Security in End-to-end Communications

Kumiko Ono†, Shinya Tachimoto, and Seiichi Sakaya

Abstract

Ensuring security in end-to-end communications is an important issue that depends heavily on the capabilities of end terminals. This article introduces a user-to-user mutual authentication mechanism to reduce the load on the user by utilizing network functions. It also introduces an end-to-middle security mechanism to enable network services to work with end-to-end security for signaling.

1. Limits in achieving end-user-based security

To achieve secure communication, it is generally thought that security should be implemented by the capabilities of the end terminals as opposed to relying on the capabilities of intermediary servers (i.e., ones in the "middle"). However, this scheme might present problems in communication involving public services such as Internet telephony services. Focusing on this issue, we introduce a user-to-user mutual authentication mechanism and an end-to-middle security mechanism for solving problems that arise when attempting to achieve security with end terminals.

2. Standard specifications for signaling security and problems

Specifications for session initiation protocol (SIP) were standardized in June 2002. They include various security mechanisms (**Table 1**).

Digest authentication can be applied to user-to-user mutual authentication and user-to-server mutual authentication. It requires a pre-shared key such as a password, which makes it applicable to mutual authentication between specific users. However, it cannot be used for mutual authentication between arbitrary users as in public services. Transport layer security (TLS) and secure multipurpose Internet mail extensions (S/MIME) are examples of authentication mechanisms that use public key certificates (PKCs). Certification for public keys is widely used for authentication for web servers, but it has hardly been used for user authentication because of the high cost of issuing and managing PKCs.

In addition to authentication on each transport path, TLS can be applied to confidentiality and integrity protection hop-by-hop at the transport layer. A TLS connection can be requested over an entire transport path even if it passes through SIP servers by setting a secure SIP uniform resource identifier (SIPS URI) as a destination address when sending a request message. S/MIME can be used to provide authentication between users as well as end-to-end confidentiality and integrity. It can also be used to prevent repudiation. However, these mechanisms for providing signaling confidentiality can only be applied on each transport path and/or between users—they cannot be applied between a user and a non-adjacent SIP server. For example, if end-to-end encryption is performed for confidentiality, then services that utilize content information included in the session description protocol (SDP) of the signaling are disabled because this information cannot be viewed by a SIP server.

3. User-to-user mutual authentication mechanism

Here, we introduce a user-to-user mutual authentication mechanism that features the use of intermediary proxy servers on the network. This mechanism is based on server intervention. It makes use of "transi-

† NTT Network Service Systems Laboratories
Musashino-shi, 180-8585 Japan
E-mail: ono.kumiko@lab.ntt.co.jp

Table 1. Security mechanisms utilized by SIP standard specifications and associated problems.

Purpose	Security mechanism		Problems
Authentication	Digest authentication	Authentication based on a pre-shared key. Applicable to authenticating the source of a request between users and between a user and any SIP server	The need for a pre-shared key makes it difficult to apply between arbitrary users. Weak against key-reconstruction attacks.
	TLS	Authentication based on PKCs. Applicable to (TLS server) authentication on each transport path, hop-by-hop	User requires a PKC if acting as a TLS server. Assumes the diffusion of PKCs among users.
	S/MIME	Authentication based on signatures for use with PKCs. Applicable to authentication of the source of message creation between users, end-to-end	
Integrity (detection of tampering)	TLS	Applicable to each transport path, hop-by-hop	
	S/MIME	Applicable between users, end-to-end	
Confidentiality (prevention of eavesdropping)	TLS	Applicable to each transport path, hop-by-hop	Cannot be applied between a user and any SIP server.
	S/MIME	Applicable between users, end-to-end	User requires a PKC if acting as a TLS server. Assumes the diffusion of PKCs among users.

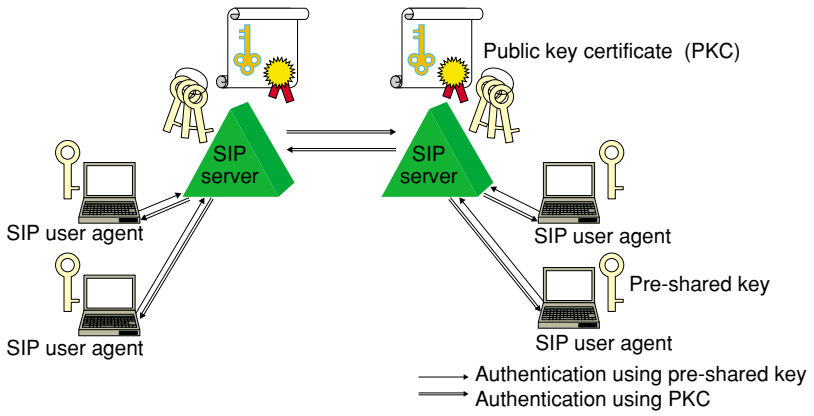

Fig. 1. Keys used in user-to-user mutual authentication.

tive trust" achieved through SIP-user-agent-to-SIP-server authentication and SIP-server-to-SIP-server authentication. Here, user authentication by a SIP server uses a pre-shared key, while SIP-server authentication by a user applies a PKC. In addition, SIP-server-to-SIP-server authentication uses a PKC (**Fig. 1**). Authentication by a PKC uses TLS and that by pre-shared keys uses digest authentication. By combining these authentication mechanisms, we eliminate the need for pre-shared keys between users or for user PKCs. Therefore, user-to-user mutual authentication becomes possible.

