SIP Call Security in an Open IP Network

Olli Rantapuska
Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory

May 28, 2003

Abstract

This paper examines the security model and features present in the Session Initiation Protocol (SIP). Security and privacy issues of SIP are important because the protocol is going to be used in many future applications in open IP networks. SIP security builds on established solutions. The currently available security and privacy features of SIP are presented and analyzed. While not perfect, the available features do improve the security of SIP.

1 Introduction

The mobile telephone networks of today make their users reachable regardless of where they are of what they are doing. It is convenient to have a personal device, a mobile phone, that can be used for communicating with other people at will. Smarter devices are also appearing all the time, combining more advanced personal information management features into the same device.

Mobile telephone networks are built on the legacy of the fixed telephone network. However, since the wide adoption of the Internet for data comunication, there have been many talks to harmonize the communication arena by combining telephone calls and data communication into the same network. Placing telephone calls over the Internet requires a common protocol for communication. SIP, the Session Initiation Protocol, has been suggested as the protocol to place telephone-like voice calls over IP networks. The 3GPP organization has chosen SIP to be the call control protocol of future third generation mobile networks.

The Internet, however, is different from the current telephone networks. Especially the open nature of the Internet compared to closed telephone networks can lead to security risks. If people are expected to place telephone calls over open networks, they must be certain that the calls cannot be intercepted or monitored by third parties. It is therefore important to study the security aspects of Internet calls and SIP in particular.

2 The Need for Call Security

The telephone network is a closed network strictly controlled by telephone operators. When a user places a call, the telephone network intelligently chooses the route for the call and opens a connection to the receiving end. A closed network is quite easy to secure; if users can only connect at the exchanges and the trunk network is controlled by the operator, there is only a limited number of possible attacks that the user can devise. The traffic cannot be monitored by outsiders because trusted operators control all components of the network.

On the other hand, IP-based internetworks are generally open networks, meaning that data traffic can in principal be routed through any intermediate networks between the two endpoints. This raises a concern that traffic may be intercepted by a third party as it is routed through the network.

Any attacks performed by a third party are called outside attacks, since they are performed from outside the actual communicating entities. Network protocols can also suffer from inside attacks, i.e. attacks that come from inside the protocol's communicating entities. For example, an entity might offer to act as the server party in a client-server transaction, but in reality the entity could also be an attacker who is only disguised as a server.

The Session Initiation Protocol (SIP) [1] brings telephone call functionality to the IP network. SIP takes care of routing calls and negotiating the data streams that make up the actual communication channel between two endpoints. With these important features available on an open network, care must be taken to avoid any possibility of misuse.

This paper focuses on the security issues around SIP, particularly where SIP is used in an open IP network environment. Both protocol security and privacy features as well as generic security requirements are covered. Inside and outside attacks along with practical security and privacy issues are illustrated and analyzed.

The theory of cryptography, cryptanalysis and any security weaknesses with specific implementations of SIP are outside the scope of this paper. Using SIP for activities not related to voice calls, e.g. instant messaging, is not covered. Billing and charging issues are a complex area in themselves and are only briefly mentioned. Securing the media streams initiated by a SIP session is also a different topic and not discussed in the paper.

3 Security Requirements

The requirements for call security in an essentially open network stem from user expectations for IP telephony as well as experience with open network protocols in general [1, 19]. The requirements are:

- **Identification**. The called party must be able to ensure the identity of the calling party, unless the call is anonymous.
- **Anonymity**. The calling party must be able to place the call anonymously, so that the called party cannot learn the identity of the caller, but accounting will still function

normally.

- **Authorization**. Users must have personal access privileges to the network, so that only authorized users (e.g. users having a subscription) can place calls on the network.
- **Accounting**. Users' calls must be reliably charged for, and there must be no outside or inside attacks against the accounting subsystem.
- **Integrity**. The signaling between calling parties must be verifiably unchanged along the signaling path; only trusted entities are allowed to modify signaling messages.
- **Confidentiality**. Outside users must not be able to intercept and eavesdrop on the signaling channel between the calling parties. Users must be able to trust the signaling network.
- **Privacy**. Users' personal information must not be readable from the signaling messages, and users must be able to control what information is given to calling parties.
- **Spam resistance**. It must not be possible to place forged calls over the system.
- **Resistance to attacks**. There must be no outside attacks that can hinder the use of the system or render it inoperable.

