Computers & Security (2005) 24, 381—386

Computers & Security

www.elsevier.com/locate/cose

Secure authentication scheme for session initiation protocol☆

Chou-Chen Yang[a,*], Ren-Chiun Wang[b], Wei-Ting Liu[c]

[a]*Department of Management Information System, National Chung Hsing University, 250, Kuo Kuang Rd., Taichung, 413 Taiwan, ROC*
[b]*Department of Information Management, Chaoyang University of Technology, 168, Gifeng E. Rd., Wufeng, Taichung County, 413 Taiwan, ROC*
[c]*Graduate Institute of Networking and Communication Engineering, Chaoyang University of Technology, 168, Gifeng E. Rd., Wufeng, Taichung County, 413 Taiwan, ROC*

Received 19 March 2004; revised 6 September 2004; accepted 15 October 2004

KEYWORDS
Session initiation protocol;
Authentication;
Security;
HTTP digest authentication;
Key agreement

Abstract The Session Initiation Protocol provides an expandable and easy solution to the IP-based telephony environment. When users ask to use an SIP service, they need to be authenticated in order to get service from the server. Therefore, some SIP authentication procedure schemes were proposed to meet the above demand. However, there are security problems that need to be solved, such as off-line password guessing attacks and server spoofing. In this article, we shall propose a new scheme for a secure authentication procedure for the Session Initiation Protocol to enhance the security of the original scheme.

Introduction

The Internet Engineering Task Force (IETF) proposed the Session Initiation Protocol (SIP) (Rosenberg, 2002; Handley, 1999) as the IP-based telephony protocol. SIP is a call setup signaling protocol for IP-based telephony services, which means it establishes, maintains, and terminates user sessions. SIP is based on the application-layer and is a text-based client—server protocol. The SIP architecture is mainly composed of a proxy server, redirect server, user agent, register server, and location server. The function of each component is described as follows.

- Proxy server:
  A proxy server forwards a request and response between a callee and caller. When the proxy server receives a request, it forwards the

☆ This research was partially supported by the National Science Council, Taiwan, ROC, under contract no. NSC92-2213-E-324-005.
* Corresponding author.
E-mail addresses: cc.yang@nchu.edu.tw (C.-C. Yang), s9114624@mail.cyut.edu.tw (R.-C. Wang), s9230602@mail.cyut.edu.tw (W.-T. Liu).

0167-4048/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.cose.2004.10.007

request to the current location of the callee, and then forwards the response from the callee to caller.

- Redirect server:
  When a redirect server receives a request, it informs the caller about the current location of the callee. Then the caller contacts the callee directly.
- User agent:
  A user agent is a logical entity, such as a callee or a caller.
- Register server:
  When a user agent changes its location, the user agent sends a register request to the register server to update its current location. In brief, the register server helps the user agent update the information of the user agent's location in the location server.
- Location server:
  The responsibility of the location server is to maintain information on the current location of the user agent. It also services the proxy server, redirect server, and register server for them to look up or register the location of the user agent.

Currently, the security of SIP is becoming more important (Arkko, 2002; Veltri et al., 2002; Thomas, 2001). When a user requests to use an SIP service, he needs to be authenticated first before getting the service from the server. For the above reasons, the procedure of the SIP authentication scheme was proposed in Rosenberg (2002), Handley (1999) and Veltri et al. (2002) to meet the above demand. In the proposed scheme, the server uses a challenge—response mechanism to verify the identity of the user. The SIP authentication scheme described in Rosenberg (2002) is derived from HTTP digest authentication (Franks et al., 1999). Franks et al. described the HTTP digest authentication scheme and pointed out that it is better than nothing. Therefore, the HTTP digest authentication scheme may not be safe enough for the SIP service. For example, if the user does not verify the identity of the server, an attacker can forge the identity of the server to obtain some secret information of the user. Furthermore, the SIP authentication scheme has some security problems such as off-line password guessing attack and server spoofing. In this article, we shall point out that the procedure of the original SIP authentication scheme is insecure. At the same time, we shall propose a secure Session Initiation Protocol authentication procedure to enhance the security of the original scheme.

This article is organized as follows: next section presents a review of the SIP authentication procedure. Then, the weakness of the SIP authentication procedure will be described which is followed by the details of the proposed scheme. Further, we shall analyze the security of our scheme. Finally, last section presents our conclusion.

SIP authentication procedure

SIP authentication security is based on the challenge—response mechanism. Before the scheme starts, the client pre-shares a password with the server. Note that the pre-share password is used to verify the identity of the client or the server because only these two sides have the pre-share password. In this case, the original SIP authentication scheme only verifies the identity of the client (Fig. 1).

Step 1. *client* → *server*: REQUEST
The client sends a REQUEST to the server.