4. Extended SIPS URI

When server intervention is used as described above, how can a SIP message be authenticated between end users over an entire transport path? Here, we introduce a scheme for extending SIPS URI to enable a SIP request message to be transferred over TLS connections over an entire transport path.

In the authentication phase of TLS, a TLS server that receives a connection request must possess a PKC. If a TLS connection is to be made between a SIP server and a SIP user agent in the destination (a destination user), then, upon receiving the SIP request message, the destination user must take on the role of a TLS server, which means that it must have a PKC. To avoid this necessity for a PKC at the destination, the TLS connection established between the destination user and SIP server at the time of destination user location registration is reused upon receipt

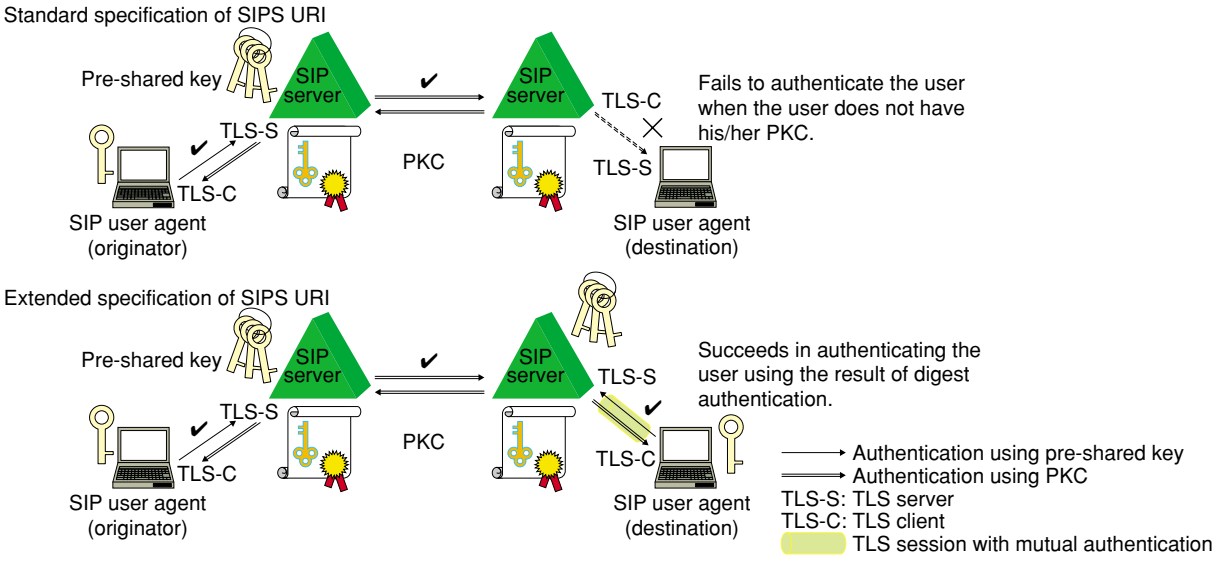

Fig. 2. User-to-user mutual authentication using extended SIPS URI.

of the SIP request message (**Fig. 2**). Because the user acts as a TLS client and the SIP server acts as a TLS server at the time of location registration, there is no need for the user to have a PKC. Specifically, user authentication is performed by digest authentication using the pre-shared key between the user and SIP server, with the result that a TLS connection is retained in a state of mutual authentication. In this way, a TLS connection between a SIP server and destination user becomes possible. Between SIP servers, mutual authentication is performed as part of the procedure for establishing a TLS connection. The presence or absence of mutual authentication can be managed as information corresponding to a TLS connection. As long as signaling continues to be transferred using that TLS connection, trust on the basis of mutual authentication can be verified.

The originating user can request authentication of a destination user by setting a SIPS URI as the destination of the SIP message request. The destination user, in turn, can register a SIPS URI as a destination address during the location registration phase, thereby limiting reception to messages from only authenticated users.