4 SIP Call Model

SIP functionality is built on a network of SIP proxy servers that are used to route call signaling between terminal endpoints. In addition to proxies, the SIP network also requires registrar servers that keep track of users' locations. The proxies and registrars reside on the application layer and are used to route SIP traffic. While user agents can roam around in different networks, the registrars keep track of the user agents' locations at all times.

The generic SIP call formation and teardown procedure is pictured in Fig. 1. The terminal SIP user agents first have to register their current contact addresses with a local SIP registrar. User agent A can then place a call through its local outbound SIP proxy using the INVITE request. The proxy server uses DNS procedures to locate proxy server B [12]. At proxy server B, the request is routed according to the current registration of user agent B.

SIP in its basic form is not a secure protocol. Messages are not encrypted or signed, and the user agents and intermediate proxies can in principle monitor and even modify requests as they are transmitted. Several security measures need to be applied to this procedure to make it at least reasonably secure. These measures are discussed in the next section.

5 SIP Security Features

The SIP protocol integrates many existing security technologies together to form a complex set of security features. There are ways of authenticating user agents, encrypting and signing messages, hiding personal information and negotiating security parameters.

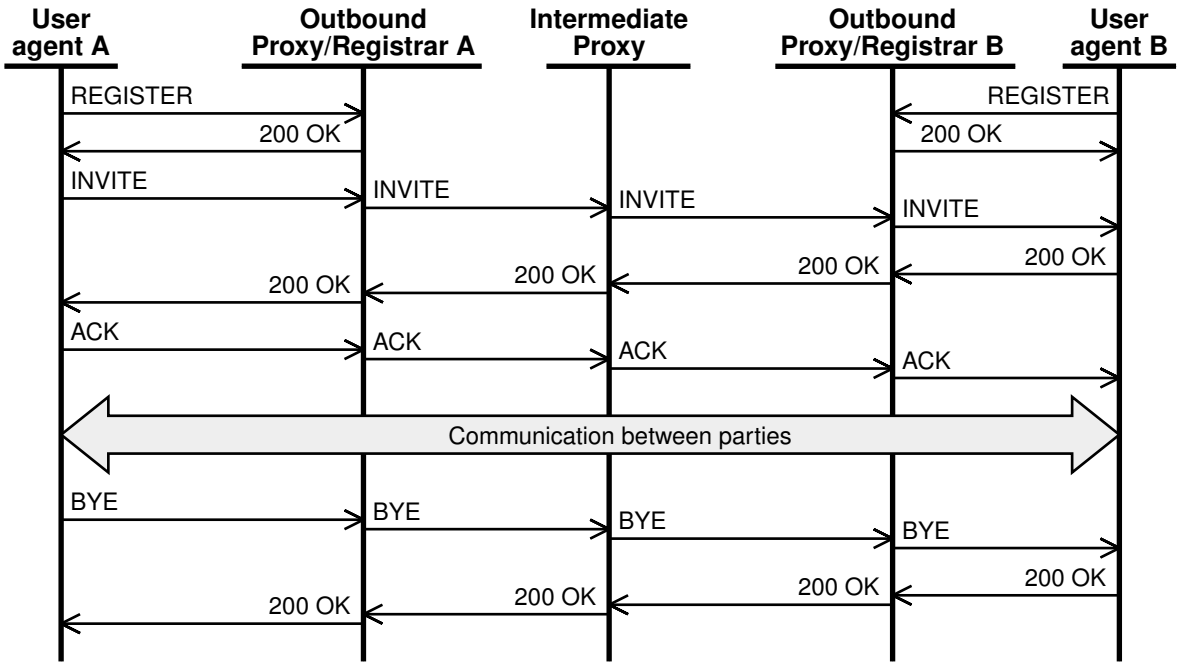

Figure 1: SIP call formation and teardown

SIP imports some security-related behavior from HTTP and SMTP, and relies on other established security solutions, loosely integrating them to form a recommended security framework.

5.1 Identifying the User Agent

In basic SIP communication, request originators can express their identity using the `From:` header field. However, the From field is set by the user agent, and there is no way for the server to verify that the field has been appropriately populated. Originators must therefore be authenticated to verify their real identity. On the other hand, the identities of intermediate proxy servers and the final recipient also have to be verified. Authentication of entities in SIP is mainly provided by a variant of HTTP Digest Authentication [4] and TLS [5].

