

# Voice Over IP: Unsafe at any Bandwidth?

Joachim Posegga, Jan Seedorf

*Security in Distributed Systems (SVS)  
University of Hamburg, Dept of Informatics  
Vogt-Kölln-Str. 30, D-22527 Hamburg  
svs-office@informatik.uni-hamburg.de*

## Abstract

*Voice over IP (VoIP) is promising a silver bullet for future voice services. There are several technical aspects which make the technology attractive; it is in particular believed to reduce operating costs and increase flexibility by converging networks.*

*This paper offers a technical analysis of the security aspects of VoIP; we discuss the major differences and implications of VoIP, in particular compared to circuit-switched voice as it is deployed today by network operators. We will concentrate on the “signalling” part of VoIP, based on the Session Initiation Protocol (SIP). Our analysis considers primarily consumer scenarios, rather than VoIP deployment by/for business customers.*

## 1. Introduction

Delivering voice services over packet-switched IP has radically different security characteristics than circuit-switched operation: the business of operating a public telephone network consisted in the past of controlling the routing in the core network, and switching lines (with guaranteed QoS) between the network's end points. The overall security of the system was based on the (physical) security of the core network components and the last mile. The “Principals” involved in a connection were practically the sockets in the walls on the customer's premises, and access to a socket implies authorisation to use the network. This model works (pragmatically seen) sufficiently well, largely due to the fact that attacks hardly scale in a distributed environment that requires physical access to exploit vulnerabilities. Furthermore, the signalling channel of the network is separated from the payload channel, which makes it easier to prevent manipulation from terminal systems.

## 2. VoIP: High-level Security Observations

The Internet comes with a radically different architecture: the route that information takes is not centrally controlled; also it offers mobility of terminal devices to the overall scenario, because the use of services is not necessarily bound to a particular location in the network. Next, the end points of communication are complex systems rather than dumb terminals. Thus: the intelligence moves from the network to its end points – not even a logical state is left. There is not a separate signalling channel, all signalling is in-band and accessible to terminal devices.

When delivering a voice service over IP, the first and most fundamental security issue is authentication as a prerequisite for authorisation: Clearly, the physical network end points cannot be used any more, unless the added value of location- and network-independence of VoIP clients is sacrificed. Options to consider are to authenticate users, or their terminals. Since physical location as in POTS is irrelevant, the only choice seems to use suitable credentials, most likely in terminals: authentication of users is probably not a realistic option for reasons of convenience.

The security characteristics of VoIP beyond authentication are also significantly different from the POTS: The lack of a central network/routing control makes it hard to embed security within the network, and,

due to the „open“ nature of a shared medium, traffic is accessible (and attackable) for all parties unless dedicated protection measures are applied. The privacy and availability threats resulting from this are significant, and will also be discussed in the course of this paper.

VoIP also needs to be interfaced with POTS: This means some sort of mapping between signalling in the Internet, and in SS7. The interoperability of an isolated and per se insecure signalling system like SS7 and the signalling in an open system like the Internet requires protected gateways. Security implications and corresponding protection of such a gateway need to be carefully considered. Addressing this in detail, however, goes well beyond the scope of this paper.

The remainder sections are structured as follows: After an introduction to the SIP protocol (3), current SIP security mechanisms are examined (4). Security weaknesses and problems of SIP are discussed in (5). In (6) solutions to several problems are presented. A summary and outlook concludes the paper (7).

### 3. Introduction to the Session Initiation Protocol

The signalling part of VoIP traffic is usually based on either of two standards H.323 [1] and SIP [2]. Both offer approximately the same functionality (a good comparison of H.323 and SIP can be found in [3]). H.323 has been developed prior to SIP by the ITU, but it seems that SIP is going to dominate the market: Recently, SIP has been selected as the standard for signalling in UMTS Release 5, and most providers that offer VoIP-connections for consumers base their solutions on SIP [4,5]. This paper therefore focuses on the security implications of signalling in VoIP networks using the SIP protocol.

