Efficient Nonce-based Authentication Scheme for Session Initiation Protocol

Jia Lun Tsai
Degree Program for E-learning, Department of Applied Mathematics, National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC (Email:crousekimo@yahoo.com.tw)
(Received June 14, 2007; revised Oct. 23, 2007; and accepted Jan. 21, 2008)

Abstract

In recent years, Session Initiation Protocol (SIP) is more and more popular. However, there are many security problems in the Session Initiation Protocol. In 2005, Yang et al. [9] proposed a secure authentication scheme for Session Initiation Protocol. This authentication scheme is based on Diffie-Hellman [2] concept, so the computation cost of this authentication scheme is very high. In order to improve this shortcoming, Durlanik et al. [3] also proposed an authentication Scheme using ECDH in 2005. However, the computation cost of this authentication scheme is still very high. In this paper, we propose an efficient nonce-based authentication scheme. The computation cost of this authentication scheme is lower than Yang et al.s authentication scheme and Durlanik et al.s authentication scheme, and it is very suitable for low computation power equipment.

Keywords: Authentication scheme, HTTP digest authentication, session initiation protocol

1 Introduction

Session Initiation Protocol (SIP) [1, 4, 7] is the Internet Engineering Task Force (IETF) standard for IP telephone. The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol that can create, modify and terminate sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these. Session Initiation Protocol is a text-based peer-to-peer protocol, and incorporates elements of two widely used Internet protocols: Hyper Text Transport Protocol (HTTP) used for Web and Simple Mail Transport Protocol (SMTP) [8]. Authentication is the most important for Session Initiation protocol. When a user wants to use Session Initiation Protocol, this user must be authenticated. However, other than needing to recognize whether a user's identity is legal, the user will also need to recognize whether the server is the correct one as well. For these reasons, the authentication scheme of SIP is not safe, because the SIP authentication scheme is derived from HTTP Digest authentication [4]. The authentication scheme of SIP uses challenge-response mechanism to only verify the identity of the user and is vulnerable to off-line password guessing attack, server spoofing. In order to resist these flaws, Yang et al. [7, 9] proposed a secure authentication scheme for session initiation protocol. This authentication scheme is based on Diffie-Hellman Key Exchange [2], which depends on the difficulty of discrete logarithms. The computation cost of the Yang et al.'s authentication scheme is very high, so it is not suitable for low computation power equipments. In order to improve this shortcoming, Durlanik et al. [3] also proposed an authentication Scheme using ECDH in 2005. In this paper, we propose an efficient nonce-based authentication scheme. This authentication scheme is based on the random nonce. All communication messages are encrypted/ decrypted by using one-way hash function and exclusive-or operation, so its computation cost is low. It is very suitable for low computation equipment.

2 Review Yang et al.'s Scheme and Durlanik et al.'s Scheme

In this section, we reviewed Yang et al.'s authentication scheme [9] and Durlanik et al.'s authentication scheme [3]. Before we review these two proposed authentication schemes for Session Initiation Protocol, all symbols are described as Table 1.

2.1 Review Yang et al.'s Authentication Scheme

In this section, we review Yang et al.'s authentication scheme. This authentication scheme is based on Diffie-Hellman Key Exchange [2] and enhances the security of the original SIP authentication scheme.

Table 1: Symbol table

Symbol	Explanation
$h()$	One-way hash function
$\oplus$	Exclusive or
PW	The remote user password
$\|\|$	Concatenation
N	The random nonce generated by the user and the server
$X \longrightarrow Y : M$	X send a message M to Y
U, S	Each represents as user and server
p	Large prime number
dc, ds	Represents the public key and private key
G	Base point in the curve

2.1.1 Registration Phase

When a user wants to register and become a new legal user, the person must first submit his or her username and password to remote server. The username and password is used to verify the identity of the user and server. When the server receives this user's username and password, the server stores this username and password.

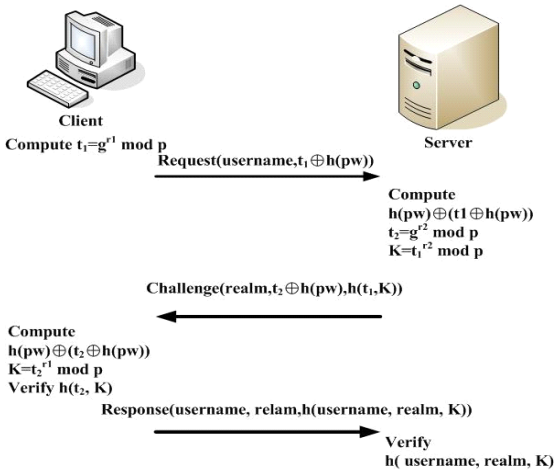

Figure 1: Yang et al.'s authentication scheme

2.1.2 Authentication Phase

If a legal user wants to login in system, this user must type his or her username and password. All steps of authentication phase execute as following.