Whenever a SIP user agent originates a SIP request, the recipient can challenge the originator to prove its identity. In this case, the recipient may be another user agent, a SIP registrar or redirect server. The authentication is performed using a variant of HTTP digest authentication, known as *SIP digest authentication* [1].

The SIP digest authentication scheme works similarly to HTTP digest authentication. In SIP digest authentication, the recipient challenges a SIP request by responding to the request with a `401 Unauthorized` response and including a `WWW-Authenticate:` header in the response. The header field contains an authentication challenge that includes a protection domain (realm) and a nonce value. [1, 10]

The originating user agent receives the response and uses the nonce as a key to perform a hashing function on a shared secret, usually a username and password combination. The

user agent may first need to ask the user for the password. The user agent then re-issues the original request, including the resulting hash string as the authorization in the request. The authorization string is placed in an `Authorization:` header field.

Upon receiving the re-issued request, the recipient should verify that the authenticated user is authorized to perform the requested action. If the authorization failed, the recipient may respond with another challenge.

Similarly to the final recipient of a SIP message, also a SIP proxy server in the intermediate signaling path may require user authentication. For example, if a proxy is providing access to the PSTN from the Internet, the proxy needs to authenticate the user for billing purposes. The proxy server performs user authentication by responding to a SIP request with a `407 Proxy Authorization Required` response, which has a `Proxy-Authenticate:` header containing the challenge. The user agent should then re-issue the request with a proper `Proxy-Authorization:` header field [1, 10].

To verify the identity of the user, the local outbound SIP proxy and registrar server are expected to have a connection to a database of local user accounts. That database would be used to check the validity of the request. The account database should be a centralized separate entity as discussed in [16].

If several proxies in the signaling path require authentication, their authorization values can be differentiated with the realm parameter. Each proxy will have a different realm of authentication. One SIP message can contain several authorization header fields with different realms. [1]

While SIP digest authentication is the current choice for user agent authentication, discussions exist to possibly move to using S/MIME for authenticating users also [18]. S/MIME may provide better security using cryptographic keys instead of passwords.

5.2 Identifying the Server

In addition to authenticating the user agent, also the server needs to be authenticated. The user agent has to be sure that it is providing its authentication information to a trusted server and over a trusted network channel. The recommended way to provide server authentication and a secure channel is to use TLS.

Much like the way today's banking services on the Web are authenticated, the SIP proxy should have a TLS certificate signed by a mutually trusted third-party certification authority. The user agent should have the required public keys of the certification authority preinstalled. This way, the user agent can verify the proxy server's certificate and ascertain that the server is in fact the local outbound SIP proxy server. TLS negotiation also provides a secure encrypted transport layer channel that the user agent can use to authenticate itself to the proxy server using e.g. SIP digest authentication. [1]

Proxy-to-proxy authentication is not natively supported in SIP [10]. However, the proxy network is expected to have TLS certificates, so the proxies can mutually authenticate themselves using the certificates [1]. This ensures that the messages are routed only via trusted proxies and trusted channels. TLS is discussed further in Sec. 5.5.

5.3 Advanced Authentication Features

There are some limitations in the current SIP authentication features. Unless every intermediate proxy server authenticates the originating user agent separately, intermediate proxy servers cannot be certain that the user agent originating a request has been authenticated. Peterson suggests that SIP messages should include a cryptographic token to assert that the originating user's identity has been verified by the originating network [14].

We have seen several ways of authenticating originators of SIP requests. Response messages, however, are in general not authenticated [10]. The response headers can still be partly signed for end-to-end integrity protection. This is also defined by Peterson [15].

There are several ways of authenticating SIP user agents. The local outbound SIP proxy and registrar are the most important parts of authenticating user agents. Once the user agent has first ascertained that the local servers are trusted, it can use a shared secret to authenticate itself to them. Local authentication then allows the user agent to send requests to the outside network.

5.4 Privacy and Anonymity

In addition to authenticating users, it is also important for users to be able to protect their personal information and their real identity. User agents can provide a certain level of anonymity by obscuring the `From:` field of SIP requests. However, not every header field can be obscured, since e.g. `Contact:` is needed for request routing. Comprehensive privacy for SIP requests cannot be provided without support from the SIP proxy network.

