Proxy-based Security for the Session Initiation Protocol (SIP)

Holger Schmidt
Institute of Distributed Systems
Ulm University
Germany
Email: holger.schmidt@uni-ulm.de

Chi-Tai Dang
Institute of Distributed Systems
Ulm University
Germany
Email: chitai.dang@uni-ulm.de

Franz J. Hauck
Institute of Distributed Systems
Ulm University
Germany
Email: franz.hauck@uni-ulm.de

***Abstract*— There is a trend towards voice-over-IP systems based on the Session Initiation Protocol (SIP), which is a protocol for session management in general. However, as signalling data is transferred using the Internet, systems face security problems. Thus, at least authentication of the participants and confidentiality of signalling data have to be ensured as basic mechanism. In this work, we propose a mechanism for assuring the identity of a group of users fulfilling the same role (e.g., employees of a customer call centre). Using our concept enables using only one certificate for the whole group for signing and encrypting messages according to the SIP standard. Our mechanism works transparently for users as we provide a special proxy server for this purpose, which significantly reduces administration efforts and resource needs on the participating nodes. Furthermore, such a proxy server can be used for transparently validating and decrypting SIP messages as well. This reduces efforts on the terminals, resulting in an improved resource-usage, e.g., on a Personal Digital Assistant. We provide an implementation of the concept based on the NIST SIP proxy server.**

I. Introduction

Currently, there is a recognizable trend from public switched telephone networks (PSTN) to voice-over-IP (VoIP) telephone networks based on the Session Initiation Protocol (SIP). SIP is a standardized protocol for session management in general, e.g., for establishing a VoIP call. Therefore, SIP specifies central network entities, SIP proxies, which are responsible for routing SIP messages (e.g., session establishment requests).

VoIP technology becomes more and more popular as many companies offer VoIP-ready hardware which reduces administration efforts and even enables technically non-experienced people to use the technology. Furthermore, using VoIP can reduce costs for companies because of using IP hardware that nowadays is in most cases already existent, instead of having to invest in separate expensive PSTN hardware. Additionally, calls within the VoIP network are in general free of additional charge (beside the Internet provider costs).

Because of these cost reduction possibilities, many customer call centres have adapted their infrastructure to VoIP. One more reason for this is that these call centres often have national telephone numbers, however, the actual call-centre agents are located in an office in a low-wage country that is connected using VoIP for reasons of economy. Furthermore, VoIP enables a cheaper realization of multi-channel customer call centres supporting VoIP, text chat or call-back.

However, a threat to VoIP technology is the usage of the Internet for communication without taking security into account. Especially when talking about sensitive data (e.g., bank account information) with a call-centre agent, the system has to ensure that (1) an authorized person answers the call and that (2) the data is transferred in a secure way. Otherwise, criminals are able to intercept these calls pretending to be a call-centre agent, or are at least able to listen to the conversation.

For ensuring a participant's identity, SIP allows digitally signing messages using S/MIME. Additionally, SIP allows encrypting messages using S/MIME encryption, which enables a secure signalling traffic. However, both approaches entail additional expenses for the participants. These have to create private/public keys, get these keys signed at a certificate authority (CA), and install these into their VoIP application. This leads to more effort for system administrators.

External signing and encryption is an approach to keep complexity out of these systems. Currently, this approach is used for transparently securing e-mail traffic by integrating a special kind of mail proxy server, e.g., using PGP [1].

In this paper, we propose an extension of the standard SIP proxy behaviour. Typically, SIP proxies are responsible to only forwarding messages from a sending terminal to a receiver. In contrast, our SIP proxy is additionally able to sign and encrypt messages. This is especially useful when used for a specific group of SIP accounts (e.g., customer call-centre employees, company support employees). Then, a group-based private key is installed on a central SIP server that handles signing and encrypting messages on behalf of the users. This eases administrative efforts and can even save resources on terminals, which do not have to spend computing resources for this purpose. Especially on mobile and pervasive devices, which are characterized by resource limitations, this can, e.g., save battery power.