Step 2. *server* → *client*:
CHALLENGE(*nonce*, *realm*)
The server generates a CHALLENGE that includes a *nonce* and the client's *realm*. Note that the *realm* is used to prompt the *username* and *password*. Then the server sends a CHALLENGE back.

Step 3. *client* → *server*:
RESPONSE(*nonce*, *realm*, *username*, *response*)
The client computes a *response* = F(*nonce*, *username*, *password*, *realm*). Note that $F(\cdot)$ is a one-way hash function and is used to generate a digest authentication message. Then the client sends the RESPONSE to the server.

Step 4.
According to the *username*, the server extracts the client's *password*. Then the server verifies whether the *nonce* is correct or not. If it is correct, the server computes F(*nonce*, *username*, *password*, *realm*) and uses it to compare it with the *response*. If they match, the server authenticates the identity of the client.

Weakness of the SIP authentication procedure

The procedure of the SIP authentication scheme given above has the following problems.

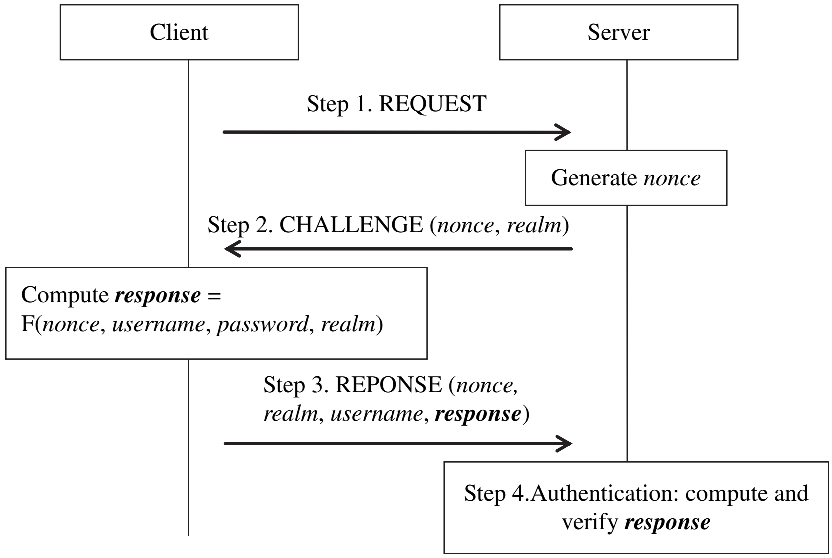

Figure 1 SIP authentication procedure.

Off-line password guessing attack

In steps 2 and 3, an attacker can easily obtain the *nonce, username, realm*, and *response*. Then the attacker guesses the password *pw′* and computes *F*(*nonce, username, pw′, realm*). If the computed result is equal to the *response*, he gets the correct password.

Server spoofing

If the user does not verify whether the identity of the server is correct or not, he always sends back an honest response after he gets the CHALLENGE. An attacker can forge the identity of the server and make a CHALLENGE to obtain the RESPONSE. After the attacker receives the RESPONSE, he can make an **off-line password guessing attack** to get the correct password of the user. Each step is described as follows.

Step 1. *client* → *attacker*: REQUEST
The client sends a REQUEST to the attacker.

Step 2. *attacker* → *client*:
CHALLENGE(*nonce′*, *realm*)
The attacker generates a CHALLENGE that includes a *nonce′* and the client's *realm*. Then the attacker sends the CHALLENGE back.

Step 3. *client* → *server*:
RESPONSE(*nonce′*, *realm*, *username*, *response*)
The client computes a *response* = *F*(*nonce′*, *username, password, realm*), and sends the RESPONSE to the attacker.

Step 4.
After the attacker receives the RESPONSE, he gets (*nonce′, realm, username, response*). Therefore, the attacker can make an **off-line password guessing attack** on the RESPONSE to get the correct password of the user.

Secure session initiation protocol authentication procedure

For the above weaknesses, we propose a new scheme to enhance the security of the original SIP authentication procedure. Our scheme is based on the Diffe—Hellman concept (Diffe and Hellman, 1976), which depends on the difficulty of discrete logarithms. First, the server issues a large prime p and a generator g. In addition, the client pre-shares a password *pw* with the server. Note that the pre-share password is used to verify the identity of the client or the server because only these two sides have the pre-share password. When a user requests to access the resource of the server, he proceeds with the following steps (Fig. 2).

Step 1. *client* → *server*:
REQUEST{*username*, $t_1 \oplus F(pw)$}
The user chooses a random number r_1 and computes $t_1 = g^{r_1} \bmod p$. Then the user sends a REQUEST to the server that includes the username and $t_1 \oplus F(pw)$. Note that $F(\cdot)$ is a one-way hash function and $\oplus$ denotes the XOR operation. The user needs to keep r_1 for the future step 3, after that he can discard r_1.