The Session Initiation Protocol (SIP) was originally specified by the IETF in 1999 [6] as a standard for signalling and control in multimedia communications over IP. It was revised and enhanced in 2002 [2,7].

SIP uses the Session Description Protocol (SDP [8]) to select the type of media and to negotiate a codec for media transmission. After a session has been established with SIP, the actual media transfer is based on the Real-time Transport Protocol (RTP [9]). Security of RTP is not discussed here.

#### *SIP Messages*

SIP is a client-server protocol similar to HTTP. Signalling in SIP is based on (ASCII compatible) UTF-8 text messages: A message consists of a message header and an optional message body. Messages can be either *requests* or *responses*. The original RFC 3261 states six types of requests (also called *methods*): INVITE, BYE, ACK, OPTIONS, CANCEL, and REGISTER. Table 1 provides a description for each request. Other requests have been added to SIP in additional RFCs to provide more functionality (e.g. for event subscription and notification, session transfer, etc.) [10].

SIP Request	Description
INVITE	Initiates a call signalling sequence
BYE	Terminates a session
ACK	Acknowledge
OPTIONS	Queries a server about its capabilities
CANCEL	Used to cancel a request in progress
REGISTER	Used to register location information at a registrar

**Table 1 SIP Requests**

SIP Response Codes
1xx – informational
2xx – ok
3xx – redirection
4xx – client error
5xx – server error
6xx – global failure

**Table 2 SIP Response Codes**

If a SIP entity receives a request, it performs the corresponding action and then sends back a response to the originator of the request. Responses are three-digit status codes (as in http/1.1), categorised into 6 classes. Table 2 lists these classes for response codes. Concrete examples for response codes are “180-ringing”, “302-moved temporarily”, or “404-not found”.

#### *SIP Addressing*

To ease integration of SIP in existing internet applications, addressing is based on a uniform resource identifier (URI). A SIP-URI is similar to an e-mail address and generally of the type “sip:user@domain”.

Instead of a domain, it can also contain a static IP-address. DNS can be used to map SIP-URIs into IP-addresses.

### SIP Entities

SIP defines five types of (logical) entities: *user agent*, *proxy*, *registrar*, *redirect server*, and *location server*. *User agent* designates any terminal (hardware or software) participating in SIP-communications. A *proxy* receives requests or responses and forwards them to another server or user agent. A *redirect server*, when receiving a message, tells the sender of the message where to send the message, rather than forwarding it. Users can register their current location with the *registrar* of their domain to facilitate mobility: A *location server* is used by a registrar to store the location of users (the binding of a SIP-URI with a current IP-address). Other SIP entities (proxies or redirect servers) can use the location server to look up the current location of SIP users.

### Example: Setting up a Simple Voice Connection with SIP

The set up and cancellation of a voice connection between two users is illustrated in Figure 1: It shows the messages (requests and responses) that are being exchanged if user agent A wants to initiate a session with user agent B. Both user agents use here the same proxy:

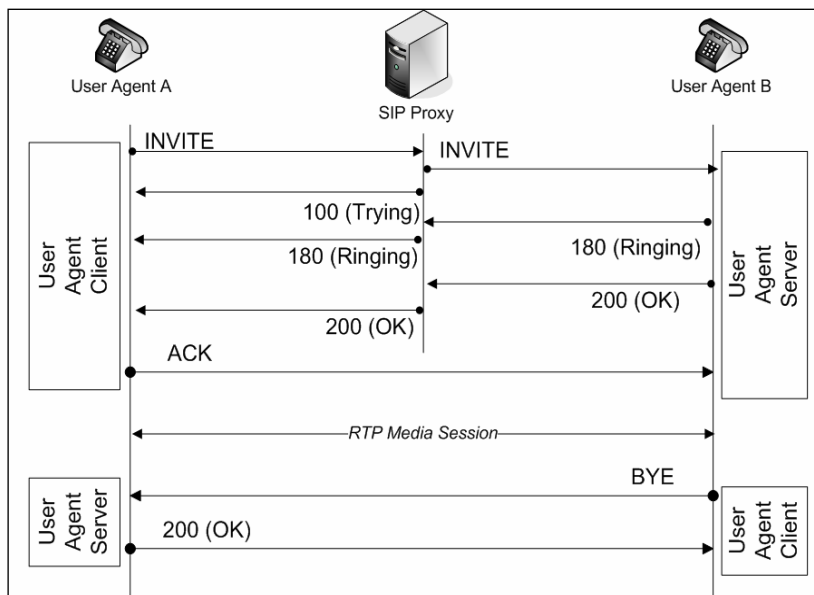
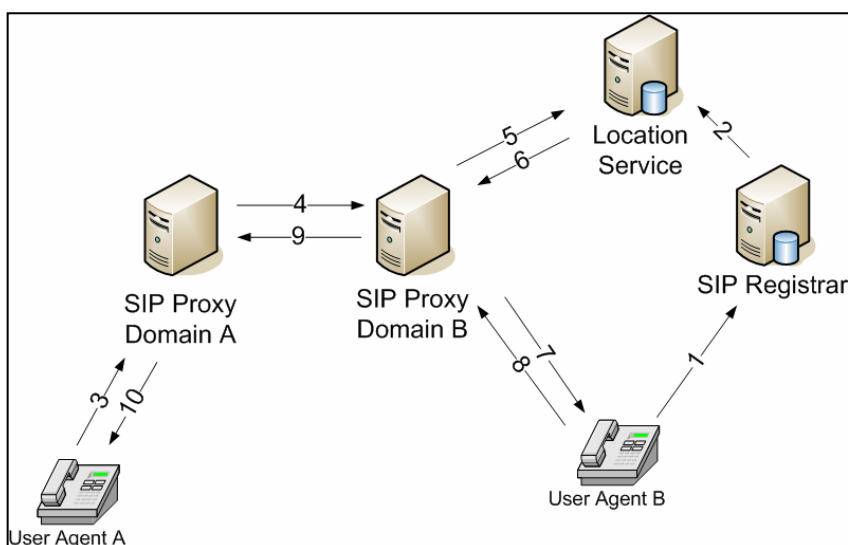


Figure 1 SIP Requests and Responses

User agent A starts the request (INVITE), the proxy passes it on to the receiver (user agent B) and sends back a 100 (trying) response message. User agent B responds to the request first with a 180 (ringing) message, and eventually with a 200 (ok) after a user has picked up the phone. Both of these messages are forwarded to user agent A by the proxy. User agent A can now request the start of the media transfer (ACK). N.b. the proxy is not needed for this: after a session has been established, user agents can communicate directly. At the end of the conversation, some user agent (here: B) terminates the session by sending a BYE request to its counterpart.

Figure 2 shows a similar situation in order to illustrate the idea behind the various SIP entities: In this example, user agent A and B are in different domains and have different proxies. First, the callee (user agent B) needs to register with its local registrar (1) to be able to receive calls. The registrar stores the location information at a location server (2). When user agent A wants to call user agent B, it sends an INVITE-request to its local SIP-proxy (3) which passes on the request (possibly after a DNS lookup) to the proxy of user B's domain (4). The proxy in domain B needs to look up the IP-address of user agent B at the location server (5, 6) before it can send the request to user agent B (7). In this example, the response message for user agent A takes the same route back (8, 9, 10), possibly for billing purposes.



**Figure 2 Session Establishment of a Voice Connection with SIP**

## 4. SIP Security – state of the art

### *Threats to session initiation using SIP*

It is well known that the SIP protocol, without any dedicated security mechanisms, can be attacked in several ways. The SIP specification acknowledges five possible attacks (cf. [2], pp. 232-236):

*Registration hijacking:* Registration authorization in SIP is implemented by the registrar, which checks the FROM header field in the registration message. As the FROM header field in a SIP-message can easily be altered by any user agent, malicious registrations are possible.