Privacy and anonymity can be provided in a SIP proxy network by a privacy service [3]. The privacy service is essentially a proxy server that can also act as a back to back user agent and proxy media streams.

The user agent can provide privacy in cooperation with a privacy service in the proxy network by using anonymous addresses and the `Privacy:` header [3]. The Privacy header field can indicate whether the user requests `header`, `session` or `user` privacy. Header privacy includes obscuring any SIP headers that might reveal the user's identity (e.g. `Via:` and `Contact:`). Session privacy means that the privacy service should anonymize the media session defined in the accompanying SDP body. User level privacy entails constructing informative SIP headers like `From:` so that they do not reveal any information about the user. User level privacy can also be provided by the user agent alone.

If the user agent requests session privacy from the privacy service, the privacy service starts a new SIP session towards the recipient and proxies all traffic between the endpoints. In this way, the endpoints only see the address of the privacy service in their communication. [10]

Providing privacy for SIP responses is also possible. If a user agent places a REGISTER request through a privacy service with an indication of user level privacy, the privacy service can insert an anonymous Contact header field to the message, thereby instructing the registrar to forward all incoming calls through the privacy service. [3]

Many users will want to conceal their identity when using the SIP network, and a privacy

service allows authenticated users to hide any personal information from SIP messages. This way, user agents can place anonymous calls through the SIP network.

5.5 Message Confidentiality

Even if the communicating entities are authenticated, the communication between them still needs to be secured against eavesdropping. Also, proxies in the middle of the signaling path must not be allowed to modify the messages sent. The user agent cannot trust every intermediate proxy, so the protocol must be able to provide end-to-end message integrity between the signaling endpoints.

The SIP protocol, however, requires that message headers like `Via:` be modified at each proxy. Hence, the entire message cannot be enrypted and signed end-to-end, since the proxies would not be able to route the message properly. The set of header fields that can be legitimately modified by proxy servers is `Request-URI, Via, Record-Route, Route, Max-Forwards` and `Proxy-Authorization`. [1]

SIP does offer end-to-end encryption of the message body. Proxies do not need to tamper with the message body, and since the body often contains sensitive information such as encryption keys used to secure the actual media session, it is important to sign and encrypt the message body.

Various mechanisms can be used for encrypting the SIP message body. S/MIME [7] is the recommended format [1, 11]. S/MIME formatted messages can be either signed for integrity protection, encrypted for confidentiality, or both. S/MIME uses a public key encryption mechanism, and latest work suggests requiring the use of AES as the main encryption algorithm [17].

S/MIME certificates can be used to identify a user. An S/MIME certificate in this case asserts that the holder is identified by a certain SIP address. The certificate should be acquired from a known public certificate authority. Other users can then trust that the certificate holder is who he claims to be. S/MIME certificates could be used for authenticating user agents, but currently S/MIME is mostly used in SIP for encrypting and signing message bodies.

Apart from the message body, the SIP message headers can also contain confidential information. As described before, the headers cannot be encrypted end-to-end, but proxy servers can employ hop-by-hop security. If the proxies utilize a secure transport layer (TLS) or a secure network layer (IPSec) between each other, outside users will not be able to eavesdrop on the communication. Hop-by-hop security is implemented on lower protocol layers, and is not a feature of SIP in itself.

IPSec [6] generally does not require integration with SIP applications. If user agents and the local outbound proxy server have a previous relationship and pre-shared keys, IPSec would be a good candidate for secure communication. Situations where no pre-existing trust association exists are better served using TLS [5]. TLS must be tightly coupled with the SIP application implementation. [1]

5.6 Protecting Message Headers

It is possible to sign part of the SIP message headers end-to-end for integrity protection. This prevents against hostile proxy servers on the signaling path from modifying the request. The message headers are signed by encapsulating the entire SIP message in a MIME body of type `message/sip` and tunneling them inside the regular SIP messages. The MIME body can be secured as any SIP message body. Upon receiving the message, the recipient can examine the differences between the inner and outer message headers, and hence determine the integrity of the headers. Differences in headers that need to be modified by proxies would not be taken as a breach of integrity.