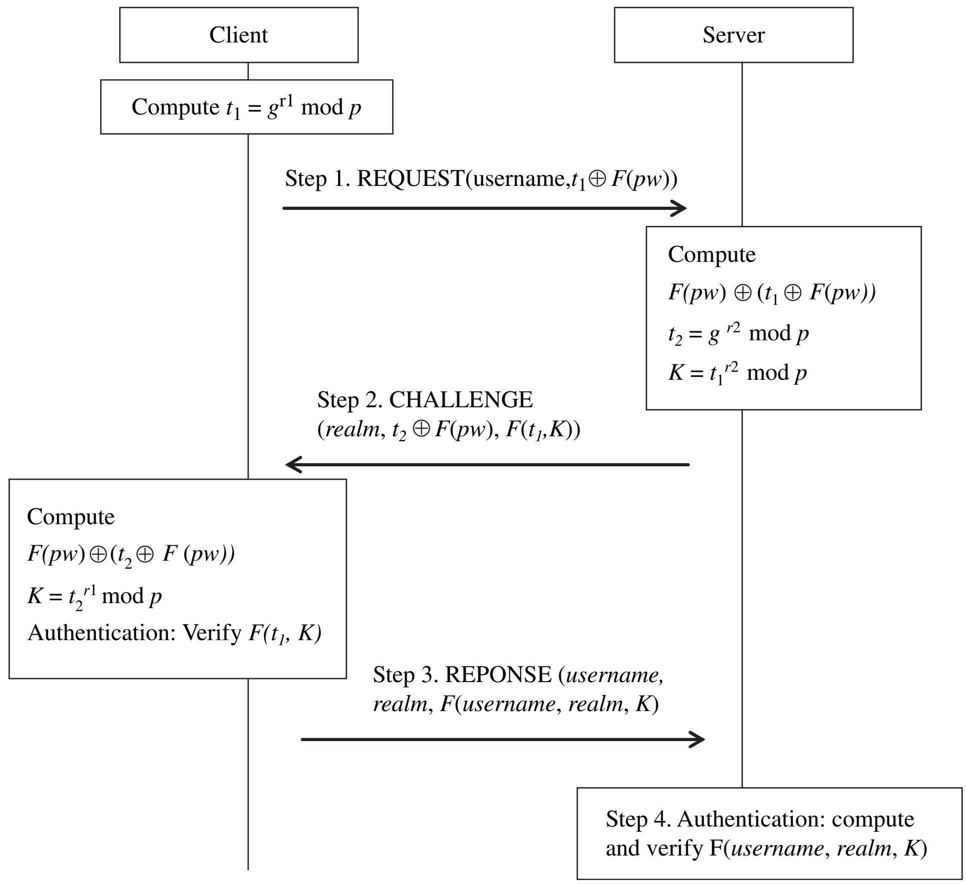

Figure 2 Secure SIP authentication procedure.

Step 2. *server* → *client*:

CHALLENGE{$realm$, $t_2 \oplus F(pw)$, $F(t_1, K)$}

After the REQUEST is received, the server uses the *username* to obtain the user's password. Furthermore, the server computes *F(pw)* to obtain t_1 by computing $F(pw) \oplus (t_1 \oplus F(pw))$. Then the server chooses a random number r_2 and computes $t_2 = g^{r_2} \bmod p$, $K = t_1^{r_2} \bmod p$ and $F(t_1, K)$. Finally, the server sends the CHALLENGE that includes *realm*, $t_2 \oplus F(pw)$, and $F(t_1, K)$ to the user.

Step 3. *client* → *server*:

RESPONSE{*username*, *realm*, *F*(*username*, *realm*, *K*)}

The user uses $F(pw)$ to get t_2 and to compute $K = t_2^{r_1} \bmod p$ and $F(t_1, K)$. If $F(t_1, K)$ is true, the user authenticates the identity of the server. Meanwhile, the user sends the RESPONSE, which includes the *username*, *realm*, and *F*(*username*, *realm*, *K*) to the server.

Step 4.

After the RESPONSE is received, the server computes *F*(*username*, *realm*, *K*). If the computed value is the same as the RESPONSE, the server authenticates the identity of the user.

Security analysis and discussion

In this section, we analyze the security of our scheme as follows.

Replay attack

If an attacker replays {*username*, $t_1 \oplus F(pw)$} to the server, the server will send {*realm*, $t_2 \oplus F(pw)$, $F(t_1, K)$} back. Because the attacker has no $F(pw)$, he cannot send the RESPONSE to the server in step 3. Therefore, replay attacks cannot work in our scheme.

