LETTER

Comments on YEH-SHEN-HWANG's One-Time Password Authentication Scheme

Il-Sun YOU[†a)] ***and*** **Kyungsan CHO**[†]**,** ***Members***

SUMMARY Recently, Yeh, Shen and Hwang proposed an one-time password authentication scheme, which enhances the S/KEY scheme to resist server spoofing attacks, preplay attacks and off-line dictionary attacks. In this letter, the weaknesses and inconveniences of their scheme are demonstrated.

key words: *authentication, S/KEY, one-time password, smart card*

1. Introduction

The S/Key one-time password scheme is designed to counter replay attacks or eavesdropping attacks [2], [3]. With this scheme, the user's secret pass-phrase never needs to cross the network at any time such as during authentication or during pass-phrase changes. Moreover, no secret information need be stored on any system, including the server being protected. Although the S/KEY scheme thus protects against passive attacks based on replaying captured reusable passwords, it is vulnerable to server spoofing attacks, preplay attacks and off-line dictionary attacks [1], [4], [5]. Recently, Yeh, Shen and Hwang proposed a one-time password authentication scheme, which enhances the S/KEY scheme to resist against the above attacks [1]. The authentication scheme uses smart cards to securely preserve a pre-shared secret $SEED$ and simplify the user login process. In addition, it provides a session key to enable confidential communication over the network. Although their scheme can overcome the flaws of the S/KEY scheme as they claimed, we discover that it is vulnerable to other attacks such as denial of service attacks, stolen-verifier attacks and Denning-Sacco attacks [5]–[7]. Because the scheme uses a pre-shared secret $SEED$ and the user's weak pass-phrase, the leakage of $SEED$ causes the scheme to retain the flaws of the S/KEY scheme. In this letter, the weaknesses and inconveniences of Yeh-Shen-Hwang's scheme are demonstrated.

2. Review of Yeh-Shen-Hwang's Scheme

Yeh-Shen-Hwang's scheme is composed of registration stage, login stage and authentication stage [1]. Let the one-way hash function and the user secret, derived from the user's pass-phrase, be denoted as $H()$ and K, respectively.

Manuscript received June 30, 2003.
Manuscript revised April 27, 2004.

[†]The authors are with the Division of Information and Computer Science, Dankook University, San #8 Hannam-dong, Youngsan-gu, Seoul 140-714, Korea.
a) E-mail: qjemfahr@hanmail.net

2.1 Registration Stage

(1) User ← Server: N, $SEED \oplus D$, $H(D)$
(2) User → Server: $p_0 \oplus D$

It is assumed that the server initially issues a smart card containing a pre-shared secret $SEED$, a large random number it generates, to the user. To register the user, the server generates a random number D, computes $SEED \oplus D$ and $H(D)$, and sends two values along with N, a permitted number of login times, to the user. After receiving, the user XORs the received $SEED \oplus D$ with $SEED$ stored in the smart card to extract D. If D's hash value and the received $H(D)$ are equal, the user computes the initial password $p_0 = H^N(K \oplus SEED)$, and then sends $p_0 \oplus D$ to the server.

2.2 Login Stage

(1) User ← Server: C, $SEED \oplus D$, $H(D) \oplus p_{t-1}$
(2) User → Server: $p_t \oplus D$

For the tth login, the server generates a random number D, computes two values $SEED \oplus D$ and $H(D) \oplus p_{t-1}$, and then sends the two values along with $C(= N - t)$ to the user. After receiving, the user XORs the received $SEED \oplus D$ with $SEED$ stored in the smart card to extract D, and verifies the received $H(D) \oplus p_{t-1}$. If it is valid, the user computes the fresh password $p_t = H^C(K \oplus SEED)$, and then sends $p_t \oplus D$ to the server.

2.3 Authentication Stage

After receiving, the server XORs the received $p_t \oplus D$ with D to obtain the fresh one-time password p_t. If the hash value of p_t is equal to p_{t-1} stored in the server, then the user is authenticated. Finally, the server updates the last one-time password with p_t and the counter value with C in its database. D, randomly generated by the server, can be used as a session key to enable confidential communication between the server and the user.

3. Weaknesses of Yeh-Shen-Hwang's Scheme

Because Yeh-Shen-Hwang's scheme contains no message integrity check, and uses a pre-shared secret $SEED$ and the user's weak pass-phrase, it has the following vulnerabilities.

Copyright © 2005 The Institute of Electronics, Information and Communication Engineers

3.1 Vulnerability to the Denial of Service Attack

Denial of service (DoS) attack is an offensive action whereby an attacker could use some methods to work upon a server so that the access requests issued by the legitimate user will be denied by the server [7]. In Yeh-Shen-Hwang's scheme, an attacker can change the messages for the registration stage without being detected, resulting in desynchronization between a server and a user. Such desynchronization causes the server to deny the user's access request.