Validation and decryption of SIP messages has to be done by the receiving terminals. However, there are no administrative efforts for this purpose. In general, VoIP applications are able to process encrypted and signed messages on their own. Therefore, digital certificates have to be available at public key servers for downloading.

However, on resource-limited pervasive devices, validation

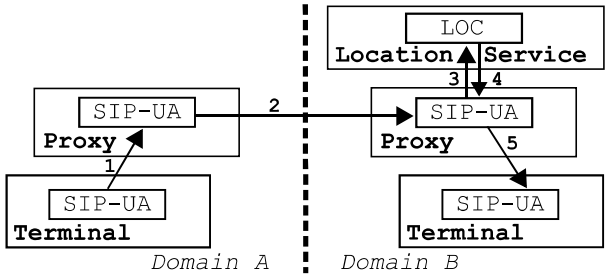

Fig. 1. SIP session establishment message flow

and decryption can be sourced out to a local proxy server as well. Therefore, we additionally propose an extension of the standard proxy server behaviour, which is able to validate and decrypt messages and then forward the message to the receiver with an indication of the message's validity.

The paper is structured as follows. First, we give basic background information on SIP and show current security concepts for SIP. After discussing related work in Section III, we present our concept of proxy-based security in Section IV. Then, for demonstrating the feasibility of our approach, we show our integration of the concept into a standard SIP proxy server in Section V. Finally, in Section VI we conclude and show possible future work.

II. Session Initiation Protocol

In this section, we first give basic background information on SIP. Then, as SIP messages are transferred using the Internet, we discuss security problems and their possible solutions.

A. Basics

SIP is a text-based Internet protocol that was specified by the Internet Engineering Task Force (IETF) in RFC 3261 [2]. The protocol is built on Internet standards, i.e., the Internet Protocol (IP), the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). SIP is used for session management and coordination. Currently, the protocol is widely used for managing multimedia sessions, particularly for signalling VoIP calls. The specification defines a set of request and response messages (client/server model) and the behaviour of affected network entities.

Figure 1 shows the message flow and affected entities for establishing a session. For initiating a session between two terminals, a so-called INVITE message has to be sent. A *user agent* is a logical SIP entity that is actually responsible for establishing, modifying and terminating sessions. Therefore, it is able to send and to receive appropriate SIP messages. Messages are sent to an initially configured SIP *proxy* containing a unique SIP URI, which identifies the target user agent. SIP proxies are responsible for forwarding messages to the target user agent. If the receiver's SIP URI is located in a remote domain, the message is forwarded to a proxy that is responsible for the specific domain (cf. Fig. 1). The address of a responsible proxy is obtained by retrieving an SRV record from the Domain Name System (DNS) [3]. If the receiver is located in the proxy's domain, a *location service* is queried for the actual communication contact address, to which the request is forwarded. Therefore, user agents have to initially register their contact data using a so-called REGISTER message. This message is sent to a *registrar*, which is able to store the information into the location service that represents some kind of data base.

The SIP protocol has reached a mature state, which results in a number of stable open source implementations, e.g., the NIST SIP protocol stack [4].

B. Security

As mentioned in Section II-A, SIP is built on Internet standard protocols. Thus, SIP has to handle equal security issues compared to other standard application layer Internet protocols (e.g., HTTP, FTP).

SIP is in the majority of cases used for managing VoIP sessions, which have a general need for securing the session signalling and the session communication. As SIP is only responsible for the control of multimedia sessions, securing session communication itself has to be handled using other protocols, e.g., using the Secure Real-time Transport Protocol (SRTP) [5].

However, securing session signalling is an important issue as well. Otherwise, session control messages could be intercepted (man-in-the-middle attack) and unauthorized user agents could illegitimately take over sessions. Thus, for VoIP calls, this results in malicious user agents pretending to be the authorized callee. This is a very severe problem when talking about sensitive data to previously unknown persons, who cannot be recognized by their voices (e.g., bank customer call centre). Another threat is unauthorized CANCEL messages that allow man-in-the-middle attackers to instantly cancel active VoIP sessions.