Off-line password guessing attack

If an attacker intercepts the messages from steps 1, 2 and 3, the attacker cannot make an **off-line**

password guessing attack on these messages. The reasons are described as follows.

1. The attacker guesses a password pw' and computes $F(pw')$.
2. Then the attacker computes $t_1' = F(pw') \oplus (t_1 \oplus F(pw))$ and $t_2' = F(pw') \oplus (t_2 \oplus F(pw))$.
3. Obviously, the attacker cannot compute the value K to match the RESPONSE, because he faces the difficulty of discrete logarithms. Therefore, the **off-line password guessing attack** cannot work in our scheme.

Server spoofing

In step 2 of our scheme, the server computes $F(pw)$ to obtain t_1 by computing $F(pw) \oplus (t_1 \oplus F(pw))$. Then the server computes $K = t_1^{r_2} \bmod p$ and sends $F(t_1, K)$ to the user. The user can verify the identity of the server by computing $F(pw) \oplus (t_2 \oplus F(pw))$, $K = t_2^{r_1} \bmod p$ and verifying $F(t_1, K)$. Obviously, the attacker cannot impersonate the server to deceive the user.

Tables 1 and 2 show the security analysis and comparisons among the HTTP digest scheme (Franks et al., 1999), EKE (Encrypted Key Exchange) scheme (Bellovin and Merritt, 1992; Bellovin and Merritt, 1993) and ours. The EKE scheme uses the Diffe–Hellman concept for authentication. As shown in Table 1, the HTTP digest authentication scheme is susceptible to the **off-line password guessing attack** and **server spoofing**. Therefore, the SIP authentication procedure based on HTTP digest authentication is not safe enough. On the other hand, our proposed scheme and the EKE scheme can resist the same attacks. But, as shown in Table 2, the EKE scheme uses 9 symmetric encryption operations which need more computation time than our scheme only using 7 one-way hash function and 4 XOR operations.

Furthermore, the EKE scheme generally needs four message flows to reach the authentication scheme. Our scheme only needs three message flows. Therefore, our scheme is efficient and secure for the SIP authentication.

Table 1 Security analysis

	HTTP digest	EKE	Ours
Replay attack	No	No	No
Off-line password guessing attack	Yes	No	No
Server spoofing	Yes	No	No

Table 2 Comparisons

	HTTP digest	EKE	Ours
One-way hash function	1	No	7
Exponentiation	No	4	4
Symmetric encryption	No	9	No
Exclusive or	No	No	4
The number of message flows	1.5	2	1.5

Conclusions

In this article, we described the original SIP authentication procedure based on HTTP digest authentication. We pointed out that the procedure of the original SIP authentication scheme is vulnerable to the off-line password guessing and server spoofing attacks. Thus, we proposed a secure authentication scheme for the Session Initiation Protocol to resist the above attacks.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments. This research was partially supported by the National Science Council, Taiwan, ROC, under contract no. NSC92-2213-E-324-005.

References

Arkko J, et al. Security mechanism agreement for SIP sessions. IETF Internet draft (draft-ietf-sip-sec-agree-04.txt); June 2002.

Bellovin SM, Merritt M. Encrypted key exchange: password-based protocols secure against dictionary attacks. IEEE symposium on research in security and privacy 1992. p. 72–84.

Bellovin SM, Merritt M. Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise. Proceedings of the 1st ACM conference on computer and communications security November 1993. p. 244–50.

Diffe Whitfield, Hellman M. New directions in cryptology. IIEEE Transaction on Information Theory 1976;IT-22(6):644–54.

Franks J, et al. HTTP authentication: basic and digest access authentication. IETF RFC2617; June 1999.

Handley M, et al. SIP: session initiation protocol. IETF RFC2543; March 1999.

Rosenberg J, et al. SIP: session initiation protocol. IETF RFC3261; June 2002.
Thomas M. SIP security requirements. IETF Internet draft (draft-thomas-sip-sec-reg-00.txt); November 2001 (work in progress).
Veltri L, Salsano S, Papalilo D. SIP security issues: the SIP authentication procedure and its processing load. IEEE Network 2002;16(6):38—44.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University in 1980. He received his M.S. in Electronic Technology from the Pittsburg State University in 1986, and his Ph.D. in Computer Science from the University of North Texas in 1994. He has been an associate professor in the Department of Management Information System at National Chung Hsing University. His current research interests include network security, mobile computing, and distributed system.

Ren-Chiun Wang received the B.S. in Information Management from Ming Chuan University in 2002, and got his M.S. in Information Management from Chaoyang University of Technology in 2004. His current research interests include information security and mobile communications.

Wei-Ting Liu received the B.S. in Computer Science and Information from Chaoyang University of Technology in 2003. He is pursuing his M.S. in Networking and Communication from Chaoyang University of Technology. His current research interests include network security and mobile communications.