A SIPS URI therefore comes to be used for requesting not only TLS, but also mutual authentication in the form of a SIPS URI implementation specification. Instead of making a major extension to standard specifications, we limit the extension to connection with SIP servers corresponding to this implementation specification, with the result that user-to-user mutual authentication becomes relatively easy to implement. This user-to-user mutual authentication mechanism was proposed by NTT as an implementation agreement at the GMI2004 event of the Multi-service Switching Forum (MSF) and is currently in final adoption proceedings [1].

5. End-to-middle security

We introduce an end-to-middle security mechanism as a means of information-disclosure control that allows the encryption of sensitive information between a SIP user agent (end) and a SIP server (middle) to solve the following problem.

If the SDP included in signaling messages can be read, then the IP addresses and port numbers of media packets can be identified, making it easy to eavesdrop on media streams. It might therefore seem desirable to encrypt the SDP using S/MIME for end-to-end communications. However, various services such as call admission control based on the network bandwidth and firewall control provided by intermediary servers are based on SDP (**Fig. 3**). For these services to work, such information must be disclosed to those servers, which rules out encryption between users. For security reasons, however, it is not desirable to disclose the information to SIP servers that are not involved in service control. Therefore, to resolve this conflict between encrypting the information between users and enabling service control based on that information, we need a flexible mechanism that

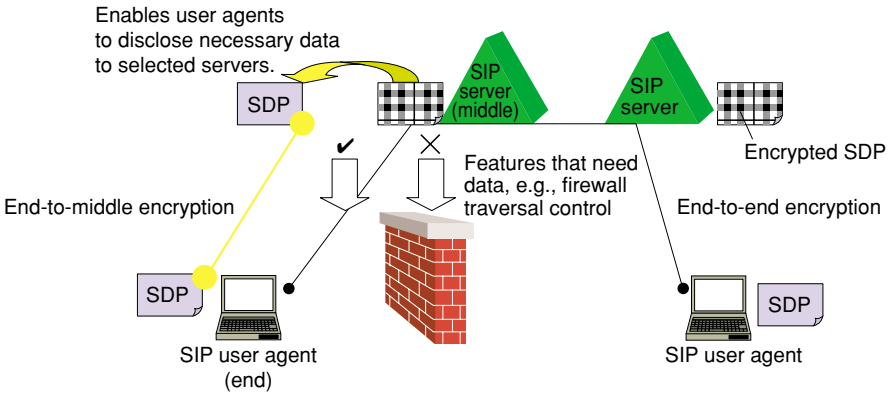

Fig. 3. End-to-middle security.

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
Route: <sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Date: Fri, 20 June 2003 13:02:03 GMT
Content-Type: application/pkcs7-mime;smime-type=enveloped-data;
              name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment;filename=smime.p7m;handling=required;
Content-Target:sip:ss1.atlanta.example.com <--Label for SIP Server #1
Content-Length: ...
                                          S/MIME Enveloped Data
*****************************************************************************
* (encryptedContentInfo) Media data encrypted with a sym-key, CEK1          *
* Content-Type: application/sdp                                             *
* Content-Length: ...                                                       *
*                                                                           *
* v=0                                                                       *
* o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com           *
* s=-                                                                       *
* c=IN IP4 192.0.2.101                                                      *
* t=0 0                                                                     *
* m=audio 49172 RTP/AVP 0                                                   *
* a=rtpmap:0 PCMU/8000                                                      *
*                                                                           *
* (recipientInfos)                                                          *
* RecepientInfo[0]  CEK1 encrypted with the public key of SIP server #1     *
* RecepientInfo[1]  CEK1 encrypted with the public key of a destination user*
*                                                                           *
* (unprotectedAttr)                                                         *
* CEKReference                                                              *
*****************************************************************************
```

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e418c4.1
;received=192.0.2.222
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=192.0.2.111
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=192.0.2.101
Record-Route: <sip:ss2.biloxi.example.com;lr>,
<sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=314159
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Content-Type: application/pkcs7-mime;
              smime-type=enveloped-data;name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment;filename=smime.p7m;handling=required;
Content-Target:sip:ss1.atlanta.example.com <--Label for SIP Server #1
Content-Length: ...
                                                    S/MIME Enveloped Data
*****************************************************************************
* (encryptedContentInfo) Media data encrypted with a sym-key, CEK2          *
* Content-Type: application/sdp                                             *
* Content-Length: ...                                                       *
*                                                                           *
* v=0                                                                       *
* o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com              *
* s=-                                                                       *
* c=IN IP4 192.0.2.201                                                      *
* t=0 0                                                                     *
* m=audio 3456 RTP/AVP 0                                                    *
* a=rtpmap:0 PCMU/8000                                                      *
*                                                                           *
* (recipientInfos)                                                          *
* RecepientInfo[0] CEK2 encrypted with CEK1                                 *
*****************************************************************************
```

Fig. 4. Example of a message applying the key reuse method.

can disclose only necessary information to selected servers. This mechanism consists of the following methods: encryption, information labeling, key reuse, and discovery of the middle.