A tunneled inner message can also include headers, for example a `Subject:` header, not present in the outer message for the purpose of confidentiality [1, 10]. Also, it is not necessary to include all headers in the tunneled inner message. Since the proxy network needs to modify some headers in the SIP request anyway, the originator may only choose to tunnel a fragment of the SIP message headers. This is discussed in [15, 8].

5.7 Security Agreement

With the selection of different security mechanisms now available, a security agreement procedure has been established for negotiating among available SIP security features [2]. According to this agreement, a user agent can use the OPTIONS request to pick and choose among available secure channels before actually initiating communication.

Using the agreement procedure, the user agent first sends an OPTIONS request to the server, containing a `Security-Client:` header that contains a list of security protocols that the user agent supports. The server responds to the user agent with its supported list in the `Security-Server:` header. The user agent then chooses the preferred protocol that both entities support and establishes the secure connection. The user agent still has to send the list of security protocols that the server supports back to the server. This additional verification is used to prevent downgrade attacks. The list is sent with the next outgoing SIP request in a `Security-Verify:` header.

It should be noted that the security agreement procedure illustrated above is only defined for the first hop of the signaling path, i.e. from the user agent to the first local outbound proxy server. The security protocols currently supported in the agreement procedure are TLS, SIP digest, and IPsec with both IKE (Internet Key Exchange) and manual keying.

Generally when placing SIP calls the `sips:` URI is used to signify use of TLS. If a SIPS URI is used as the Request-URI or the `To:` field of a SIP request, every hop prior to the destination domain must be secured with TLS. This way the user agent can request that every proxy employ secure TLS at each hop in the signaling path. [1]

The SIP specification requires that every proxy, registrar and redirect server implement TLS, and that they must support both mutual and one-way authentication. These servers should also possess a site certificate asserting their identity and corresponding to their canonical hostname. Every SIP element supporting TLS must also support the SIPS URI scheme [1]. Independent of proxies, the user agents can use encryption and integrity protection for SIP message bodies and s subset of header fields as well. These security ele-

ments together prevent SIP signaling traffic from being eavesdropped or modified en route.

5.8 Resistance to Attacks

Firewalls and Network Address Translators (NATs) are often used for securing internetworks. However, there are many problems with using SIP in networks with firewalls and NATs [10].

SIP transmits IP addresses and port numbers inside messages, which causes a problem with NATs since they do not know how to modify the SIP messages appropriately. The result is that SIP messages that pass through NATs cannot be responded to, since the responses will be misrouted. Also media connections negotiated in SDP messages will be misrouted if NAT devices are present. SIP messages also leak information about the internal network topology behind the NAT device, which is a security risk. [10]

Also many firewalls only permit outbound traffic and responses to outbound TCP connections. If a SIP message is sent using UDP, its response will not be allowed through the firewall. SIP over TCP will work, but any RTP media streams will function in only one direction. Also, no SIP messages can originate from the outside network. Of course, firewalls can be configured to allow all SIP messages, but this is often too big a security risk for administrators to take.

The best solution to firewall and NAT problems is a genuine SIP application layer gateway (ALG). The ALG proxies SIP and RTP traffic and is trusted by the firewall. As a user agent places a SIP call, it is routed through the ALG that modifies the SIP message to point to the ALG for RTP packets. The ALG can also make sure that needed security policies and authentication rules are respected [10].

Denial of service attacks are an important risk with SIP. Especially forking proxies can amplify the traffic of the originator. If the traffic is all destined to a single point, that point can be easily congested. Therefore, all requests should be authenticated and measures should be taken to prevent flooding. Authentication challenge responses (401 and 407) should be transmitted only once, forgoing the normal retransmission algorithm. [1]

A secure SIP network should be protected from the outside by firewalls, and connections from outside should only be permitted to SIP proxies on the edge of the proxy network. The edge proxies authenticate users and relay messages between the ouside network and the proxy network. Core proxies inside the proxy network only perform message routing and need not take care of authentication. Headers like `Via:` can also be stripped by the edge proxies to hide the network's internal topology. For response routing, a `branch` parameter can be used to identify the missing via headers. [10]

The edge proxies also help to protect the proxy network from denial of service attacks, flooding and viruses. The proxy can be made to scan for known viruses in messages and to block frequent transmissions. Hostile traffic is always easier to recognize if traffic is routed through a single monitoring point.