The SIP specification includes mechanisms for ensuring user authentication and data transfer confidentiality using S/MIME encryption and signing, which actually has to be done by the user agents on their own.

Digital signing of SIP messages using S/MIME is realized by attaching a digital signature of the complete SIP message, which is additionally attached as MIME type *message/sip* as well. This basic mechanism can be optimized by only transferring and signing a so-called *Authenticated Identity Body* (AIB) [6], which contains only necessary headers for digitally signing a SIP message, which is attached as MIME type *message/sipfrag* [7].

For basic confidentiality, Transport Layer Security (TLS) can be used according to the SIP standard, which allows a secure transfer of messages between SIP entities. For even preventing SIP proxies to read confidential data, transferred SIP messages can be encrypted using S/MIME. However, it is not possible to encrypt the complete SIP message, as proxies rely on specific headers for message routing (To, From, Call-ID, CSeq, Contact). Thus, these headers have to be transferred unencrypted with the attached S/MIME-encrypted original SIP

message. This mechanism can be optimized by attaching a *message/sipfrag* body containing only data that should be encrypted [6].

Both mechanisms for certification and encryption of SIP messages have not been widely-used within current VoIP applications so far. We think that this results from the fact that there is a lack of technical understanding of common users, which are not able to handle administrative tasks of creating and installing needed certificates.

Another issue is certificate management for groups, e.g., call-centre agents should sign their messages using a call-centre certificate (see Section IV-A). A solution to this would be to sign every agent's certificate using the call centre's certificate. However, in most cases, call-centre agents should not be contacted directly. Instead, a group-based SIP URI is addressed, whose certificate is used for signing SIP messages. This would result in installing the call centre's certificate on every agent's VoIP application and thus cause more administrative efforts.

When a digitally signed SIP message is received, a user agent is able to validate the message. Therefore, the certificate has to be installed on this machine. This can in the simplest case be done manually by downloading the appropriate certificate or automatically. An automatic approach can be based on receiving and installing the certificate that is attached in the SIP message body. Therefore, for ensuring a level of trust, this certificate has to be signed by trusted parties or a special certificate authority [8], [9]. A SIP-only solution proposes to use a special kind of credential service for discovering other user agent's certificates, from which certificates can be downloaded [10].

III. Related Work

For a user-friendly proxy-based integration of security, there is some work for transparent certification of e-mails based on PGP and S/MIME, respectively. Amongst others, there is the PGP certification proxy [1] and the Z1 SecureMail Gateway [11], which are able to digitally sign outgoing e-mail messages and validate incoming e-mail messages. Both applications are able to transparently encrypt and decrypt e-mail messages as well. However, there is no support for the SIP protocol.

There is related work for reducing e-mail SPAM. RFC 4871 introduces DomainKeys identified mail (DKIM) signatures [12]. Within this work, mail servers are able to automatically and transparently sign messages. This is comparable to our presented concept, however, there is no support for encryption and the SIP protocol.

For the HTTP protocol, there are similar approaches for hiding the user's identity. There, proxies, e.g., Anonymizer [13] and JAP [14], perform the task of anonymising the user's IP address. The user only has to ensure that the browser uses the anonymising proxy. These systems do not support the SIP protocol.

In general, these concepts can be transferred to the SIP protocol. However, there is no such approach for the SIP protocol so far. In the next sections, we present our approach of a user-friendly integration of proxy-based security into the SIP protocol.