- **Encryption**

Encryption can accommodate multiple recipients. We apply S/MIME as prescribed by standard specifications because S/MIME, as a security mechanism for e-mail, makes it easy to disclose the same information to multiple recipients such as selected servers and the destination user. A SIP server can perform SDP-based service control by decrypting the SDP. Signature verification can be used to check whether the SDP has been tampered with.

- **Information labeling**

To make the referencing of target information more efficient, a user agent labels the target information in order to indicate the disclosure destination. The label also needs a signature to ensure secure transport of the label itself. To simplify the message format when attaching such a signature, we use a MIME header to label this information.

- **Key reuse**

When the SDP is encrypted, the reuse of encryption keys can make the encryption and decryption of the SDP within multiple SIP messages more efficient. For example, to perform firewall control by an intermediary SIP server in the network to which the originating user is connected, the SDP in response messages from the destination user must be inspected by the SIP server. One option for doing this would be for the originating user to send the PKC of the SIP serv-

er to the destination user, but a key reuse mechanism is even simpler.

When the SDP in messages is encrypted, the key used for that encryption (content-encryption key (CEK)) must itself be encrypted by the public keys (key-encryption keys) of the destination user and SIP server. However, if the key reuse mechanism is used, the key used for encrypting content information in a response message, for example, could be encrypted by the same key as used for encrypting the request message (**Fig. 4**). Thus, key encryption could be performed by a symmetric encryption algorithm instead of by an asymmetric encryption algorithm, making the encryption process more efficient.

- **Discovery of the middle**

A method for finding a SIP server to perform service control may also be needed depending on the application format. If a SIP server cannot reference the information needed for service control, an error reply is returned to the sending user together with that server's own public key. In this way, a SIP server that can perform service control for the destination user can be detected by the sending user even when content information within the signaling transmitted by the sending user must be referenced.

This end-to-middle security mechanism is now undergoing standardization in IETF (the Internet Engineering Task Force) based on a proposal submitted by NTT [3].

6. Future outlook

While this article has discussed security with regard to signaling, security for media streams is equally important. Secure RTP (SRTP) [4] has recently been standardized as a security mechanism for RTP/RTCP that be used for audio and video communications. (RTP: realtime transport protocol, SRTP: secure realtime transport protocol, RTCP: RTP control protocol) This mechanism efficiently detects tampering with media streams and prevents eavesdropping on them. Standardization of SDP extended specifications to enable the exchange of key parameters for this SRTP by SIP signaling is also in progress [5].

In media streams, mutual authentication between users can be roughly achieved by coordinating signaling with dynamic port-number assignment. Firewall control based on port filtering is also being achieved through coordination with signaling, but this requires that a SIP proxy server be co-located with the firewall or that the firewall be dynamically controlled from a SIP proxy server. To eliminate these configuration constraints, we are studying an authentication token mechanism in which the firewall inspects the authentication token attached to RTCP to enable/disable port opening and closing. In future research, we plan to examine the feasibility of this authentication token mechanism.

References

[1] http://www.msforum.org/

[2] K. Ono and S. Tachimoto, "End-to-Middle Security in Session Initiation Protocol," FIT2003, 2003 (in Japanese).

[3] K. Ono and S. Tachimoto, "Requirements for End-to-Middle Security for the SIP," draft-ietf-sipping-e2m-sec-reqs-03, July 2004.

[4] M. Baugher, *et al*., "The Secure Real-time Transport Protocol (SRTP)," IETF RFC3711, Mar. 2004.

[5] F. Andreasen, *et al*., "Session Description Protocol Security Descriptions for Media Streams," draft-ietf-mmusic-sdescriptions-06.txt, July 2004.

Kumiko Ono

Research Engineer, Network Software Service Project, NTT Network Service Systems Laboratories.

She received the B.S. degree in mathematics from Ochanomizu University, Tokyo in 1992. Since joining NTT in 1992, she has been engaged in R&D of realtime communications, especially voice over IP. She participates in standards bodies such as IETF and MSF. She is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Shinya Tachimoto

Senior Research Engineer, Supervisor, Network Software Service Project, NTT Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in mechanical engineering from Tokyo Institute of Technology, Tokyo in 1988 and 1990, respectively. He joined NTT in 1990. He is currently researching session management. His other research interests include the next-generation network architecture, secure end-to-end communications, and high-availability middleware for reliable node systems. He is a member of IEEE.

Seiichi Sakaya

Researcher, Network Software Service Project, NTT Network Service Systems Laboratories.

He received the M.S. degree in physics from Tokyo University of Science, Tokyo in 2003 and joined NTT the same year. He is currently researching session management. His other research interests include the next-generation network architecture. He is a member of IEICE.