*Impersonating a SIP server:* Any redirect server can impersonate another SIP-server by forging answers to requests from a user agent to this other server, thereby redirecting future calls to itself, or to any other SIP server.

*Tampering with message bodies:* Any proxy server on the route from one user agent to another user agent can read messages and also modify headers as well as message bodies.

*Tearing down sessions:* An attacker capable of sniffing and sending packets on an IP-network can also tear down SIP-sessions by sending BYE or CANCEL messages to either of the participating parties. This can be done by forging the FROM header field, thus claiming the message is coming from one of the participants of the session.

*Denial of service:* The availability of any SIP-server or user agent can be threatened by an exorbitant number of requests coming from an attacker, eventually exhausting the capacity of the targeted server/client to answer requests.

Besides these, there are other obvious attacks:

*Call hijacking:* A spoofed “moved permanently” message can be used by an attacker to redirect calls to another user agent or server.

*Client impersonation:* A malicious attack on a client could install a Trojan horse which impersonates the client. Such an impersonated client can be used by the attacker to send out (authenticated) calls or registrations from the client.

*Eavesdropping:* If an attacker can sniff packets on the network, he has also access to the content included in the header and body of a SIP-packet, unless encryption is used.

*Spam:* Spam-messages are possible, where servers or clients send out automatically generated packets to user agents.

*Traffic logging and Analysis:* Any SIP-server (and any network node) on the route between user agents can log the messages it receives during a session. Some users might not want their connections to be traceable on the entire route from sender to receiver. Nevertheless, this might partly be necessary for billing purposes.

The following security requirements for SIP-messaging can be derived from the attacks described above in order to establish a session securely:

- Message Confidentiality (to prevent reading of message headers/bodies)
- Message Integrity (to prevent alteration of message bodies)
- Authentication of SIP nodes (to prevent spoofing, server impersonation, registration & call hijacking, tearing down of sessions)
- Availability (of user agents and SIP servers)
- Privacy (of SIP-traffic)

### ***SIP Security Mechanisms***

The SIP standard, as specified in RFC 3261, includes several security mechanisms in order to establish a secure session (cf. [2], pp. 236-240). These suggested security mechanisms can be distinguished by the scope of security they provide: either end-to-end or hop-to-hop:

#### *End-to-end security*

Digest Authentication: Principals sharing a secret (e.g. a password) can mutually authenticate - with a challenge-response authentication. To prevent replay attacks, this challenge-response authentication includes nonces.

S/MIME: Because SIP is using MIME for message bodies, S/MIME [11] can be used to send authenticated and encrypted messages between user agents. Thus, S/MIME can provide end-to-end confidentiality, integrity, and authentication for SIP.

#### *Hop-to-hop security:*

TLS: The transport layer security protocol [12] supports encryption and authentication of packages over connection-oriented protocols (e.g. TCP). The SIPS URI Scheme ([2], pp. 147-157), an extension to SIP similar to HTTPS, uses TLS: A request sent to a SIPS URI is successful if a TLS connection is available on every hop on the route; otherwise, the connection will fail.

IPSec: IPSec [13] can be used to secure the session initiation process on the network layer. IPSec offers authentication, integrity, and confidentiality of packets. Key exchange for IPSec can be done with the IKE (internet key exchange) protocol [14].

## **5. Weaknesses and Problems of the Proposed SIP Security Mechanisms**

Confidentiality, integrity, and authentication can, in principle, be provided by the security mechanisms above. In practice, however, these mechanisms are hard to deploy. Furthermore, privacy and availability remain as open challenges in VoIP session initiation using SIP.

Specifically, the following areas are insufficiently addressed:

### ***Authentication***

VoIP comes with a completely different setting than POTS: The physical location of clients is not fixed, thus authentication is required as a basis for authorising service use. A realistic concept probably requires some form of a secret for authentication, as relying upon the network topology (as POTS does) would sacrifice mobility of terminals. Such a secret needs to be stored in the terminal platforms: direct user authentication for each communication session is likely not acceptable for users. But it is anything but

clear how to protect complex computing platforms (VoIP clients) against compromising the secrets they hold (see “platform security” below).

