Using PGP Signatures for Securing SIP Infrastructures

Sebastian Hübner, Nicolas Rüger, Bettina Schnor
Institute of Computer Science
University of Potsdam
Potsdam, Germany
{huebners, rueger, schnor}@cs.uni-potsdam.de

***Abstract*—Because of increasing bandwidth and decreasing costs for the provider, Voice-over-IP is an alternative to the Public Switched Telephone Network for many users. But with the propagation of Voice-over-IP new harassments and threats occur. Assuring the identity of communication partners is significant in this context. Without the authentication of communication partners, the infrastructure is vulnerable to attacks like URI-spoofing, call and registration hijacking. Authenticity is necessary for detecting and avoiding Spam over Internet Telephony (SPIT). Only if the identity of a caller can be verified, a source of SPIT can be exposed and appropriate countermeasures can be taken. In this paper, we present a decentralized approach for authentication in the Session Initiation Protocol (SIP) using PGP signatures. Due to already existing data structures this mechanism can be easily integrated in the SIP without the need of new SIP extensions. Measurements show that our approach results into tolerable overhead.**

***Keywords*-Voice-over-IP (VoIP); Authentication; Pretty Good Privacy (PGP); Session Initiation Protocol (SIP); Signature**

I. Introduction

The Session Initiation Protocol (SIP) [14] is one of the most commonly used protocols in Voice-over-IP communications. SIP handles the signaling, which includes establishment, modification and termination of a media session between two or more communication endpoints. During the signaling the negotiation of call properties is done Further, necessary data for a call, e. g. identities of the communication partners is exchanged. In combination with SIP the Real-time Transport Protocol (RTP) [15] is usually utilized for transferring the media data. Figure 1 shows a common network topology known as the SIP Trapezoid. It consists of the following components: registrar, proxy, User Agent Client (UAC) and User Agent Server (UAS). The UAC sends a request to an UAS. A UAS receives a request and answers with one or more appropriate responses. In SIP, different network entities may be in the role of a UAC or UAS. For example, during the call invitation the caller acts as UAC and the callee as UAS.

Our goal is a secure Voice-over-IP (VoIP) architecture, which guarantees a quality of service comparable to the Public Switched Telephone Network (PSTN) service. It offers the possibility to integrate Pretty Good Privacy (PGP) signatures in the SIP context similar to their more common use in e-mails.

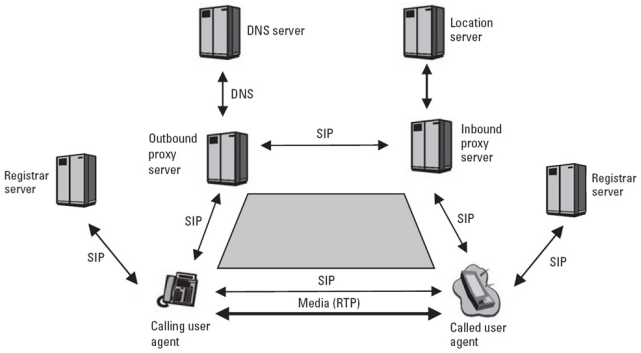

Figure 1. SIP Infrastructure [8]

This paper is structured as follows: In Section II, we define the requirements for a secure SIP infrastructure. In Section III, we discuss related work. In Section IV, we present our approach and its integration into SIP. A security analysis of our concept is given in Section V. In Section VI, we describe the implementation of our protoype and present measurement results in Section VII.

II. Security Requirements in SIP

This section describes security requirements we consider important in a SIP infrastructure:

A. End-to-End Authentication

The assurance about the identities of the involved communication partners is mainly in the interests of the endpoints. They exchange potentially private or personal data during their communication and want to be sure about the recipient's identity. Thus, the decision whether certain information is given or not depends on the authenticity of the communication partners. An authentication mechanism has to realize a direct end-to-end authentication between the endpoints.