Step 1: $U \longrightarrow S$: $Request(username, t_1 \oplus h(pw))$, where $t_1 = g^{r_1} \bmod p$.

The user generates a random number r_1, and use r_1, g, p to compute $t_1 = g^{r_1} \bmod p$.

Step 2: $S \longrightarrow U$: $Challenge(realm, t_2 \oplus h(pw), h(t_2, K))$.

When the server receives the user's messages, the server uses username to get the user's pw from its database. The server uses pw to compute $h(pw) \oplus t_1 \oplus h(pw)$ to get t_1. Furthermore, the server generate a random number r_2 and use t_1, r_2, g, p, pw to compute $t_2 \oplus h(pw), K = t_1^{r_2} \bmod p, and h(t_1, K)$. Then the server sends Challenge $(reakm, t_2 \oplus h(pw), h(t_1, K))$ to the user.

Step 3: $U \longrightarrow S$: $Response(username, realm, h(username, realm, K))$.

When the user receives the Challenge message, the user uses $h(pw)$ to compute $h(pw) \oplus t_2 \oplus h(pw)$ to get t_2. Then the user uses t_2, r_1, p to compute $K = t_2^{r_1} \bmod p$ and $h(t_1, K)$. If the computed $h(t_1, K)$ is not the same as the $Challenge(h(t_1, K))$, the user rejects the server request. Otherwise, the user use username, realm, K to compute $h(username, realm, K)$ and sends $Response(username, realm, h(username, realm, K))$ to the server.

Step 4: When the server receives the Response message, the server uses $username, realm, K$ to compute $h(username, realm, K)$. If the computed $h(username, realm, K)$ is the same as the $Response(h(username, realm, K))$, the server accepts the user's connection. Otherwise, the server rejects the user request.

2.2 Review Durlanik et al.'s Authentication Scheme

In this section, we review Durlanik et al.'s authentication scheme. This authentication scheme is based on Elliptic Curve Diffie-Hellman (ECDH).

2.2.1 Registration Phase

In Durlanik et al.'s authentication scheme, both the server and the user have a pre-shared password for authentication and an elliptic curve public key pair.

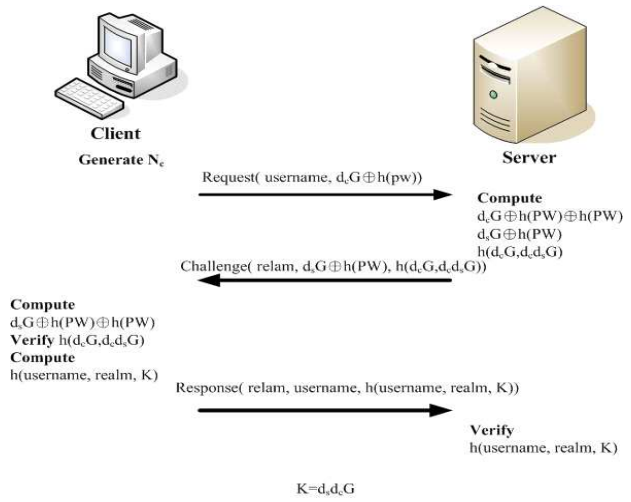

Figure 2: Durlanik et al.'s authentication scheme

2.2.2 Authentication Phase

If a legal user wants to login in system, this user must type his or her username and password. All steps of authentication phase execute as following.

Step 1: $U \longrightarrow S$: $Request(username, d_cG \oplus h(pw))$.

The user sends a Request message including its username and its public key xor by its hashed password.

Step 2: $S \longrightarrow U$: $Challenge(realm, d_sG \oplus h(PW), h(d_cG, d_cd_sG))$.

When the server receives the Request message, the server computes $d_cG \oplus h(PW) \oplus h(PW)$ to obtain d_cG. Then, the server computes a session key $K = d_sd_cG$, $d_sG \oplus h(PW)$, and $h(d_cG, d_cd_sG)$. Finally, the server send $Challenge(realm, d_sG \oplus h(PW), h(d_cG, d_cd_sG)$ message to the user.