When thinking rigorously about authentication, there is also the question of *what* to authenticate: Is it the client platform itself (hardware or software?), the network interface, or possibly the process running in the client that shall be authenticated?

### ***The Necessity of a (Globally Interoperable) Certificate Infrastructure***

For mutual authentication, some form of certificate management will be needed. For the connection of phone calls on a global scope, certificate exchange among different public key infrastructures (alternatively infrastructures based on symmetric cryptography) must be possible. Also, a unified mechanism for certificate revocation has to be in place. However, experience with large PKIs has shown that deploying this is a major undertaking under “real life conditions”– if feasible at all.

### ***What does a Certificate Mean in Practice?***

Certificates bind names (identities) to key pairs and ease authentication, but they do not necessarily assure trustworthiness of the authenticated principals: For instance, a certificate will usually not provide any assurance about the security policy or the protection against malicious intrusion the authenticated principal has in place. In other words, mutual authentication on the entire route between user agents does not guarantee a standard of security functionality at proxies and redirect servers. Another example is spam (see below): even authenticated parties might be the source for spam messages.

### ***Denial of Service***

Security mechanisms for SIP do not, and probably cannot, provide protection against denial-of-service (DoS) attacks in general. Neither TLS nor IPSec are of any help here. SIP servers will be an easy target for denial-of-service attacks, since they need to be publicly available. Distributed denial-of-service attacks are common to HTTP web servers and have led to loss of revenue for many companies. SIP-proxies will likely attract similar attacks. In addition, terminals will have only limited protection against denial-of-service attacks, but user agents should be available for emergency calls.

### ***Spam***

VoIP spam differs from e-mail spam in that it is significantly more obtrusive (a phone will actually ring with every spam message, possibly in the middle of the night). Furthermore, E-mails get “pulled” from a server by the user, while VoIP calls are “pushed” to the user. Authentication can only provide limited protection against spam: Certificate authorities would need a policy that revokes certificates of servers that are used for spamming, and they would need to do so very quickly.

### ***Platform Security***

The “intelligence” needed to establish VoIP connections is in terminals and not within network as in the POTS: thus, terminals need a certain degree of computational power in order to establish connections. It is obvious that such systems will have vulnerabilities in their implementation, and the programming of malicious software for such devices is feasible and it will happen. A Trojan horse installed on a user agent can result in impersonation of that client by an attacker; a series of impersonated user agents, for instance, can be used to execute a distributed denial-of-service attack on a particular SIP-server.

Malware threats exist not only for terminals: also a compromised server can redirect calls, read message headers and bodies, alter message headers and bodies, or tear down sessions at will, just to name a few threats. Furthermore, worms targeting VoIP systems and exploiting vulnerabilities could spread from phone to phone (or server to server), just like internet worms on PC operating systems.

### ***Privacy***

Some headers of SIP messages must contain information about the sender’s identity (e.g. contact header). Thus, any SIP-server can log information about connection requests and connection termination. Although users might accept this on some proxies (e.g. for billing purposes), it can be done by any SIP-server on the route of a SIP-request.

In addition, VoIP providers can analyze traffic coming through their proxies. This does not only enable providers to analyze their own traffic. They can also analyze traffic with any other provider's domain as origin or destination.

### ***NAT Gateways and/or Firewalls***

Customers using VoIP through their broadband connection will use Network Address Translation (NAT) and have several phones/PCs with one IP-address (seen from "outside"). But NAT presents a hurdle for incoming calls: the caller needs to know the IP address and the callee's port (cf. [15], pp. 46-57). This means that users cannot easily connect more than one IP-phone to a router that uses NAT. For the transmission of media traffic, RTP uses arbitrary UDP-ports. This makes it impossible to define static rules to let RTP traffic pass a firewall while filtering other traffic.