B. Mutual Authentication

Caller and callee have to authenticate themselves against each other. Both endpoints of the communication want to

 ISBN: 978-1-61208-183-0

be sure about the other's identity. In SIP, messages are exchanged among different network entities, not only between the endpoints. For example endpoints also communicate with registrar or proxy servers. The SIP standard describes various threats and attacks caused by missing authentication of server components, e.g., Call Hijacking, Registration Hijacking or Impersonation [14] . Their attempt is to make endpoints unreachable to others (DoS). These attacks do not directly affect the authenticity of the communication partners but they have impact on the call's quality. To ensure the quality of calls it is necessary to apply mutual authentication between all UAs in a SIP network. Consider that in SIP different entities are able to act as a UA, not only the endpoints. Any logical entity that creates and sends a request is a UAC, any logical entity that creates and sends responses to a request is a UAS. Thus, requests and responses should be exchanged between mutual authenticated network components only.

C. Authentication During Signaling

Authentication has to be realized during signaling. Once a media stream is established confidential information can be transmitted. Therefore, before a call is accepted by the callee or before the phone even rings, the authenticity of the caller has to be verified. Moreover, it is important to secure all relevant signaling messages during a SIP session. Especially, the termination of a call is a crucial point. In SIP, there are several signaling messages, i. e. *BYE* or *CANCEL* requests to terminate a session. Only authenticated participants should be able to send these requests to avoid an unwanted call termination. To prevent Man-In-The-Middle or reply attacks, the signaling messages have to be protected by a signature field.

D. Technical Requirements

The use of security mechanisms in a given SIP infrastructure has to be practical. The routing of messages may not be hindered or impeded. Additionally, the functionalities of the different SIP components may not be affected.

III. Related Work

There are different approaches that relate to requirements presented in the previous section.

A. SIP Digest Authentication

SIP Digest [14] is based on a challenge/response principle. For its appliance a shared secret between UAS and UAC is necessary. Usually, this is the case between UAs and the registrar or proxy server. The User Agent (UA) authenticates itself against the server by using its associated credentials (user name, password).

In reality, there is no such relationship between two endpoints. The called party cannot hold personal data for any possible caller. But, this would be necessary to verify the origin of an incoming call. Moreover, SIP Digest authentication only allows the authentication of the caller. The initiator of a conversation is not able to authenticate the callee. Therefore, this method is not suitable for mutual authentication. Guillet et al. [5] extend SIP Digest authentication by mutual authentication, but still a shared secret is needed.

Strand and Leister [17] point out some weaknesses and drawbacks of SIP Digest Authentication. It is not suitable for end-to-end or cross-domain authentication. Moreover it is vulnerable to different attacks. The authors focus on a register attack, which is caused by modifying the *Contact* header of SIP message during the registration phase. They suggest extending SIP Digest Authentication by including the contact header value in the digest computation to counter that specific register attack.

B. TLS

RFC 3261 [14] defines the utilization of Transport Layer Security Protocol (TLS) [3] within a SIP network. By using client-side and servers-side certificates a mutual authentication can be achieved. However, TLS realizes a hop-by-hop security. Only the connection to the next node is secured and authenticated. To realize a secure connection between the endpoints via TLS a chain of trust has to be established between all hops on the path from the caller to the callee. The endpoints trust in each other's identities because of an existing trusted relation between them. But there is no direct end-to-end authentication.

SIP provides the SIPS URI Scheme to initiate a hop-by-hop TLS connection. But the last hop between the inbound proxy and the callee is not necessarily included in this trust chain. According to RFC 3261 the security mechanisms on that last hop depends on the policy of the domain.

In spite of this, there exist different approaches to secure SIP infrastructures on the base of TLS. Jiang [7] uses a hop-by-hop TLS connection to exchange a session-key to encrypt the following media streams and a so-called setup-key. The setup-key is valid only for the next call and used for a direct end-to-end authentication. But the concept is based on the trust in the hop-by-hop TLS connection.

In [10], Kong et al. present a solution for securing the localization of communication partners. This is achieved by providing integrity for the contact header of a SIP message by using signatures. For that purpose each endpoint generates a public and a private key. During call initiation the caller creates a signature for the contact header using his private key. Now, the callee is able to verify the identity of the caller and sends a signed *200 OK* response, if authentication was successful. After receiving the response the caller verifies the callee's identity as well. The endpoints exchange their public keys using a hop-by-hop TLS connection outbound and inbound proxy. Again, the last hop is not considered. While the authors focus on the localization of

 ISBN: 978-1-61208-183-0

communication partners, the integrity of other SIP messages and header fields is not verified.