Step 3: When the user receives the challenge messages, this user computes $d_sG \oplus h(PW) \oplus h(PW)$ to obtain d_sG. Then, the user computes $h(d_cG, d_cd_sG)$ to verify received $h(d_cG, d_cd_sG)$. If they are equal, the user computes a session key $K = d_sd_cG$.

Step 4: $U \longrightarrow S$: $Response(realm, username, h(username, realm, K))$.

The user computes $h(username, realm, K)$. Then, the user sends $Response(realm, username, h(username, realm, K))$ to the server.

Step 5: When the server receives $Response(realm, username, h(username, realm, K))$, the server computes $h(username, realm, K)$. Then, the server verifies Response message. If they are equal, the server accepts this connection. Otherwise, the server disconnect this connection.

3 Our Proposed Authentication Scheme

In this section, we reviewed our authentication scheme. Before we review our proposed authentication scheme for Session Initiation Protocol, we define the symbols first.

3.1 Registration Scheme

When a user wants to register and become a new legal user, this user must first submit his/her username and password to remote server. The username and password PW is used to verify the identity of the user and server. When the server receives this user's username and password, the server stores the user's username and password PW.

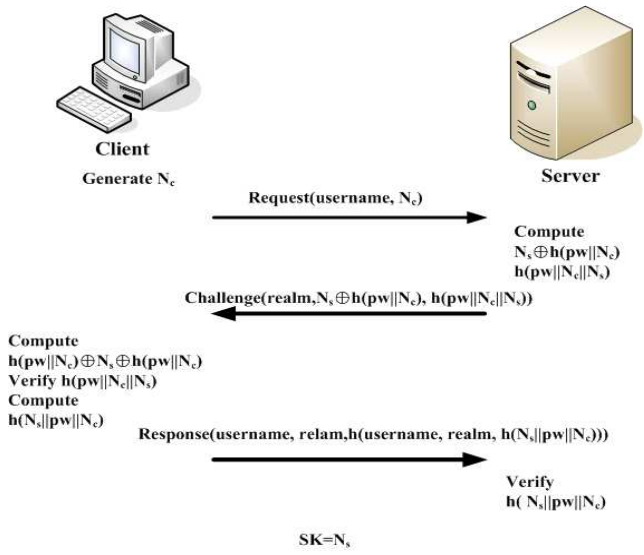

Figure 3: The general description of the head-end

3.2 Authentication Scheme

If a legal user wants to login in system, he/she must type his or her username and password. All steps of authentication phase execute as following.

Step 1: $U \longrightarrow S$: $Request(username, N_c)$.

The user generates a random number Nc and sends $Request(username, N_c)$.

Step 2: $S \longrightarrow U$: $Challenge(realm, N_s \oplus h(pw||N_c), h(pw||N_s||N_c))$.

When the server receives the Request message, the server generates a random N_s and uses N_s, N_c, pw to compute $N_s \oplus h(pw||N_c)$. Then, the server uses pw, N_s, N_c to compute $h(pw||N_s||N_c)$ and sends $Challenge(realm, N_s \oplus h(pw||N_c), h(pw||N_s||N_c))$ to the user.

Table 2: Symbol table

	Hash Function	Exclusive Or	Exponent-iation Computation	ECC Computation	Number of message flows	Session Key	Server Spoofing	Off-Line password guessing
Our	7	4	0	0	1.5	Yes	No	No
Yang et al.	7	4	4	0	1.5	No	No	No
Durlanik et al.	7	2	0	6	1.5	Yes	No	No

Step 3: $U \longrightarrow S$: $Response(username,\ realm,\ h(N_s||pw||N_c))$.

When the user receives the Response message, this user uses N_c, pw to compute $h(pw||N_c)$ and uses $h(pw||N_c)$, $N_s \oplus h(pw||N_c)$ to compute $h(pw||N_c) \oplus N_s \oplus h(pw||N_c)$ to get N_s. Then, the user uses pw, N_s, N_c to compute $h(pw||N_s||N_c)$. If the computed $h(pw||N_s||N_c)$ is not the same as $Challenge(h(pw||N_s||N_c))$, the user rejects the server request. Otherwise, the user uses N_s, pw, N_c to compute $h(N_s||pw||N_c)$ and sends $Response(username, realm, h(N_s||pw||N_c))$ to the server.

Step 4: When the server receives Response message, the server uses N_s, pw, N_c to compute $h(N_s||pw||N_c)$. If the computed $h(N_s||pw||N_c)$ is not the same as $Response(h(N_s||pw||N_c))$, the server rejects the user request. Otherwise, the server accepts the connection.