### ***Terminal User Interfaces***

The user interface of an IP-phone must be simple: customers are used to the simple terminals of the POTS. Nevertheless, terminal devices must validate certificates for authentication. It is not obvious how to do this in practise: If, for instance, terminals have a hard-coded public key for certificate verification they are limited to connections over certain proxies unless a world-wide CA will eventually exist. Thus, they cannot be used with any provider like the POTS terminals the customer is accustomed to. Furthermore, what happens if the corresponding private key has been compromised?

From a pragmatic point of view, solutions to other questions have to be found: What happens if a certificate is not valid anymore (e.g. revocation, expiration) or cannot be verified? Are users expected to understand the consequences of a false certificate and prompted each time this happens?

## **6. Approaches to SIP security problems**

This section presents some approaches to solutions for the SIP-security problems described in the previous section.

### ***Authentication***

Authentication is the core of SIP security, and indispensable for users and providers (e.g. for billing). The technical challenge is to securely authenticate users/customers in a scenario that provides (network) mobility to these users. From a security perspective, this is very similar to the GSM/UMTS setting, with the additional threat of permanent, high-bandwidth Internet-connections and (most likely) highly vulnerable terminal devices.

Authentication will likely use credentials, preferably stronger ones than user-chosen passwords. A smart card-based solution similar to today's mobile networks would be very suitable: anything relying on platform security of clients will probably be broken quickly under "Internet circumstances". Other kinds of security modules (Trusted Computing/TPMs) could be used as well.

As of today, it seems unlikely that reasonably strong authentication will be seen in SIP/VoIP solutions for the consumer market. The implications of this should carefully be considered by service providers.

### ***Denial of Service***

DoS-attacks are common in the Internet, and little can be done to generally solve the problem of DoS or DDoS attacks in IP-networks. This means that availability for SIP-servers will be as hard to achieve as for HTTP-servers. Furthermore, terminal protection against DoS-attacks can, if at all, only be done by the terminal's network (e.g. router, firewall, intrusion detection system). This implies to deploy such technologies at the end user's network, which is hard and expensive.

### ***Platform Security***

SIP-servers are likely to be implemented on platforms as insecure as operating systems like Windows or Linux. Numerous vulnerabilities for these systems exist and are known. Thus, SIP-servers need sophisticated protection against malicious attacks (e.g. firewalls, intrusion detection systems, anti-malware-software) and must be patched on a high frequency.

Several vulnerabilities have already been published for H.323 terminals [16] and IP-telephones (e.g. [17]). Furthermore, it can be possible to capture credentials for client impersonation on softphones<sup>1</sup>, due to the insecurity of the underlying operating system. Thus, there is a high risk of malware -namely Trojan horses or phishing worms- for softphones or also dedicated terminals. The same countermeasures as for servers (see above) should therefore be taken for terminals.

Even with these countermeasures, terminals and servers must be considered vulnerable.

### ***NAT/Firewall Traversal***

Several approaches exist to solve the problem of middlebox-traversal of VoIP-traffic, (cf. [3], pp. 1514-1516):

- *Proxy placement between two domains*: A dedicated proxy for SIP and RTP traffic is placed between two networks that are separated by a NAT/firewall, all other traffic is directed to the NAT/firewall.
- *Application level gateway*: An application level gateway (as part of a NAT/firewall) can handle VoIP protocols and make the necessary modifications for packets to traverse the middlebox.
- *Control proxy*: With this approach, the NAT/firewall is controlled by an external device. This external proxy monitors VoIP connections and commands the NAT/firewall in a way to let this traffic pass through (cf. [18]).
- *External proxy with persistent connections to VoIP devices*: To traverse a NAT/Firewall without any modifications to the middlebox, terminals can frequently send probe packets to an external server. The server can then send back to the terminal the IP-address and port information it received, enabling the terminal to determine its NAT-binding. An example is the STUN (simple traversal of UDP through NATs) protocol [19]. Most VoIP providers for consumers use this kind of solution (e.g. [5]).