C. S/MIME

S/MIME [13] allows end-to-end encryption. Entire SIP messages are encapsulated within a MIME body. They are signed with the sender's private key and encrypted with the public key of the intended recipient. To allow the routing of encrypted messages their header is duplicated. So, the recipient has to deal with "inner" and "outer" message headers (SIP Tunneling). The "outer" header is used to verify the authenticity of the encapsulated information [14]. But, there are parts of the header, e.g., the via header field, which is legitimately modified during routing. Thus, end-to-end authentication can only be realized for unchangeable parts of the header.

D. PGP

RFC 2543 [6], the previous SIP standard, describes the usage of PGP-based encryption to provide authenticity of SIP messages. RFC 2543 introduces the basic structures and headers for the appliance of PGP in the SIP context. A complete description of security aspects and mechanisms, which are realized by PGP, is not given. This may be a reason why the usage of PGP is described as "incompletely specified". The current RFC 3261 deprecates PGP in favor of S/MIME.

IV. Our Approach: PGP Signatures

Next, we present our approach using PGP signatures in SIP. Our approach fulfills the requirements from Section II. It can be used within SIP infrastructures conform to RFC 3261.

A. Motivation

Although PGP is deprecated in the current RFC 3261, we favor it for the following reasons:
In TLS and S/MIME hierarchical PKIs depending on X.509 certificates are used. Among others, Ellison and Schneier discuss the risks of this approach [4]. They argue that vague Certificate Authority practices to issue certificates cause an imprecise meaning of the word "trust". Furthermore, current events show that a valid certificate does not necessarily mean the owner is trustworthy [1].

Unlike this hierarchical approach PGP, utilizes a "Web of Trust" in which trust is considered private information (cf. IV-C). In [19], Ulrich et al. point out that this trust concept helps to prevent the propagation of faked certificates.

Since PGP is in widespread use for encrypting and signing e-mails, we propose to use the already existing PGP-Keys and trust relationships to secure VoIP communications as well.

B. Concept

In contrast to SIP digest, we do not use message authentication codes, but signatures. Every entity of the SIP infrastructure needs a pair of PGP keys for signing and verifying messages.

After receiving a request the UAS sends an appropriate response to challenge the identity of the UAC. The UAC repeats the initial request and appends a signature for the message body, a subset of the header fields and certain elements from received challenge. The UAS verifies the signature and sends a signed response. In result, the UAC can verify the server's identity. Figures 2 and 3 show the computation of signatures for requests and responses.

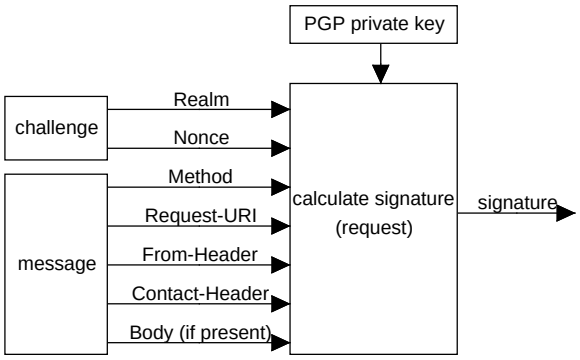

Figure 2. PGP Signature - Request

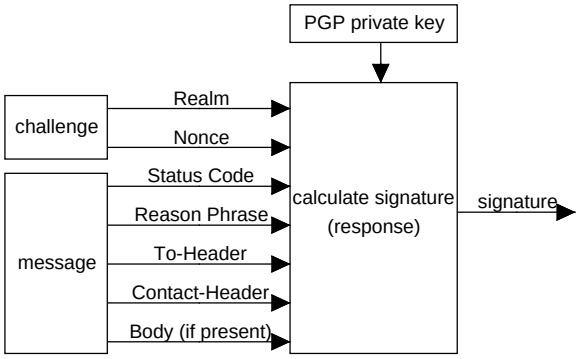

Figure 3. PGP Signature - Response

The verification of a signature is the same process for both UAC and UAS. The signature of a message is calculated by using the sender's private key. So the recipient needs the corresponding public key. Before checking the signature it is crucial to verify the key's associated identity (see Section IV-C). If the key's identity could be verified, the recipient checks whether the signature of the message is correct or

 ISBN: 978-1-61208-183-0

not. Calls should only be established if the *INVITE* request and the referring *200 OK* response are correctly signed and the identity of the corresponding keys is verifiable by the recipient.

