminals. Recommendation H.235, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, February 1998.

[27] C. Jennings J. Peterson. Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP). Internet-draft, 2005.

[28] D. Willis J. Rosenberg, G. Camarillo. A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP). Internet-draft, 2006.

[29] C. Jennings. Computational Puzzles for SPAM Reduction in SIP. Internet-draft, 2006.

[30] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[31] R. MacIntosh and D. Vinokurov. Detection and Mitigation of Spam in IP Telephony Networks Using Signaling Protocol Analysis. In *Advances in Wired and Wireless Communication, 2005 IEEE/Sarnoff Symposium on*, pages 49–52, April.

[32] P.C.K. Martin, M.V.; Hung. Towards a Security Policy for VoIP Applications. In *Electrical and Computer Engineering, 2005. Canadian Conference on*, pages 65–68, May.

[33] Bogdan Materna. A Proactive Approach to VoIP Security. White paper, CTO VoIPShield, April 2006.

[34] S. McGann and D. Sicker. An Analysis of Security Threats and Tools in SIP-Based. In *2nd Annual VoIP Security Workshop*, 2005.

[35] Microsoft. Microsoft Security Bulletin. In *MS04-001*, 2004.

[36] Saverio Niccolini. SPIT Prevention: State of the Art and Research Challenges. In *NEC*, 2006.

[37] Peter Thermos. Two attacks against VoIP. http://www.securityfocus.com/infocus/1862, 2006.

[38] Till Andreas Radermacher. Spam Prevention in Voice over IP Networks. In *Diploma Thesis*, 2005.

[39] Till Andreas Radermacher. Spam Prevention in Voice over IP Networks. Master's thesis, University of Salzburg, November 2005.

[40] B. Ramsdell. S/MIME Version 3 Message Specification. RFC 2633 (Proposed Standard), June 1999. Obsoleted by RFC 3851.

[41] C. Rensing, U. Roedig, R. Ackermann, and R. Steinmetz. A survey of requirements and standardization efforts for iptelephony -security, 2000.

[42] Thomas Rohwer and Carsten Tolkmit. Abwehr von "Spam over Internet Telephony" (SPIT-AL). White Paper, 2006.

[43] J. Rosenberg, C. Jennings, and J. Peterson. The Session Initiation Protocol (SIP) and Spam. draft-ietf-sipping-spam-02, March 2006.

[44] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320.

[45] M. Stiemerling S. Niccolini, S. Tartarelli. Requirements and methods for SPIT identification using feedbacks in SIP. Internet-draft, 2006.

[46] David Schwartz and Baruch Sterman. SPIT (SPAM for Internet Telephony) Prevention Security Model. White Paper, Kayote Networks, June 2005.

[47] Dongwook Shin and Qovia Choon Shim. Voice Spam Control with Grey Leveling. In *2nd Workshop on Securing Voice over IP*, 2005.

[48] Sipera Systems, Inc. Comprehensive VoIP Security for the Enterprise: Not Just Encryption and Authentication. Sipera Whitepaper, http://www.sipera.com/, 2006.

[49] Kumar Srivastava and Henning Schulzrinne. Preventing Spam for SIP-based Instant Messages and Sessions. Technical report, Department of Computer Science, Columbia University, New York, NY, Technical Report CUCS-042-04, October 2004.

[50] Jennifer Steiner, Clifford Neuman, and Jeffrey Schiller. Kerberos: An Authentication Service for Open Network Systems. In *USENIX Association Winter Conference 1988 Proceedings*, pages 191–202, February 1988.

[51] Baruch Sterman. Proposal for a SPIT Prevention Security Model. White Paper, Kayote Networks, Marc 2005.

[52] T. J. Walsh and D. R. Kuhn. Challenges in Securing Voice over IP. In *IEEE Security & Privacy*, 2005.