6 Analysis of SIP Security

SIP has many security features that can be applied on top of the protocol to improve security and prevent attacks. The key question is whether these security features are sufficient for providing the required level of security and privacy. Relevant security requirements were presented in Sec. 3. This section analyzes how those requirements are fulfilled by the security features that SIP currently provides.

6.1 User Identity and Anonymity

As a user agent receives a SIP request from another user agent, the recipient should be able to reliably verify the identity of the originator. If the two user agents have a previous relationship and have exchanged public keys, they can use S/MIME signatures to sign the message body and possibly part of the message headers. A global S/MIME public key infrastructure is needed for wide scale deployment, but no such infrastructure currently exists [1]. Public keys would hence be self-signed, which makes the initial key exchange vulnerable to a man-in-the-middle attack.

If the two user agents have no previous relationship, the recipient currently has to trust the proxy network to have authenticated the request originator. This can be done explicitly using an AIB [14], or implicitly by using TLS and trusting the proxy network. In the latter case, the proxies should only accept connections from trusted proxies. It is important to note that if the proxy network cannot be trusted, the user agents must employ S/MIME signatures for authenticating each other, integrity protection and possibly encryption also.

If the request originator wishes to remain anonymous, he should use the privacy procedures in [3]. The simplest way is to obscure the relevant SIP headers in the user agent. However, even if the user agent hid the user's identity by obscuring the necessary headers, proxy servers further away on the signaling path may add headers that reveal the company that the user is working for. For best privacy for SIP message headers, the user agent should also request header privacy from a network privacy service. The privacy service can conceal all message headers by adding its own address in the place of the user agent's contact information.

Even if headers are obscured, the actual media session may still reveal the network address of the originating user agent. To hide the network address from the recipient, a privacy service will be used as a back to back user agent to proxy the entire session on behalf of the originator. This is a reliable way of hiding the identity of the originator from the recipient, but the real originator can still be identified and authenticated by the privacy service. The originator therefore has to trust the privacy service with all traffic. The messages also cannot be encrypted if they are to be proxied by a privacy service, which is a serious problem if confidentiality is required. There is a tradeoff between privacy and confidentiality.

Whenever a privacy service is used to conceal personal information, the user agent should have a direct, secure connection to the privacy service. Otherwise, the SIP messages can be modified before they reach the privacy service and possibly routed around the service. The privacy service can use any connection method towards the final recipient, but should of course use a secure signaling path.

6.2 User Authorization and Accounting

The proxy network has to authenticate users to ensure that they are authorized to send requests. Users' subscription information can be stored in the user agent in some secure manner. The SIP digest authentication scheme used for challenging the user to authenticate himself is adequate, although digest authentication is natively vulnerable to man-in-the-middle attacks [4]. SIP digest authentication should therefore always be run over a secure connection with the server already authenticated by some other security association [2]. In the future, S/MIME authentication based on cryptographic certificates would probably be more secure than SIP digest [18].

Authenticating users allows the network operator to apply logging and charging capabilities to the SIP network. Accounting generally requires the users to be reliably authenticated before they can originate any SIP requests. The SIP proxy network must therefore challenge all unauthorized requests and bind charging mechanisms to the local outgoing proxy servers that relay the requests. There should be a separate interface towards accounting systems [16].

One other aspect of authenticating users is how to prevent spamming, i.e. prevent users from sending unsolicited messages to other users anonymously. If a SIP proxy accepts requests from unauthenticated users, or from other proxies that accept requests from unauthenticated users, the users can use the proxy to inject arbitrary requests to the SIP network. This scenario is similar to the SMTP "open relays" present in the e-mail network on the Internet. Trust is therefore important in a SIP proxy network.

There are also performance issues with security features. The SIP authentication procedures have been examined in practice, and they in fact cause a big hit in server performance. It has been shown that incorporating SIP security accounts for almost 80 percent of the processing cost of a stateless SIP server and 45 percent of a call stateful server [9]. The main processing cost comes from the extra messaging required for authentication. Encrypting parts of SIP messages using S/MIME also increases the size of the messages, which may be undesirable in many cases.

User authentication is crucial if a network operator wants to charge for sending SIP messages. If unauthenticated users can send requests using the proxies, they can send unsolicited requests and place crank calls without being traced. Authentication is also critical for registrations. Even if the user wants to place anonymous calls, he still has to authenticate himself to the network.