After the establishment of a call all SIP messages, which affect the state of the session, have to be signed (see Figure 4).

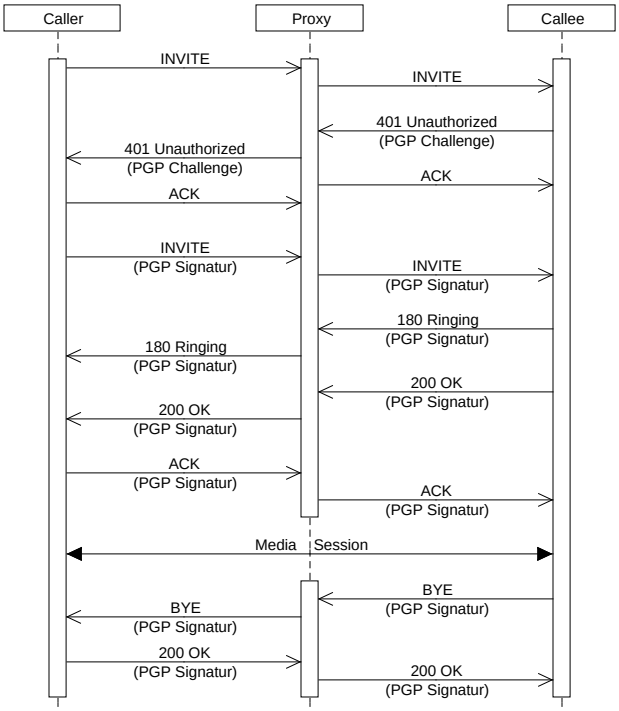

Figure 4. PGP Authentication - Message Sequence

Since a UAC can be challenged by different components of a SIP infrastructure, for example proxy or callee, a request may contain more than one signature. This procedure has to be applied between all components of a SIP infrastructure, which act as UAC and UAS to fulfill the requirements from Section II. In the following, the focus is primarily on the endpoints.

C. Evaluation of the concept

The main feature of this concept is a direct end-to-end authentication. Signed messages are generated and verified by the endpoints. The authenticity of the participants does not depend on the intermediary network entities (cf. TLS). So the last decision on the call establishment is up to the endpoints. This also means that keys with every communication partner have to be exchanged.

The functionality of PGP signatures is based on the binding of the keys and their associated identities. The OpenPGP standard [2] defines two concepts for establishing trust: By signing another key a user claims to be sure of the key-entity binding ("Public-Key Trustworthiness" [19]). By adding a certain trust level it can be determined how much another user is trusted to sign other keys carefully ("Introducer Trustworthiness" [19]). Unlike signatures, the trust level can only be set manually in the local keyring, it is not exported. A key is valid if it is signed directly by the recipient or the validity can be derived from a transitive trust chain ("Web of Trust"). For that the following conditions have to be met: Each key has to be signed by the preceding node and for each key the trust value must be set. Therefore, this chain can only be generated within the local keyring of a user.

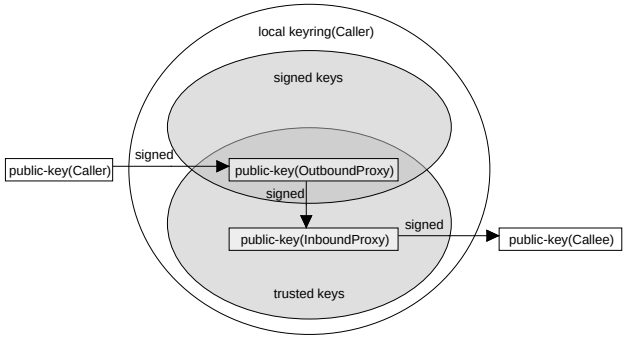

Figure 5. PGP Trustchain in SIP Infrastructure

Figure 5 shows a chain of trusted keys in a SIP infrastructure (see Figure 1) from the caller's view. To verify the callee's key the public keys of the proxies have to be trusted. For establishing a trust chain the key of the Outbound Proxy has to be signed by the caller and the key of the Inbound Proxy has to be signed by the Outbound Proxy. Note that it is also possible to find another path to the callee's key, for example with keys from existing social relationships of the caller. The "Web of Trust" is practical especially for closed groups with signed keys or those users that frequently sign other keys and get signed by them [19].

