Migration towards a more secure authentication in the Session Initiation Protocol

Lars Strand
Norwegian Computing Center / University of Oslo
Oslo, Norway
lars.strand@nr.no

Wolfgang Leister
Norwegian Computing Center
Oslo, Norway
wolfgang.leister@nr.no

Alan Duric
Telio Telecom AS
Oslo, Norway
alan.duric@telio.no

***Abstract*—This paper specifies a two-step migration towards a stronger authentication in the Session Initiation Protocol. First, we add support for a Password Authenticated Key Exchange algorithm that can function as a drop-in replacement for the widely adopted Digest Access Authentication mechanism. This new authentication mechanism adds support for mutual authentication, is considered stronger and can rely on the same shared password used by the digest authentication. A more long-term solution is to replace the authentication scheme with the Simple Authentication and Security Layer. The Simple Authentication and Security Layer separates the authentication mechanisms from the Session Initiation Protocol, and adds support for a range of more secure authentication mechanisms in a generic and unified way. Both methods are presented, discussed, and shown how to integrate into the Session Initiation Protocol.**

***Keywords*—VoIP, SIP, authentication, PAKE, SASL.**

I. INTRODUCTION

Voice over IP (VoIP) is rapidly taking over for the traditional, public switched telephone networks (PSTN). Although there exist several competing network protocols that are capable of delivering VoIP, the Session Initiation Protocol (SIP) [1] and the Real-time Transport Protocol (RTP) [2] developed by the IETF have become the *de facto* industry standard. These two protocols fulfill two different functions – SIP is used for signaling, e.g., responsible for setting up, modifying and tearing down multimedia sessions, while RTP transports the actual media stream (voice). Although the SIP protocol is flexible and rich in functionality [3], several vulnerabilities and security attacks have been found [4]–[6].

Securing a SIP-based VoIP system has proven challenging and the reasons are multi-faceted:

- The scale and complexity of the SIP protocol specification, with primary focus on functionality rather than a sound security design [7].
- SIP usage of intermediaries, expected communication between nodes with no trust at all, and its user-to-user operation make security far from trivial [1, page 232].
- A large number of threats against VoIP systems have been identified [8]. Several security mechanisms for countermeasures have been proposed, but no single security mechanism is suited to address all these security threats concerning VoIP and SIP [9], [10].
- Since the SIP and RTP protocols share the same infrastructure as traditional data networks, they also inherit the security problems of data communication.
- VoIP services have strict requirements to the network performance with respect to Quality of Service since it is a duplex communication with low tolerance for latency, packet loss and saturation. Introducing strong security mechanisms might affect network performance [11].

Signaling in PSTN has traditionally involved trust between carriers, and by end users (caller-id). To achieve the same trust level using SIP, we need to employ secure authentication.

In VoIP, authentication tries to validate the identity of the communication peers and to bind that identity to a subject (peer). It must be stressed that the user is not authenticated, but the user's phone. In VoIP terminology, a subject could be a User Agent (UA), such as a phone, identified by a phone-number/username and IP-address/hostname pair, denoted as an Address-of-Record (AoR). The authentication in VoIP is therefore the assurance that a communicating entity, the UA, is the one that it claims to be [12]. However, the authentication in SIP has proven weak [13] and vulnerable to a real-world security attack [14].

Equally important for the UA is to establish the identity of the communicating peer, i.e., the SIP server. If the client does not authenticate the SIP server, it might risk to communicate and send content to a hostile SIP server.

The main contribution of this paper is to propose a migration towards a more secure SIP authentication. First, we introduce an authentication method based on the Password Authenticated Key Exchange (PAKE) [15], which provides mutual authentication based on a shared secret, and can function as a drop-in replacement of the digest authentication currently used. However, a more flexible authentication method is desired. As a second authentication method, we propose the Simple Authentication and Security Layer (SASL) [16], which enables SIP to transparently support and use more secure authentication methods in a unified and generic way.