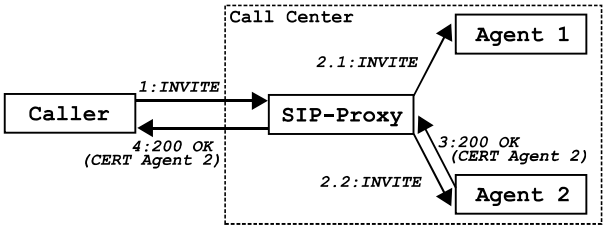

Fig. 2. Caller establishing call with validated call-centre agent 2

IV. Proxy-based Security

In this section we provide our solution for transparent integration of security, especially for a group of users fulfilling the same role (e.g., call-centre agents), into SIP networks. First, we describe the standard scenario in SIP networks according to Figure 2. Then, we present our novel and more elegant solution for proxy-based security.

A. Scenario

Proxy-based security for SIP is especially useful for a set of SIP user agents fulfilling a certain role (e.g., banking support or software support). Current SIP networks do not provide transparent security for such a scenario.

Figure 2 shows the standard case for establishing a call with a validated user agent. There, a standard `INVITE` message is sent from the initiating user agent (*Caller*) to the target user agent (*Agent 2*). Then, this user agent sends a digitally signed `200 OK` response using the user agent's certificate.

However, this approach has some drawbacks. In this scenario, target user agents (e.g., *Agent 2*) use their own certificate and are therefore visible to the calling user agent, which is something, call centres try to avoid (call centres have a general phone/SIP address, calls are forwarded to the actual locations). A solution to this would be the usage of a call-centre certificate for digitally signing SIP messages. This entails efforts to install the call-centre certificate on every call-centre agent's machine. Furthermore, call-centre user agents have to be changed in such manner, that the call-centre's SIP URI is returned in the agent's SIP messages. Thus, such an approach is not an elegant solution.

In the next section, we present our approach using enhanced proxy servers that are able to offer proxy-based security without any need to change the involved user agents.

B. Solution

In contrast to the standard SIP scenario, our concept of proxy-based security takes advantage of already existent SIP proxy servers. These proxy servers are responsible for forwarding SIP messages to the target user agent location.

As shown in Figure 3, we propose an extension of the standard SIP proxy server behaviour to provide a certification

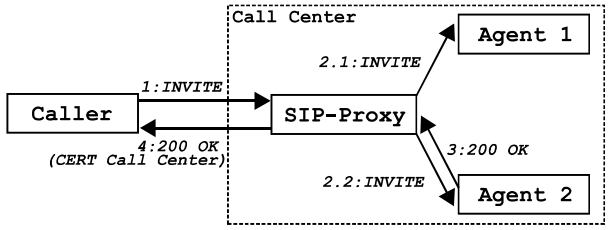

Fig. 3. Caller establishing call with call-centre-validated agent

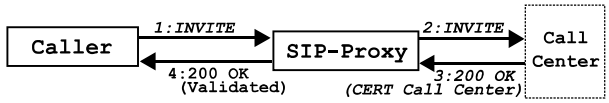

Fig. 4. Automatic proxy validation for call with call-centre-validated agent

and encryption service. Thus, the proxy server should be able to sign and encrypt messages sent from known user agents within the proxy's domain, e.g., from Agent 2 to the Caller in Figure 3. This results in fewer administration efforts at the affected user agents. Furthermore, these user agents do not have to use their own certificates, which results in more anonymity of the user agents, which is, e.g., demanded by customer call centres.

However, there is still a security problem. When allowing every known user-agent to use the certification and encryption service, there might be malicious user agents pretending to be one of these known user agents. To ensure the identity of these user agents, the proxy should provide an authentication service. We propose using standard SIP proxy authentication according to the SIP standard [2]. There, the proxy returns a `407 Proxy Authentication Required` message, which has to be answered by correct credentials. S/MIME authentication could be used as well, which would entail administrative efforts at the user agents for certificate management. However, this would enable to either use the user agent's certificate or the proxy's certificate (according to the user agent's requirements to anonymity which can be expressed by adding special header information, e.g., by adding a parameter *substitute_by_proxy_cert* to the Content-Type header).