In case the key's identity cannot be assured by the described concept the recipient will not be able to verify the sender's identity. Consider that it is still possible to check the message's signature with an unknown key. But even though the signature of a SIP message is correct the corresponding key may be falsified. So the endpoint has to decide whether the call should be established or rejected without any further knowledge of the sender's identity. It is also important to note that a signature only assures the integrity and authenticity of elements, which are included in its calculation. So for our concept it is crucial to choose the parts of a message, which are necessary to verify the sender's identity.

 ISBN: 978-1-61208-183-0

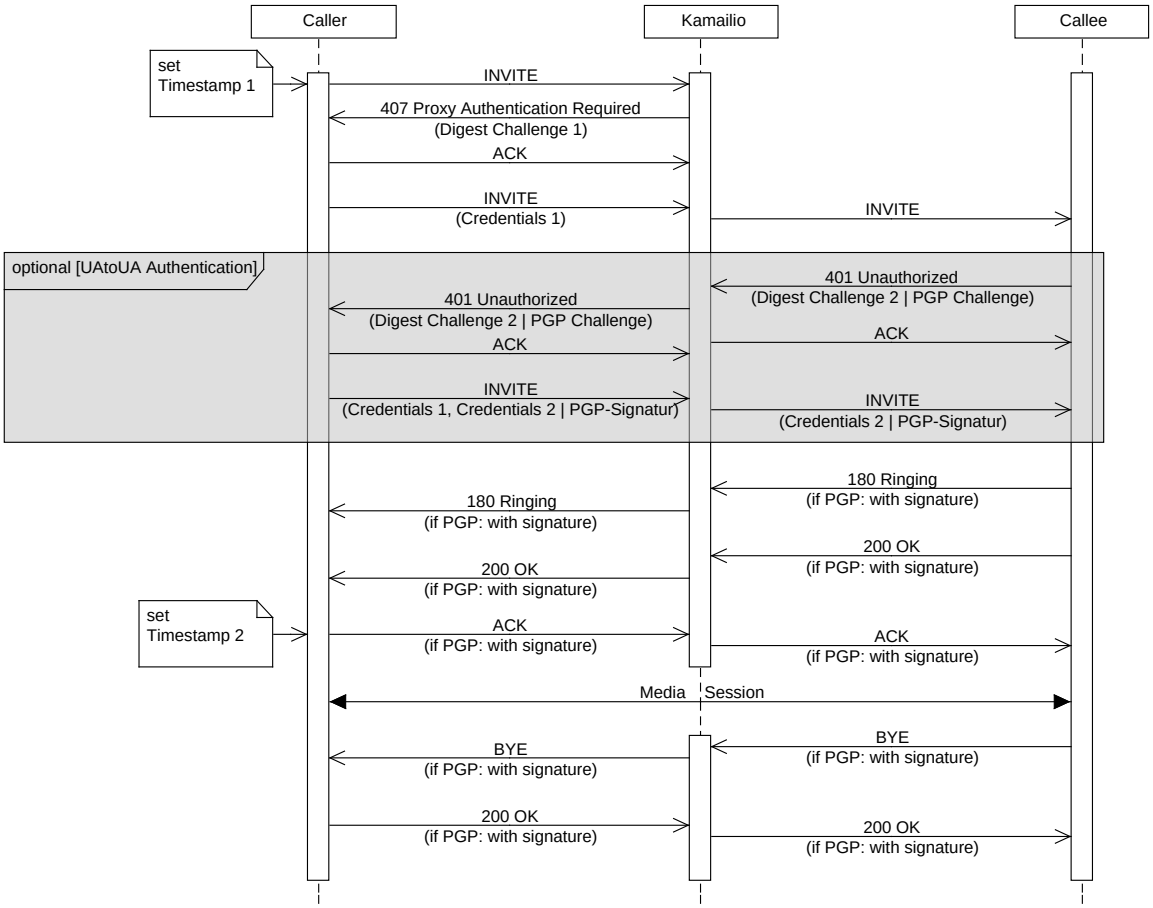

Figure 6. Measurement Scenarios

D. Integration in SIP

Appropriate SIP messages and header fields are necessary for the transmission of the authentication challenge and the corresponding signatures. SIP already defines the responses *401 Unauthorized* (sent by UAS) and *407 Proxy Authentication Required* (sent by proxy) to transport the authentication challenge for SIP Digest. Therefore, the messages contain a *WWW-Authenticate* header and a *Proxy-Authenticate* header. These messages and header fields can also be used to transport a PGP Authentication challenge. After receiving the challenge the UAC has to extend the initial request by a signature and has to send it again. RFC 3261 defines the request-header fields *Authorization* and *Proxy-Authorization* to transport the response of a received digest challenge. Again, these elements can also be used to transport a PGP signature. Hence, no special SIP extension is necessary. Moreover, the PGP Authentication does not affect the existing authentication mechanism in SIP. A message header, which already contains the information for a SIP Digest Authentication (e.g., between endpoint and proxy) can also carry the PGP Authentication (e.g., between the endpoints).

V. Security Analysis

Signatures are applied in communication infrastructures to provide authenticity, data integrity, and non-repudiation. With the awareness of the PGP trust concept and its weaknesses (cf. IV-C), our concept provides countermeasures against the following threats and harassments:

URI-Spoofing: By the lack of authenticity of signaling messages, it is possible to falsify the identity of communication partners to obtain sensitive information or to use personalized services. In our concept this is avoided by signing the proper header fields (*To* header, *From* header) and using the key-entity binding in PGP.