The rest of the paper is organized as follows: In Section II we give an introduction to authentication in SIP and discuss related work. In Section III we show how a modified PAKE can be used to add mutual authentication in SIP. The second authentication mechanism added to SIP, SASL, is explained

 ISBN: 978-1-61208-010-9

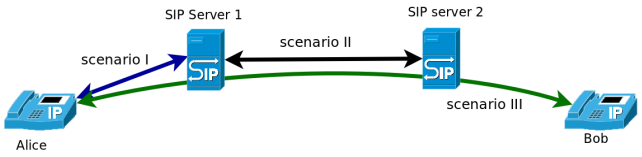

Figure 1: Three different usage scenarios where authentication in SIP is desired.

and discussed in Section IV. Conclusion and future work is presented in Section V.

II. Identity in the Session Initiation Protocol

We identify three scenarios where identity in SIP needs to be handled, as depicted in Figure 1: Scenario *I* between the UA and the local SIP server; Scenario *II* between SIP servers; and Scenario *III* end-to-end.

Scenario I between the UA and the local SIP server is relevant when the UA comes online and before any outgoing calls can be placed. Then, the UA must register itself to a local SIP server. During the SIP register handshake, the server usually challenges the UA to authenticate. Before placing a call (sending a SIP `INVITE`), the UA might be challenged again by the server to authenticate. The most common authentication method used between UA and server today is the Digest Access Authentication (DAA) [17].

Scenario II handles the authentication between SIP servers to achieve trust between SIP servers. It is not desirable to have SIP traffic handled by an unknown or untrusted SIP server that might have malicious intent. However, since most SIP servers today use some kind of SIP peering [18], the relationships between servers are often static and pre-defined. Therefore the identities between SIP servers are often predetermined by other security mechanisms than what are offered by SIP (like IPSec, TLS etc.).

Scenario III is about end-to-end authentication, which determines the identity of both the caller and the callee across different SIP domains. This is of particular importance and not easily attained in SIP. There is an increased threat and fear for both VoIP phishing and SPIT (Spam over Internet Telephony), that might seriously affect SIP-based VoIP services. By enforcing end-to-end authentication in SIP, these threats might be mitigated or prevented.

We list authentication mechanisms in SIP and their support in these three SIP scenarios in Table I. We shall discuss these authentication mechanisms in the following.

The DAA is currently the most common authentication mechanism for SIP. DAA is simple but rather insecure. It is the only authentication mechanism which support in SIP is mandatory [1, Section 22]. DAA uses the MD5 hash function and a challenge-response pattern, and relies on a shared secret between client and server within a SIP domain [17]. DAA is performed during the SIP `REGISTER` handshake between the UA and the SIP server, as depicted in messages 1-3 and 6 in Figure 2. The UA receives a nonce value from the SIP server, computes a digest hash value over the nonce, the shared secret and some other SIP header values, and send it to the SIP server. The SIP server computes the same digest hash. If both digests are identical, the UA is authenticated. The DAA is weak and vulnerable to a serious real-world attack [14]. Since the DAA relies on a shared secret and is only meaningful for a specific realm, its usage is limited to Scenario I.

Secure MIME (S/MIME) [19] is an authentication mechanism presented in the SIP core specification document RFC3261 [1]. S/MIME intends to achieve end-to-end authentication between UAs. The entire SIP message is encapsulated in a specific SIP message using MIME, which is signed and optionally encrypted. The receiving UA checks whether the sending UA's certificate is signed by a trusted authority. Since S/MIME depend on end-user certificates, the UAs must support multiple root certificates since no consolidated certificate authority exists. Additionally, certificate handling issues, such as revocation and renewal, complicate the use of certificates. There has been rather limited industry support for S/MIME.

Palmieri et al. [20] introduce a new authentication mechanism using digital signatures. But since they rely on certificates, their solution suffers under similar certification handling issues as S/MIME. They also admit that relying on public key infrastructure (PKI) is both difficult and costly to implement. Liao et at. [21], propose an improved authentication in SIP with self-signed public keys on elliptic curves. However, Liao's proposal uses smart-cards to store authentication data and rely on a trusted third party [22].

The SIP protocol needs an authentication mechanism that avoids the security vulnerabilities the currently used DAA has. A replacement authentication mechanism should preferably not rely on PKI, have support for strong mutual authentication, and support all three scenarios listed previously in this section.