The first three approaches require changes to the architecture of a network perimeter while the last approach works without changing the middlebox configuration at all. This makes the last approach the one of choice for wide deployment of VoIP for consumers.

### ***Privacy***

Users could encrypt message bodies and anonymise some header fields to gain a certain level of privacy. However, if a request gets routed through one or more intermediaries, users cannot fully conceal their identity: Some headers must contain information about the user so he can receive messages within the current session. The only option to achieve privacy in SIP communications is to disguise all headers by using some form of a *pseudonymity service*. A pseudonymity service replaces sensitive headers before passing messages on; it acts as a transparent user agent in both directions. By saving the binding of the replacement in the outgoing direction, a pseudonymity service can route incoming SIP-messages to a masked identity. See [20] for an enhancement to SIP for such privacy-preserving services.

To ensure confidentiality in media transmission, users can encrypt their media transmission once a session has been established. Furthermore, users can communicate directly with each other for media transmission. Users can thereby not only circumvent providers' logging of session cancellation but also prevent a provider from logging the actual media transmission data.

### ***Lawful Interception***

Lawful interception can be seen as an authorized privacy breach. In a SIP/VoIP telephony setting, two fundamental problems arise for LI-interfaces: First, SIP handles only the signalling part of a voice call, and it is not connected to the actual media transfer. Second, the network provider is often different from the SIP provider; even worse: due to the "mobility" of VoIP clients, both providers are not even statically bound to each other.

These characteristics are an immense technical challenge for providing lawful interception: It meant to technically link the SIP session establishment to the subsequent media stream, without knowing its route -

---

<sup>1</sup> As a proof-of-concept, we have been able to copy the windows registry-entry of a popular SIP user agent software. This resulted in a successful impersonation of a SIP-registration at a German SIP-provider: the user agent with the copied windows registry-entry could make phone calls with the captured identity.

or even the provider(s) used for the transport- ahead of the session set up. We do not see a realistic way to implement this in real-world scenarios and meet the (distributed) real time requirements.

The only other option seems to require a suitable "footprint" in terminal devices for LI, which would also "solve" the problem of reading encrypted media streams: terminals are the only nodes where media stream and signalling are (necessarily) connected. But this meant a completely new scenario for LI, which is probably even harder to deploy.

## **Spam**

An overview of potential solutions against SIP-spam is given in [21]. Some of the most effective anti-spam techniques used on e-mail spam are hardly any use against VoIP-spam: *whitelisting*, *blacklisting*, and *content filtering*. Using *whitelisting* would limit the set of callers to those on the white list: a VoIP user would not receive calls from someone not on the whitelist. *Blacklisting* is only of limited help because spammers can either forge SIP-addresses, or (if authentication prohibits message spoofing) create new SIP-addresses not on the blacklist. For *content filtering*, a semantic analysis of real-time audio traffic would be necessary. With today's technology, this is not feasible. Thus, while content-filtering is very effective on e-mail spam, it cannot be used on VoIP-spam.

More sophisticated methods against VoIP-spam include *payments-at-risk*, *memory bound functions* and *turing tests*. *Payments-at-risk* is a concept that relies on micropayment. Unfortunately, there is no micropayment standard in the internet today, but a connection to the existing telephone networks could solve this problem. *Memory bound functions* would consume computing power at the sender's device for each SIP-message being sent, making huge amounts of spam in a short period of time expensive, while not bothering the average user. A *turing test* is a challenge for a sender intended to distinguish automatically generated messages from human interaction. For example, a caller could be asked to enter the result of a numerical calculation into his phone. Though these methods pose some additional problems (see [21] for details), they appear promising to be used against VoIP-spam.

Authentication is only a partial solution to VoIP-spam because it can only provide assurance about a sender's identity, not about a sender's trustworthiness respective to spamming. To be effective against spam, authentication needs to be used in conjunction with an anti-spamming policy which is enforced at all participating principals.