Call Hijacking: Without authentic localization information a call can be redirected toward an attacker's device. For example, the attacker can act as Man-In-The-Middle. As a countermeasure, our concept provides integrity for the identities and the localization information as well.

 ISBN: 978-1-61208-183-0

Registration Hijacking: Similar to Call Hijacking, an unauthenticated registration information allows redirection of calls as well. Moreover many Denial of Service attacks are caused by unauthenticated or unauthorized REGISTER requests [14]. Hence, the affected endpoints are not available anymore. To counter this attack our concept has to be applied between all components of a SIP infrastructure which act as UA (cf. IV-B).

Impersonation: Without a proper authentication an attacker can impersonate every component of a network. Similar to URI-Spoofing and Registration Hijacking this is avoided by providing authenticity of the communication partners and applying our concept between the different SIP entities.

Terminating Sessions: Within established sessions or during their establishment, requests can be sent which take effect on the dialog state. An attacker can inject falsified BYE or CANCEL requests and terminate the call or its establishment. For that reason it is crucial to consider the entire SIP session, not only the establishment of a call, as presented in our concept.

VI. Implementation

The presented concept was implemented at the endpoint side to analyze its behavior in practice and getting aware of the involved overhead. For the underlying SIP stack, the PJSIP - Open Source SIP Stack [12] was used. PJSIP is a complete SIP stack written in C. The PGP functionality is provided by GnuPG [18], which is an implementation of OpenPGP. To get access to GnuPG in PJSIP the library GnuPG Made Easy (GPGME) [9] was used. The following functionalities of the endpoints are implemented:

Callee: generation of the PGP challenge after receiving the initial *INVITE*, verification of the repeated *INVITE* and (if verification was successful) calculation of the signature and sending signed *200 OK* response

Caller: processing of the received PGP challenge, calculation of the signature and repeating the *INVITE*, verification of the signature in the received *200 OK* response and (if verification was successful) sending signed *ACK*

After the session initiation, all SIP messages were signed by the endpoints.

VII. Measurements

Since we wanted to investigate the overhead introduced by our approach, we compared the authentication with PGP to SIP Digest, and a call setup without any authentication.