6.3 Confidentiality

When a user agent places a call using SIP, the user agent can choose whether the call should be confidential or not. A confidential call essentially means that nobody can eavesdrop on the call signaling or other contents. In an open network, confidentiality is preferred. The SIPS URI allows the originator to require intermediate proxies to employ TLS encryption at every hop up to the recipient's domain. There is, however, a risk that the message is intercepted inside the recipient's domain if the last hop is not encrypted. [1]

The main problem with TLS is that it does not run over UDP. Keeping up several TCP

connections becomes arduous for proxy servers. Especially since local outbound proxies need to support an open TLS connection to every user, scalability problems can arise. [1]

Since SIP messages are transmitted end to end, the receiving user agent gets to see all the message headers that have been used when delivering the message. In particular, the SIP message may leak undesirably broad information about intermediate networks. The originating user agent can request privacy by using special headers [3]. In any case, network edge proxies may have to employ special handling of e.g. `Via:` headers to prevent information leaks [10]. These procedures naturally add to the complexity of the protocol, which is not preferred.

In an open network, secure connections should be the norm. Proxy servers should use TLS for authentication and encryption. User agents can negotiate a secure connection with the nearest proxy using the security agreement procedure of [2]. While SIP digest authentication is workable as an authentication tool, SIP digest should not be preferred as the only security mechanism because of its limitations.

6.4 Outside Attacks

Providing SIP application layer gateways at the edge of the network helps to conquer the problem with firewalls. The ALG is responsible of proxying all traffic between the two networks, and it should also be able to modify SIP message headers to hide the topology of the private network.

It should be noted, however, that the ALG cannot process encrypted message bodies. If an encrypted message body contains the private network address of the originator that is going to receive media streams, those streams will be blocked by the firewall since they would not go through the ALG. Signed message bodies will also be invalidated by an intermediate ALG.

The impact of denial of service attacks can be mitigated by providing a single entry point for outside authentication requests. However, having every connecting proxy server authenticate itself using TLS requires a fair amount of processing. Denial of service attacks cannot be completely avoided in a SIP network since any SIP network needs a connection to outside networks for better coverage.

Of course, a central item in the security of SIP is the robustness of individual SIP implementations. Several vulnerabilities, e.g. buffer overflows with the processing of SIP message headers, have recently been found in various implementations [13]. Direct SIP connections to user agents should be blocked, forcing all connections to traverse a local proxy server.

7 Conclusions

SIP is not easily secured. The protocol reuses security measures present in the HTTP protocol, and those measures do not address all the security concerns with SIP. Therefore, additional security features such as S/MIME and TLS are also applied to the framework.

Using many different kinds of security mechanisms poses difficulties with integration. It is also hard to show that all relevant security risks are taken into account. In SIP, user agents are generally forced to trust the proxy network, since encrypting message bodies end-to-end causes problems with privacy services and application layer gateways. The proxy network should be built on mutual trust and secure intermediate connections.

In the end, robust SIP implementations that include support for SIP security features will be the proof that the protocol is ready for adoption on a wide scale.

References

[1] Rosenberg et al. *SIP Session Initiation Protocol*. RFC 3261, IETF Network Working Group, June 2002. Available at http://www.cs.columbia.edu/sip/drafts/rfc3261.pdf.

[2] Arkko et al. *Security Mechanism Agreement for the Session Initiation Protocol (SIP)*. RFC 3329, IETF Network Working Group, January 2003. Available at ftp://ftp.rfc-editor.org/in-notes/rfc3329.txt.

[3] Peterson. *A Privacy Mechanism for the Session Initiation Protocol (SIP)*. RFC 3323, IETF Network Working Group, November 2002. Available at ftp://ftp.rfc-editor.org/in-notes/rfc3323.txt.

[4] Franks et al. *HTTP Authentication: Basic and Digest Access Authentication*. RFC 2617, IETF Network Working Group, June 1999. Available at ftp://ftp.rfc-editor.org/in-notes/rfc2617.txt.

[5] Dierks and Allen. *The TLS Protocol Version 1.0*. RFC 2246, IETF Network Working Group, January 1999. Available at ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt.

[6] Kent and Atkinson. *Security Architecture for the Internet Protocol*. RFC 2401, IETF Network Working Group, November 1998. Available at ftp://ftp.rfc-editor.org/in-notes/rfc2401.txt.