III. Password Authenticated Key Exchange

We propose to add support for a variant of PAKE denoted as "Key Agreement Method 3" (KAM3) as a cryptographic protocol [15, page 17]. PAKE has the following attractive features: *1*) PAKE provides mutual authentication between UA and the SIP server, and thus a rogue SIP server can not claim that the authentication succeed without knowing the shared password. PAKE assures the UA that the SIP server knows the UA's encrypted password. *2*) Reuse of the shared password used by DAA as the UA's credential, which enables our approach to easily replace DAA used within a local SIP domain (scenario I). *3*) PAKE offers strong protection of the shared secret if the communication is eavesdropped, that prevents brute-force attacks, including dictionary-based off-line attacks, to which the DAA is vulnerable to.

Our approach follows the work of Oiwa et al. [23]. They use KAM3 to introduce a stronger authentication in HTTP and their initial design and specification is submitted to the IETF as an Internet Draft [24]. We have adapted their approach to SIP, since SIP closely resembles HTTP in both message structure and flow, and we need to prevent the `REGISTER` hijack attack presented earlier [14].

 ISBN: 978-1-61208-010-9

Authentication mechanisms	Supported authentication scenarios			Supported SIP methods	
	scenario I	scenario II	scenario III	REGISTER	INVITE
Digest authentication	yes	no	no	yes	yes
PAKE	yes	no	no	yes	yes
SASL	yes	yes	yes	yes	yes
GSS-API	yes	yes	yes	yes	yes
S/MIME	no	no	yes	no	yes

Table I: List of SIP authentication mechanisms and their support.

In KAM3, the UA and the SIP server compute cryptographic keys based on the shared password. These keys are exchanged, and a shared session secret is computed based on these keys. Each peer then computes a hash value of the session secret and some other values, which is sent to the requesting peer. The receiving peer computes the same hash value, and compares it with the received hash value. If these are identical, the sending peer is authenticated.

PAKE supports several authentication algorithms, which differ in their underlying mathematical groups and security parameters [24]. The only mandatory supported authentication algorithm, the *iso-kam3-dl-2048-sha256*, uses the 2048-bit discrete-logarithm defined in RFC3526 [25] and the SHA-256 hash function.

A. Initial requirements

In the following section, we let q an odd prime integer defining the number of elements in $F(q)$ which is a representation of a finite group. We let g the generator of a subgroup of r elements in $F(q)$. The one-way hash function is denoted as H.

Before the authentication starts, username and password must be set and configured. We compute a weak secret π used by the client as a one-way hash of the values $realm$, $username$ and $password$:

$$\pi = H(realm, username, password)$$

Here, $realm$ is the protection domain where SIP authentication is meaningful for a set of $username$ and $password$. The server does not need to store the shared password directly, only a specially encrypted version $J(\pi)$, where J is the password verification element derivation function defined as:

$$J(\pi) = g^{\pi} \bmod q$$

B. Message exchange

We need to extend the current SIP `REGISTER` handshake by one extra round-trip of SIP messages between the UA and the SIP server. These two extra messages are depicted in blue and numbered (4) and (5) in Figure 2. A more detailed specification is given in the following paragraphs, where the numbers refer to the protocol clauses depicted in Figure 2.

The UA registers to a SIP *location service* (SIP server). The initial SIP `REGISTER` message (1) from the UA is not authorized, and must be authenticated. The SIP server responds with a `401 Unauthorized` status message (2),

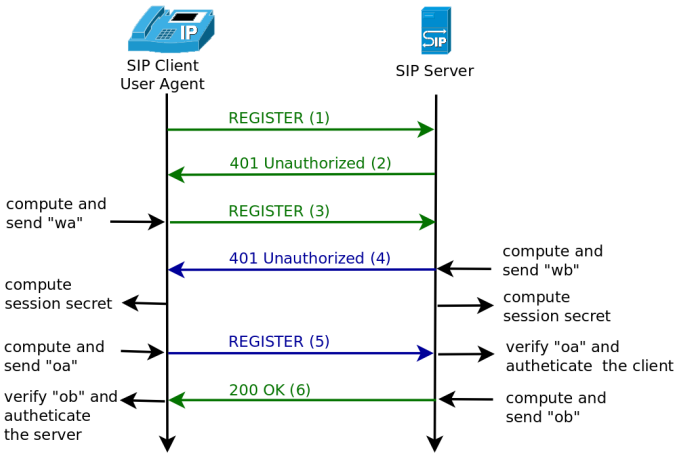

Figure 2: SIP `REGISTER` message flow with mutual authentication security using PAKE.