The mechanisms were compared regarding their performance, not their security aspects. In our measurements, only the call setup between caller and callee was considered. The measured parameters were duration, memory consumption and CPU utilization.

A. Testbed and Scenarios

We used three nodes (each with Intel Core Duo E7500 CPU (2,93GHz), 2 x 2048MB Dual Channel DDR2 RAM, Gigbit Ethernet Interconnection) to setup a SIP Proxy (Kamailio v3.1.1) [11] and two SIP endpoints (PJSIP v1.8.5 with implemented PGP functionality). The underlying operating system was Debian 5.0.5 (Lenny) on each node.

We measured three scenarios: a) no authentication b) SIP Digest Authentication and c) PGP Authentication between the endpoints. The proxy was used with activated Digest Authentication in each scenario.

The message sequence is shown in Figure 6. For the measurement of the duration and the CPU utilization we only considered the call set up, which is labeled by two timestamps. The first timestamp is set when the initial *INVITE* is sent by the caller. When the caller sends the *ACK* after receiving the *200 OK* of the callee the call is successfully set up and the second timestamp is set. The memory consumption was measured for the whole SIP session. All measurements were done on the caller's side.

B. Results

For each scenario we performed 51 measurements and calculated the median for call set up duration and CPU utilization. The measurement of the memory consumption was done once by using valgrind [16]. The results are shown in Figures 7–9.

The overhead for authentication with PGP, i. e., the compute and memory demands at the caller's site, only slightly increase compared to no authentication or SIP Digest. The caused overhead is still acceptable. The CPU utilization rises by 10ms (see Figure 8), the need of memory increases by 2MB (see Figure 9). PGP Authentication increases the duration of the call setup by about 40ms compared to SIP Digest (see Figure 7). However, this delay is tolerable. The quality of the telephone service is not particularly affected.