## **7. Conclusion**

We have reviewed the basic characteristics of VoIP based on SIP and provided a security analysis of such systems. Our analysis was concentrated on SIP and highlighted in particular the differences to conventional telephone systems deployed into a consumer market.

Our conclusion is that many essential security requirements for providing voice services in public networks cannot be met by today's VoIP/SIP technologies:

- The very basic problem of user authentication is insufficiently addressed, thus any security property that relies upon user/terminal authentication will be hard to achieve. The reason for this is, simplified, that VoIP provides user/terminal mobility, a "feature" which lacks "compensation" in terms of authentication technology.
- Moreover, VoIP/IP inherits most well-known Internet-threats, such as network-based attacks (e.g. Denial-of-Service), the issue of platform security of Internet nodes, etc. These threats contribute significant risks compared to traditional telephone networks, which seem hard to impossible to manage - at least in the consumer market.

Evolving VoIP based on SIP to a solution which comes with security characteristics similar to e.g. today's mobile telephone networks is a significant undertaking, which did not even start yet; the technical differences are significant: signalling, just to mention one example, is done in-band in an open network for VoIP, whereas a separate signalling channel is used in POTS/GSM for decades due to good security reasons.

Technically seen, only an additional security infrastructure based on some secure hardware (like the SIM/USIM in GSM or UMTS) seems a realistic option for achieving a security level that would bring

VoIP close to that of the traditional telephone system. However, other aspects -like availability- would still be missing and seem to be even harder to provide.

Overall, we believe that the various aspects of security in VoIP are little understood today. Since the implications to businesses, end users, and governments (e.g. in terms of lawful interception) can be (and probably will be) significant, further work on the issues we touched in our paper is clearly needed.

## References

- [1] H.323, <http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-H.323>, last checked January 22<sup>nd</sup> 2005
- [2] Rosenberg, Schulzrinne et al., "SIP: session initiation protocol", RFC 3261, 2002
- [3] B. Goode, "Voice over internet protocol", Proc. of the IEEE, Vol. 90, No. 9, September 2002, pp. 1495-1517
- [4] Net2Phone, US VoIP provider, [www.net2phone.com](http://www.net2phone.com), last checked January 22<sup>nd</sup> 2005
- [5] Sipgate, German VoIP provider, [www.sipgate.de](http://www.sipgate.de), last checked January 17<sup>th</sup> 2005
- [6] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, 1999
- [7] Additional RFCs complementing the SIP protocol: RFC 3262, 3263, 3264, 3265
- [8] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327
- [9] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889
- [10] RFCs defining additional request types for SIP: RFC 2976, 3262, 3311, 3428, 3515, 3265
- [11] S. Dusse, P. Hoffman et al., "S/MIME Version 2 Message Specification", RFC 2311
- [12] T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246
- [13] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401
- [14] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409
- [15] D. R. Kuhn, T. J. Walsh, S. Fries, "Security considerations for voice over IP systems", NIST special publication 800-58 Draft, NIST, April 2004
- [16] R. Ackermann, M. Schumacher, U. Roedig, R. Steinmetz, "Vulnerabilities and Security Limitations of current IP Telephony Systems", Proc. of the Conference on Communications and Multimedia Security (CMS 2001), pp. 53-66, <http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/ASRS01-1.html>, last checked January 21<sup>st</sup> 2005
- [17] Cisco Security Advisory: Multiple Vulnerabilities in Cisco IP Telephones, <http://www.cisco.com/warp/public/707/multiple-ip-phone-vulnerabilities-pub.shtml>, last checked January 21<sup>st</sup> 2005
- [18] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, A. Rayhan, "Middlebox communication architecture and framework", RFC 3303
- [19] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489
- [20] J. Peterson, "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323
- [21] J. Rosenberg, C. Jennings, "The Session Initiation Protocol (SIP) and Spam", internet draft, July 2004