which contains a `WWW-Authenticate` header with details of the challenge, including *realm* and *algorithm*. The UA constructs a cryptographic value w_a generated from a random integer s_a:

$$w_a = g^{s_a} \bmod q$$

This value is sent in a new SIP `REGISTER` message (3) to the SIP server. The SIP server proceeds to generate and send another cryptographic value w_b, which is generated from $J(\pi)$, the received value w_a and a random integer s_b:

$$w_b = (J(\pi) \times w_a^{H(1,w_a)})^{s_b} \bmod q$$

At the next step, each peer computes a session secret z. The UA derives z based on π, s_a, w_a and w_b:

$$z = w_b^{(s_a + H(2,w_a,w_b))/(s_a * H(1,w_a) + \pi) \bmod r} \bmod q$$

Likewise, the SIP server derives z based on s_b, w_a and w_b using the following function:

$$z = (w_a \times g^{H(2,w_a,w_b)})^{s_b} \bmod q$$

The session secret z matches only if both peers have used the secret credentials generated from the same shared secret. The above equations are directly derived from the PAKE HTTP authentication specifications [24]. The next step is to validate the value of z at the communicating peer.

 ISBN: 978-1-61208-010-9

The UA sends a third SIP `REGISTER` message (5) and includes the value o_a which is a hash value computed as:

$$o_a = H(4, w_a, w_b, z, contactURIs)$$

Here, $contactURIs$ is the value of the UA's `Contact` SIP header value. This value is integrity-protected to prevent register hijacking attacks as presented in [14]. The SIP server, upon receipt of o_a, performs the same hash operation, and compares the results. If these results are identical, the UA is authenticated. The SIP server then sends a final message (6), with the value o_b computed as:

$$o_b = H(3, w_a, w_b, z, contactURIs)$$

When the UA receives o_b, it verifies this value by computing its hash value. If the results are identical, the SIP server is authenticated to the UA. After a complete message exchange, the UA is authenticated to the SIP server, and the SIP server has been authenticated to the UA.

C. SIP message syntax

A SIP message consists of several headers and a body. The SIP header fields are textual, always in the format `<header_name>: <header_value>`. The header value can contain one or more parameters. We embed the cryptographic values derived in the previous section as base64-encoded [26] SIP header values. We re-use the SIP DAA headers to carry PAKE authentication data, so that PAKE can be used as a drop-in replacement for DAA. A SIP `REGISTER` message with a DAA `Authorization` header is depicted in Figure 4. Again, we refer to the protocol clauses with a number in parentheses as depicted in Figure 2.

The UA first sends a SIP `REGISTER` without any authentication credentials (1). The SIP server responds with a `401 Unauthorized` status message (2), which contains a `WWW-Authenticate` header with header values $realm$ and $algorithm$:

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: Mutual realm="asterisk",
  algorithm="iso-kam3-dl-2048-sha256"
```

The UA then computes w_a and sends it to the SIP server using a new SIP `REGISTER` message (3), with the required values embedded in the `Authorization` header:

```
SIP/2.0 REGISTER
Authorization: Mutual user="alice",
  algorithm="iso-kam3-dl-2048-sha256",
  wa="Q29tcHV0ZWQgd2E...ljaCBcyBsb25nCg=="
```

The next required values in the authentication mechanism w_b, o_a and o_b are embedded and sent using these two SIP headers.

IV. Simple Authentication and Security Layer

The Simple Authentication and Security Layer (SASL), defined in RFC4422 [16], provides an interface for authentication and an authentication negotiation mechanism. The SASL specification is developed and maintained within the IETF, and have been scrutinized by security professionals over the years. It has been extensively tested, and is now classified as a mature standard by the IETF. SASL is also implemented and used in several popular communications protocols applications like IMAP, SMTP and LDAP[1].

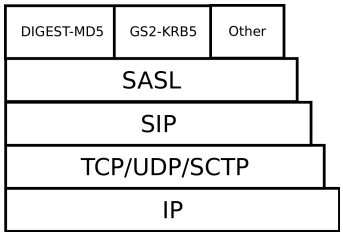

Figure 3: The SIP protocol stack with SASL and underlying security mechanisms.

The SASL framework does not provide authentication mechanisms in itself, but supports different underlying authentication mechanisms through a standardized interface[2]. SASL does not provide a transport layer and thus relies on the application to encapsulate, send and extract SASL messages between client and server. The SASL messages sent between client and server contain authentication data, and are opaque from the viewpoint of the calling application. The application only needs to add support to a SASL software library implementation, and thus have support to a range of underlying authentication mechanisms the library supports.

Adding support for the security abstraction layer framework Generic Security Services API (GSS-API) has been done earlier [27]. While the GSS-API is intended for use with applications, SASL is used in, and intended for, communication protocols. The functionalities offered by the GSS-API and SASL are alike, but the SASL specification is more high-level, and allows more freedom in implementing the SASL requirements. SASL also supports more underlying security mechanisms than the GSS-API. By using the "GS2" mechanism family, the GSS-API can be used as an underlying security mechanism in SASL. However, the GSS-API negotiation mechanism cannot be used due to security concerns [28, Section 14].

A. SASL profile for SIP

A modified PAKE authentication can more easily replace the current digest (DAA) authentication used in SIP, since they both rely on a shared secret and use the same SIP headers. PAKE also introduces a stronger authentication than DAA. However, a more flexible authentication mechanism is desired. Different VoIP scenarios require different security requirements, and the communicating peers should be able

[1]The Carnegie Mellon University's implementation: http://asg.web.cmu.edu/sasl/ and the GNU SASL library: http://www.gnu.org/software/gsasl/ are two popular and freely available SASL libraries.

[2]A list of registered SASL mechanisms is maintained by IANA: http://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xml

 ISBN: 978-1-61208-010-9

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: Digest
    username="alice",realm="asterisk",nonce="3b7a1395",response=
    "ccbde1c3c129b3dcaa14a4d5e35519d7",uri="sip:CompanyA",
    algorithm=MD5
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: SASL mechanism="DIGEST-MD5"
    data="YAzgusSGGeRFGw9nfUvOAxcEDzZCBmKYlH2ElnegaccBcx3DUSkGNW
    Y4qfiSwcXwjLtoqW0eBNog7ixHN"
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

Figure 4: A SIP `REGISTER` message with the original DAA `Authorization` header to the left, and the same header carrying SASL data to the right.

to negotiate the best possible authentication mechanism supported.

Instead of adding numerous different authentication mechanisms to SIP based on different security requirements, it is desirable to keep the changes to the SIP standard to a minimum. The industry might also be reluctant to adopt immature and non-standardized security services, like different (new) authentication mechanisms. Adding support for SASL requires only small changes to the SIP standard, and can utilize several underlying authentication mechanisms. The SIP protocol stack with SASL is shown in Figure 3.

In SASL terminology, the description on how to encapsulate SASL negotiation and SASL messages for a given protocol, is called a "SASL profile". We create a SASL profile for SIP by reusing the `WWW-Authenticate` and `Authorization` SIP headers used by the digest authentication, shown earlier. Instead of encapsulating DAA data, we embed SASL messages, as depicted in Figure 4.

When discussing PAKE authentication earlier, we added one round-trip of SIP messages between the UA and the SIP server. When using SASL, the number of messages going back and forth depends on the underlying authentication mechanism. We therefore extend the SIP `REGISTER` handshake with an arbitrary number of round-trips, until the underlying authentication mechanism has completed communication.

In the following paragraphs, the numbers in parentheses refer to the SIP message numbers in Figure 2. The SASL specification only outlines a very high-level method of how the server should advertise its supported mechanisms to the client. We implement the mechanism negotiation in the first three messages in the SIP `REGISTER` handshake (1-3). The UA starts by requesting authentication from the SIP server, with no `Authorization` header (1). The SIP server responds with a `401 Unauthorized` SIP message (2), with the supported and available mechanisms embedded in the `WWW-Authenticate` header:

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: SASL
  negotiate="DIGEST-MD5 NTLM GS2-KRB5"
```