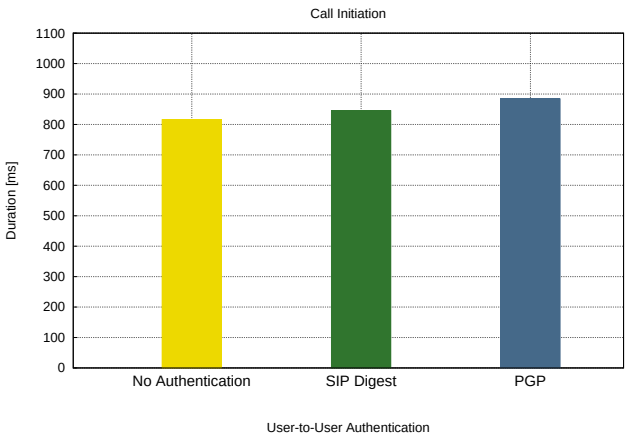

Figure 7. Median of Duration of Call Initiation

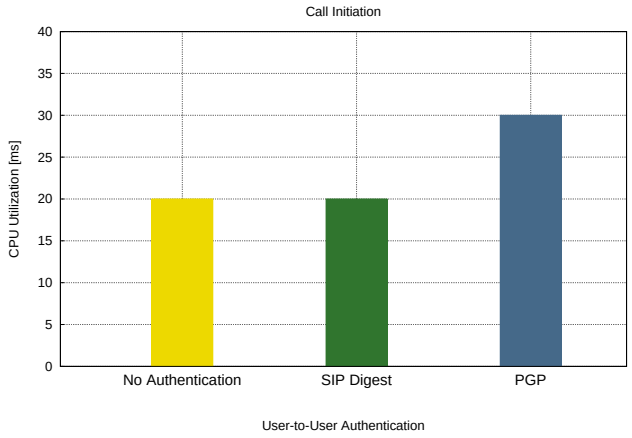

Figure 8. Median of CPU Utilization

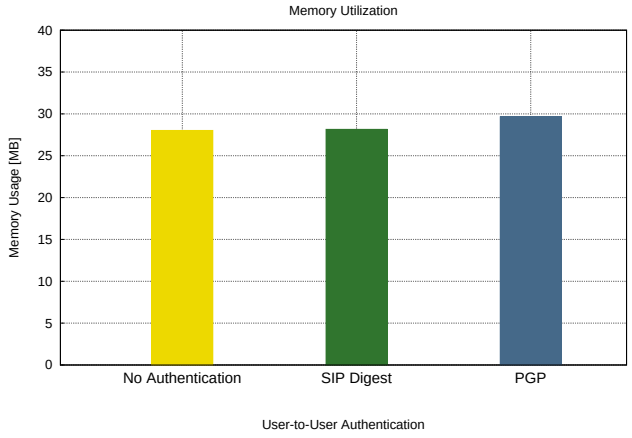

Figure 9. Memory Consumption

VIII. Conclusion and Future Work

We have shown how PGP signature can be used to secure SIP messages. We argued that an end-to-end and mutual authentication is necessary.

The measurements with our prototype have shown that the overhead is tolerable at the caller's side.

The PGP Authentication mechanism can be easily integrated in SIP infrastructures since necessary messages and header fields are already defined in RFC 3261.

The next step is to implement the mechanism also on the proxy side. The aim is to evaluate whether it is possible to use PGP Authentication with tolerable overhead also on these components of a SIP infrastructure.

References

[1] The H Security - Telecommunications regulator bars DigiNotar from issuing certificates. http://www.h-online.com/security/news/item/Telecommunications-regulator-bars-DigiNotar-from-issuing-certificates-1344786.html. Website accessed: September 30, 2011.

[2] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, and Rodney Thayer. OpenPGP Message Format. RFC 4880 (Proposed Standard), November 2007. Updated by RFC 5581.

[3] Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878.

[4] Carl Ellison and Bruce Schneier. Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure. In *Computer Security Journal*, 2000.

[5] Thomas Guillet, Ahmed Serhrouchni, and Mohamad Badra. Mutual Authentication for SIP: A semantic meaning for the SIP opaque values. In *Proceedings of New Technologies, Mobility and Security (NTMS)*, pages 1–6, November 2008.

[6] Mark Handley, Henning Schulzrinne, Eve Schooler, and Jonathan Rosenberg. SIP: Session Initiation Protocol. RFC 2543 (Proposed Standard), March 1999. Obsoleted by RFCs 3261, 3262, 3263, 3264, 3265.

[7] Weirong Jiang. A Lightweight Secure SIP Model for End-to-End Communication. In *Proceedings of the 10th International Symposium on Broadcasting Technology (ISBT)*, 2005.

[8] Alan B. Johnston. *SIP: Understanding the Session Initiation Protocol*. Artech House, 2^{nd} edition, 2004.

[9] Werner Koch and Wojciech Polak. Gnupg made easy [v1.2.0]. http://www.gnupg.org/, 2006.

[10] Lei Kong, Vijay Arvind Balasubramaniyan, and Mustaque Ahamad. A Lightweight Scheme for Securely and Reliably Locating SIP Users. In *Proceedings of 1st IEEE Workshop Voip Management and Security*, pages 9 – 17, April 2006.

[11] Ramona-Elena Modroiu, Bogdan Andrei Iancu, Daniel-Constantin Mierla, et al. Kamailio (openser) [v3.1.1]. http://www.kamailio.org/, December 02^{th}, 2010.

[12] Benny Prijono et al. Pjsip [v1.8.5]. http://www.pjsip.org/, October 20^{th}, 2010.

[13] Blake Ramsdell and Sean Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751 (Proposed Standard), January 2010.

[14] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922.

[15] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761.

[16] Julian Seward, Nicholas Nethercote, Tom Hughes, Jeremy Fitzhardinge, et al. Valgrind [v3.6.0]. http://www.valgrind.org/, October 2010.

[17] Lars Strand and Wolfgang Leister. Improving SIP authentication. In *Proceedings of The Tenth International Conference on Networks (ICN)*, pages 164 – 169, January 2011.

[18] The GNU Privacy Guard Team. The GNU Privacy Guard [v1.4.9]. http://www.gnupg.org/, 2008.

[19] Alexander Ulrich, Ralph Holz, Peter Hauck, and Georg Carle. Investigating the OpenPGP Web of Trust. In *16th European Symposium on Research in Computer Security (ESORICS 2011)*, pages 489–507, September 2011.

 ISBN: 978-1-61208-183-0