Step 5: After the server and the remote user authenticate each other, they use N_s as a session key $SK = Ns$.

4 Security Analysis

Below we can examine whether the communication agreement is safety, we will examine our authentication scheme on various known attacks.

4.1 Replay Attack

Our authentication scheme uses the random nonce to withstand the replay attack. Authentication messages $N_s \oplus h(pw||N_c)$, $h(pw||N_s||N_c)$, $h(N_s||pw||N_c)$ are generated by random nonce N_c and the random nonce N_s. The random nonce N_c and the random nonce N_s are generated independently, and both values will deficient in each session. Assume an adversary wants to enter the system. This adversary can not enter the system by re-sending messages ever transmitted by a legal user, because the random nonce N_s is different.

4.2 Password Guess Attack

In our authentication scheme, it is impossible for an adversary to guess the user's password. The password of a user was protected by the random number Nc and the random number N_s. The random number N_s is generated by the server and the random number N_c is generated by the user. They are different in the next session, so our authentication scheme can withstand password guess attack.

4.3 Server Spoofing Attack

In our authentication scheme, we will obviously ask the user to know whether the server is the correct one before authenticating the user. Therefore, if an attacker wants to masquerade as the server to cheat the user, this attacker must have the user password PW. If someone is discovered to masquerade as the server to cheat the user, the user will disconnect all following transmissions. Hence, any server spoofing attack will fails.

4.4 Impersonation Attack

In our scheme, an attacker can not masquerade as a legal user. To successfully perform the impersonation attack, the attacker must require the knowledge of PW to generate and interpret authentication messages correctly, because all the authentication messages between the server and the user are protected by pw. pw is memorized by the user. If someone is discovered to masquerade as the legal user to cheat the server, the server will disconnect all following transmissions. Hence, any impersonation attack will fails.

4.5 Security of Session Key

A session key is generated by user's password PW and a random number. Whenever the communication ends between the user and the server, the key will self destruct immediately and will not be reused at the next time. When the user reenters the system, a new session key will be generated to encrypt the information during the communication process. Therefore assuming the attacking has obtained a session key, the person will not be able to use the session key to decode the information in other communication processes. Because the random nonce N_s and random nonce N_c is generated randomly, it will not be able to use a known session key to calculate the value of the next session key.

5 Compare with other Authentication Scheme

In this section, we compare our nonce-based authentication scheme with Yang et al.'s authentication scheme [9] and Durlanik et al.'s authentication scheme [3]. All comparisons are described as Table 2.

6 Conclusion

In this paper, we describe Yang et al.'s authentication scheme. The computation cost of Yang et al.'s authentication schemes is very high. In order to improve this flaw, we propose a nonce-based authentication scheme. This nonce-based authentication scheme is only based on nonce. The computation cost of this authentication scheme is lower than Yang et al.'s authentication schemes. It is very suitable for low computation equipment.

References

[1] J. Arkko, et al., "Security mechanism agreement for SIP sessions", *IETF Internet Draft (draft-ietf-sip-sec-agree-04.txt)*, June 2002.

[2] W. Diffie, and M. Hellman, "New directions in cryptology," *IEEE Transaction on Information Theory*, vol. 22, no. 6, 1976.

[3] A. Durlanik, and I. Sogukpinar, "SIP authentication scheme using ECDH," *World Enformatika socity Transaction on Engineering computing and technology*, vol.8, pp. 350-353, 2005.

[4] J. Franks, et al., "HTTP authentication: basic and digest access authentication", *IETF RFC2617*, June 1999.

[5] M. Handley, and et al., *SIP: session initiation protocol*, *IETF RFC2543*, March 1999.

[6] J. Rosenberg, et al., "SIP: session initiation protocol", *IETF RFC3261*, June 2002.

[7] M. Thomas, *SIP Security Requirements*, IETF Internet Draft (draftthomas-sip-sec-reg-00. txt), Nov. 2001 (work in progress).

[8] L. Veltri, S. Salsano, and D. Papalilo, "SIP security issues: the SIP authentication procedure and its processing load, " *IEEE Network*, vol. 16, no. 6, pp. 38-44, 2002.

[9] C. C. Yang, R. C. Wang, and W. T. Liu, "Secure authentication scheme for session initiation protocol," *Computers and Security*, vol. 24, pp. 381-386, 2005.

Jia-Lun Tsai received her M.S. degrees in E-learning from National Chiao Tung University, in 2007 respectively. His reaches interests include Authentication scheme, Network Security, Wireless Security, and Cryptography etc..