Such an enhanced proxy server can also be used for validation and decryption of received messages (cf. Figure 4). If the proxy server validates the SIP message, it is able to mark the forwarded response, e.g., by adding an additional subject-header field containing 'validated <SIP-URI>', with the responding SIP user agent's SIP-URI. This feature is especially useful for pervasive, resource-limited devices, which can avoid local decryption and validation costs using this mechanism. However, as there is still communication between the decrypting or validating proxy and the user agent, this communication has to be secured, e.g., by building the communication on Transport Layer Security (TLS) [15]. Otherwise, the communication might be intercepted and the message can be marked spuriously valid.

```
1  public void processResponse (ResponseEvent e) {
2    Response rsp = e.getResponse();
3    CSeqHeader cseqHeader = (CSeqHeader) rsp.getHeader(
         CSeqHeader.NAME);
4
5    ...
6
7    if ( rsp.getStatusCode() == Response.OK &&
         cseqHeader.getMethod().equals("INVITE") ) {
8      HeaderSignature hs = new HeaderSignature(this,
           rsp);
9      rsp = hs.signMessage();
10   }
11
12   ...
13
14 }
```

Fig. 5. Code for intercepting message processing in the NIST SIP proxy server

C. Certificates and Distribution

As validation and decryption should be done automatically, we integrated a concept for transparent appliance of certificates. Therefore, we use S/MIME certificates for signing and encrypting messages according to the SIP standard. These certificates for a specific SIP URI are automatically created at the proxy if not existent. Then, the proxy is able to sign or to encrypt the response message and to attach the certificate. The receiving user agent and the decrypting/validating proxy, respectively, are able to extract the certificate and use it for validation or decryption of the message.

For proving the authenticity of the certificate this certificate has to be signed by a trusted certificate authority (CA). We support this by allowing a CA to certify our proxy-generated certificates.

Otherwise, it is also possible to use a certificate management service, as introduced by Jennings and Peterson [10]. Such a service enables discovering and installing certificates within a SIP network on demand.

V. Exemplary Integration of Proxy-based Security in the NIST SIP Proxy

In this section, we demonstrate the simplicity of our approach of integrating proxy-based security into a SIP network. Therefore, we integrated our concept into the NIST SIP proxy server. This introduced only few lines of code for intercepting and processing SIP messages.

Figure 5 shows an exemplary integration of our mechanism for signing responses to an INVITE message. Thus, it enables to validate the authenticity of the called user agent, e.g., a customer call-centre agent.

In JAIN–SIP-based applications like the NIST SIP proxy server, an event-based communication is used between the SIP stack and the application. Therefore, the application has to implement a `SIPListener` interface that provides, amongst others, methods for processing requests and responses (`processRequest()`, `processResponse()`). These methods enable the developer to handle requests and responses, respectively, and to create messages that are sent in reaction by the local SIP stack. For integration of proxy-based

security, we provide certain classes for signing/validating and encrypting/decrypting SIP messages. Figure 5 shows the listener method for processing incoming responses. Within this method, there is first checked, which kind of response is received (line 7). If the response is a reaction to an INVITE request, the message is signed using our provided class `HeaderSignature` (line 8-9). Thus, these lines implement the transparent authentication of a callee using S/MIME signed SIP messages.

If there is no existing certificate for the SIP address, our class is able to create a new certificate automatically. This certificate is exported as a text-document in PKCS#12 Format [16] to the local file system that can be signed by a trusted certificate authority (CA). For signing/validating and encrypting/decrypting, our class builds on the Java Cryptography API (JCA) [17]. As implementation of the JCA, we use the cryptographic service provider Bouncycastle [18]. In case of any failure, an unchanged response message object is returned. Thus, failure in signing/encrypting SIP messages does not result in a proxy server error. In contrast, it results in sending a plain SIP message only, which at least enables standard proxy server behaviour in case of failures.