The client selects the best mechanism from the received list that it supports and sends a new SIP `REGISTER` message (3). This message includes an `Authorization` header requesting authentication with "GS2-KRB5" as the preferred mechanism. The initial authentication data is embedded base64 encoded to the *data* parameter:

```
SIP/2.0 REGISTER
Authorization: SASL mechanism="GS2-KRB5",
  data="SUZZT1VDQU5SR...JUPVVQU5FUkQK="
```

The server retrieves the SASL data, and passes the message to the SASL library which handles the authentication. The selected authentication method continues to pass SASL messages between client and server as many times as necessary to complete the authentication (messages 4-5 are repeated). Once the authentication is complete, the SIP server sends a `200 OK` SIP message. Should the server have some last SASL data to be communicated to the client to complete the authentication, it can be carried in a `WWW-Authenticate` header embedded in the `200 OK` message:

```
SIP/2.0 200 OK
WWW-Authenticate: SASL mechanism="GS2-KRB5",
  data="TFoG9rP56zrVH...YaAOndwPew6NdxKr"
```

As soon as the `200 OK` message is received and processed, the client is authenticated to the SIP server. Since the mechanism negotiation is not integrity-protected, the UA is vulnerable to a "down-grade" attack. An attacker can intercept and modify the negotiation messages so that the least favorable authentication method is used.

V. Conclusion and Future Work

In our earlier work, we have shown and implemented a real-world attack to the widely deployed DAA method [27]. In this paper we have added support to a new improved authentication mechanism that can easily replace DAA based on a modified PAKE algorithm. This new authentication mechanism adds support for mutual authentication and is more secure than DAA. We have also shown that the modified PAKE authentication can easily function as a drop-in replacement for DAA. However, a more flexible authentication mechanism is desired in the long-term.

 ISBN: 978-1-61208-010-9

Our second authentication mechanism supported in SIP is SASL, which is not an authentication mechanism *per se*, but introduces a security abstraction layer. This abstraction layer adds support to a range of underlying authentication mechanism in a unified way. As long as SIP supports SASL, new authentication mechanisms can be added later to the SASL library, without requiring any change to the SIP protocol. We have also introduced a SASL mechanism negotiation that enables the communicating peers to agree upon the "best" available authentication mechanism.

We envisage a two-step migration towards a stronger authentication scheme in SIP. First, the modified PAKE authentication is implemented and deployed. Second, the long-term solution is to deploy SASL with support for a range of underlying authentication mechanisms.

Future work will look into implementing a proof of concept for PAKE-enabled UA and SIP server, including overhead evaluation benchmarks for the new authentication algorithm. We also plan to evaluate different SASL security mechanism and their implications for SIP, and decide which authentication mechanisms should be mandatorily supported through SASL. A co-operation with the IETF and the kitten Working Group to further elaborate a SASL profile for SIP is also planned. We hope our work will gain acceptance and industrial deployment, so that the previously mentioned security attacks can be countered.

ACKNOWLEDGMENT

This research is funded by the EUX2010SEC project in the VERDIKT framework of the Norwegian Research Council (Norges Forskningsråd, project 180054). The authors would like to thank the anonymous reviewers for their valuable comments on earlier drafts of this paper.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt 11. Apr 2011

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt 11. Apr 2011

[3] H. Sinnreich and A. B. Johnston, Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., August 2006.

[4] H. Dwivedi, Hacking VoIP: Protocols, Attacks, and Countermeasures, 1st ed. No Starch Press, Mar. 2009.

[5] D. Endler and M. Collier, Hacking Exposed VoIP: Voice over IP Security Secrets and Solutions. McGraw-Hill Osborne Media, November 2006.