[7] Ramsdell, Editor. *S/MIME Version 3 Message Specification*. RFC 2633, IETF Network Working Group, June 1999. Available at ftp://ftp.rfc-editor.org/in-notes/rfc2633.txt.

[8] Sparks. *Internet Media Type message/sipfrag*. RFC 3420, IETF Network Working Group, November 2002. Available at ftp://ftp.rfc-editor.org/in-notes/rfc3420.txt.

[9] Salsano et al. *SIP security issues: The SIP authentication procedure and its processing load*. IEEE Network, November 2002

[10] Sinnreich and Johnston. *Internet Communication Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*. Wiley, 2001.

[11] Camarillo. *SIP Demystified*. McGraw-Hill Professional Book Group, 2001.

[12] Rosenberg et al. *Session Initiation Protocol (SIP): Locating SIP Servers*. RFC 3263, IETF Network Working Group, June 2002. Available at ftp://ftp.rfc-editor.org/in-notes/rfc3263.txt

[13] CERT. *Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)*. CERT Advisory CA-2003-06, February 2003. Available at http://www.cert.org/advisories/CA-2003-06.html.

[14] Peterson. *Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)*. Internet-Draft, IETF SIP Working Group, February 2003. Available at http://www.ietf.org/internet-drafts/draft-ietf-sip-identity-01.txt Work in Progress.

[15] Peterson. *SIP Authenticated Identity Body (AIB) Format*. Internet-Draft, IETF SIP Working Group, February 2003. Available at http://www.ietf.org/internet-drafts/draft-ietf-sip-authid-body-01.txt. Work in Progress.

[16] Loughney and Camarillo. *Authentication, Authorization and Accounting Requirements for the Session Initiation Protocol*. Internet-Draft, IETF SIP Working Group, February 2003. Available at http://www.ietf.org/internet-drafts/draft-ietf-sipping-aaa-req-02.pdf Work in Progress.

[17] Peterson. *S/MIME AES Requirement for SIP*. Internet-Draft, IETF SIP Working Group, October 2002. Available at http://www.ietf.org/internet-drafts/draft-peterson-sip-smime-aes-00.txt Work in Progress.

[18] Mahy. *Discussion of suitability: S/MIME instead of Digest Authentication in the Session Initiation Protocol (SIP)*. Internet-Draft, IETF SIPPING Working Group, October 2002. Available at http://www.ietf.org/internet-drafts/draft-mahy-sipping-smime-vs-digest-00.txt Work in Progress.

[19] Jacobs et al. *MGCP, MEGACO, and SIP VoIP Signaling Protocol Security Requirements*. Internet-Draft, October 2002. Available at http://www.ietf.org/internet-drafts/draft-jacobs-signaling-security-requirements-00.txt Work in Progress.

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP can also invite participants to already existing sessions, such as multicast conferences.Â To that end, SIP provides a suite of security services, which include denial-of-service prevention, authentication (both user to user and proxy to user), integrity protection, and encryption and privacy services. SIP works with both IPv4 and IPv6. 3 Terminology.Â The ones present in an INVITE include a unique identifier for the call, the destination address, Alice's address, and information about the type of session that Alice wishes to establish with Bob. The INVITE (message F1 in Figure 1) might look like this: Rosenberg, et. al. Although secure TCP sessions can require more call set-up processing, the frequency of on-going SPA SIP registration messages can be reduced. This reduction in signaling messages can reduce the load on service provider networks. What is IP Telephony Security? IP telephony security requires securing both the control channel and the voice path. Session security is based on authenticating the parties in the call and then encrypting the communication between the parties. Standards-based security methods for authentication and encryption ensure that the communication is not compromised. A SIP proxy/registrar is an essential part of a VoIP network. Today I will focus on all Open Source available solutions for deploying SIP proxies. Some proxies are useful for beating NAT by rewriting IP addresses in SIP messages, some proxies are useful as security tools and some of them act as registrar proxies which are the most important part of a VoIP network. There are many types of SIP proxies.Â Kamailio: Kamailio is an Open Source SIP Server released under GPL, able to handle thousands of call setups per second.Â It handles registrations of SIP clients on a private IP network and performs rewriting of the SIP message bodies to make SIP connections work via a masquerading firewall (NAT).