VI. Conclusion and Future Work

In this paper, we presented a novel approach of transparently securing SIP-based networks by extending the standard behaviour of SIP proxies. This results in fewer administrative efforts as our proposed SIP proxy server is able to handle certificate management. Moreover, proxy-based security allows handling certificates for a whole group of users represented by only one SIP URI. This is especially useful for securing customer call-centres, in which user agents should only be contactable using a specific SIP URI in contrast to their actual SIP address. Our approach allows transparent authentication for callees in such a scenario.

We build on the SIP-proposed standard S/MIME for signing/validating and encrypting/decryption SIP messages. This enables standard SIP user agents to transparently interoperate with our proxy server without any needed modifications.

Furthermore, our extended proxy server can be used for client-side transparent validation and decryption of SIP messages as well. Especially resource-limited devices benefit from such architecture, as cost-extensive tasks of validation and decryption are done within the network instead of the local machine (e.g., PDA or smart phone).

Our approach can be easily integrated into standard SIP proxy servers, as we showed by an integration of our concept into the NIST SIP proxy server.

For future work, we think of extending the behaviour of our proxy server for filtering SPAM over Internet Telephony (SPIT). Particularly, this concept can be used for simple integration of certificates without further user interaction. Thus, it could enable a wider spread of certificate usage and therefore could be helpful for preventing SPIT by ensuring the identities of the participating entities [19]–[21].

Acknowledgment

The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

[1] PGP Corporation. PGP Universal. http://www.pgp.com/products/universal/index.html, 2006.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. IETF RFC 3261, 2002.

[3] J. Rosenberg and H. Schulzrinne. Session Initiation Protocol (SIP): Locating SIP Servers. IETF RFC 3263, 2002.

[4] National Institute of Standards and Technology (NIST). Project IP telephony / VoIP. http://snad.ncsl.nist.gov/proj/iptel/, 2005.

[5] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). IETF RFC 3711, 2004.

[6] J. Peterson. Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format. IETF RFC 3893, 2004.

[7] R. Sparks. Internet Media Type message/sipfrag. IETF RFC 3420, 2002.

[8] International Organization for Standardization and International Telecommunications Union. "information technology - open systems interconnection - the directory: Public-key and attribute certificate frameworks". "iso standard 9594-8:2001, itu-t recommendation x.509", 2000.

[9] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI Certificate Theory. IETF RFC 2693, 1999.

[10] C. Jennings and J. Peterson. Certificate Management Service for the Session Initiation Protocol (SIP). IETF Internet-Draft: draft-ietf-sipping-certs-02, 2005.

[11] Zertificon Solutions. Z1 SecureMail Gateway. http://www.bonelabs.com/secure_mail_gateway.php, 2006.

[12] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. DomainKeys Identified Mail (DKIM) Signatures. IETF RFC 4871, 2007.

[13] Inc. Anonymizer. "anonymizer - internet privacy & security solutions". http://www.anonymizer.com/, 2006.

[14] JAP Team. "jap - anonymity and privacy". http://anon.inf.tu-dresden.de/, 2006.

[15] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. IETF RFC 4346, 2006.

[16] RSA Laboratories. PKCS 12 v1.0: Personal Information Exchange Syntax. Public-Key Cryptography Standards, 1999.

[17] Inc. Sun Microsystems. Java Cryptography Architecture - API Specification & Reference. http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html, 2002.

[18] Bouncycastle.org. The Legion of the Bouncy Castle. http://www.bouncycastle.org, 2006.

[19] G. Dawirs, T. Froment, and H. Tschofenig. Authorization Policies for Preventing SPIT. IETF Internet-Draft: draft-froment-sipping-spit-authz-policies-01.txt, 2006.

[20] D. Schwartz, B. Sterman, E. Katz, and H. Tschofenig. SPAM for Internet Telephony (SPIT) Prevention using the Security Assertion Markup Language (SAML). IETF Internet-Draft: draft-schwartz-sipping-spit-saml-01.txt, 2006.

[21] S. Niccolini, S. Tartarelli, M. Stiemerling, and S. Srivastava. SIP Extensions for SPIT identification. IETF Internet-Draft: draft-niccolini-sipping-feedback-spit-02, 2006.