[6] A. M. Hagalisletto and L. Strand, "Designing attacks on SIP call set-up," International Journal of Applied Cryptography, vol. 2, no. 1, pp. 13–22(10), July 2010.

[7] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, and H. Schulzrinne, SIP Security. WileyBlackwell, Mar. 2009.

[8] VoIPSA, "VoIP security and privacy threat taxonomy," Public Realease 1.0, Oct. 2005. [Online]. Available: http://voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf 1. Nov 2011

[9] D. York, Seven Deadliest Unified Communications Attacks. Syngress, Apr. 2010.

[10] P. Park, Voice over IP Security. Cisco Press, Sep. 2008.

[11] S. Salsano, L. Veltri, and D. Papalilo, "SIP security issues: The SIP authentication procedure and its processing load," Network, IEEE, vol. 16, pp. 38–44, 2002.

[12] International Telecommunication Union (ITU), "Security Architecture For Open Systems Interconnection (OSI)," The International Telegraph and Telephone Consultative Comittee (CCITT), X.800 Standard X.800, 1991.

[13] A. M. Hagalisletto and L. Strand, "Formal modeling of authentication in SIP registration," in Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE '08. IEEE Computer Society, August 2008, pp. 16–21.

[14] L. Strand and W. Leister, "Improving SIP authentication," in Proceedings of the Tenth International Conference on Networking (ICN2011). Xpert Publishing Services, Jan 2011, pp. 164 – 169.

[15] International Organization for Standardization and ISO, "ISO/IEC 11770-4:2006: Information technology – Security techniques – Key management – Part 4: Mechanisms based on weak secrets," 2006.

[16] A. Melnikov and K. Zeilenga, "Simple Authentication and Security Layer (SASL)," RFC 4422 (Proposed Standard), Internet Engineering Task Force, Jun. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4422.txt 11. Apr 2011

[17] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), Internet Engineering Task Force, Jun. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt 11. Apr 2011

[18] L. Strand and W. Leister, "A Survey of SIP Peering," in NATO ASI - Architects of secure Networks (ASIGE10), May 2010.

[19] J. Peterson, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)," RFC 3853 (Proposed Standard), Internet Engineering Task Force, Jul. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3853.txt 11. Apr 2011

[20] F. Palmieri and U. Fiore, "Providing true end-to-end security in converged voice over IP infrastructures," Computers & Security, vol. 28, no. 6, pp. 433–449, Sep. 2009.

[21] Y. Liao and S. Wang, "A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves," Computer Communications, vol. 33, no. 3, pp. 372–380, Feb. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366409002631 11. Apr 2011

[22] Y. Liao, "Secure password authenticated key exchange protocols for various enviroments," Ph.D. dissertation, Tatung University, Dec. 2009.

[23] Y. Oiwa, H. Watanabe, and H. Takagi, "Pake-based mutual http authentication for preventing phishing attacks," CoRR, vol. abs/0911.5230, 2009. [Online]. Available: http://arxiv.org/abs/0911.5230 11. Apr 2011

[24] Y. Oiwa, H. Watanabe, H. Takagi, Y. Ioku, and T. Hayashi, "Mutual Authentication Protocol for HTTP," Internet Engineering Task Force, Oct. 2010. [Online]. Available: http://tools.ietf.org/html/draft-oiwa-http-mutualauth-08 11. Apr 2011

[25] T. Kivinen and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)," RFC 3526 (Proposed Standard), Internet Engineering Task Force, May 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3526.txt 11. Apr 2011

[26] S. Josefsson, "The Base16, Base32, and Base64 Data Encodings," RFC 4648 (Proposed Standard), Internet Engineering Task Force, Oct. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4648.txt 11. Apr 2011

[27] L. Strand, J. Noll, and W. Leister, "Generic security services API authentication support for the session initiation protocol," in Proceedings of The Seventh Advanced International Conference on Telecommunications (AICT2011). Xpert Publishing Services, Mar 2011, pp. 117 – 122.

[28] S. Josefsson and N. Williams, "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family," RFC 5801 (Proposed Standard), Internet Engineering Task Force, Jul. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5801.txt 11. Apr 2011

 ISBN: 978-1-61208-010-9