In this scheme, DoS attack can be launched as follows. Assumes that an attacker can eavesdrop, record, inject, reorder, and re-send (altered) messages. To register the user, the server makes a message for step (1) of the registration stage, and then sends the message to the user. During step (1), an attacker can replace N of the message with $\underline{N}$, resulting in desynchronization between the server and the user. After receiving and verifying the message, the user computes the initial password $p_0' = H^{\underline{N}}(K \oplus SEED)$, and then sends $p_0' \oplus D$ to the server. Later, the server with p_0' will deny the user's access request, even though the user correctly generates one-time password. In addition, the attacker can replace $p_0 \oplus D$ of step (2) with an equal-sized random number r, which causes the server to extract the wrong initial password p_0'.

Thus, it is possible for any attacker to easily mount a denial of service attack without using any cryptographic methods.

3.2 Vulnerability to the Stolen-Verifier Attack

Stolen-verifier attack is an offensive action carried by an attacker who has compromised the password database and obtained a verifier for a particular user. Recently, as attacks conducted by internal users become increased and more critical, security schemes are strongly required to resist this attack [7]. Unfortunately, Yeh-Shen-Hwang's scheme is vulnerable to this attack, since it uses $SEED$ as a pre-shared secret. If stolen, $SEED$ can be used as a verifier, through which an attacker can mount off-line dictionary attacks to learn a user secret K. Since the user secret K is derived from the user's pass-phrase, such off-line dictionary attacks are available.

Assumes that an attacker succeeds in stealing a user's $SEED$ from the server. The attacker can mount the following off-line dictionary attack by using the value. Iterating upon all possible choices of secret K: ① Pick a candidate K' ② Compute $p_t' = H^C(K' \oplus SEED)$ ③ Compare the tth one-time password p_t and p_t'. A match in the last step indicates correct guess of the user secret. The tth one-time password p_t can be obtained as follows. The attacker intercepts the $(t+1)$th message from step (1) of the login stage, XORs the intercepted $SEED \oplus D$ with the stolen $SEED$ to extract D, and then XORs the intercepted $H(D) \oplus p_t$ with D's hash value to obtain the tth one-time password p_t.

In addition, by using the stolen $SEED$ and p_t, the attacker can forge the message for step (1) of the login stage to impersonate the server to the user, and then obtain a fresh one-time password that is guaranteed to be valid at some point on the future.

Thus, the leakage of $SEED$ enables the stolen-verifier attack, which causes Yeh-Shen-Hwang's scheme to be still vulnerable to off-line dictionary attacks and preplay attacks. Furthermore, the stolen $SEED$ allows the attacker to compromise past session keys.

3.3 Vulnerability to the Denning-Sacco Attack

The Denning-Sacco Attack is an offensive action where an attacker captures a session key from an eavesdropped session and uses the key either to gain the ability to impersonate the user directly or to mount a dictionary attack on the user's password [5], [6]. Yeh-Shen-Hwang's scheme is vulnerable to the Denning-Sacco attack based on a compromised session key D.

Assumes that an attacker records all messages exchanged between the server and the user during a session and somehow obtains the old session key D. The attacker XORs $SEED \oplus D$ obtained from the recorded messages with the session key D to extract $SEED$. Finally, the attacker initiates the off-line dictionary attack mentioned in 3.2 by using the extracted $SEED$, p_t and C, where p_t and C are obtained from the recorded messages. Furthermore, the extracted $SEED$ enables preplay attacks as mentioned in 3.2.

3.4 Inconveniences

Yeh-Shen-Hwang's scheme causes the following inconveniences to the user.

First, as the scheme provides no method for the server to securely distribute $SEED$ to the user, the user should request the administrator to directly generate $SEED$ and store it in the user's smart card in the case of reinitializing the user's login information such as N and $SEED$.

Second, because of using the user secret K derived from the user's pass-phrase, the user should input both the Personal Identification Number (PIN) of the smart card and the pass-phrase for the registration or login process.

4. Discussion

To improve Yeh-Shen-Hwang's scheme, we propose the followings.

Because of having no way to detect that messages are changed or altered in the registration stage, Yeh-Shen-Hwang's scheme is vulnerable to DoS attacks. Therefore, it is desirable to use cryptographic techniques such as MAC, HMAC or digital signature for message integrity check.

The stolen-verifier attack, the Denning-Sacco attack and the inconvenience of reinitializing the user's login information result from a pre-shared secret $SEED$ and the user's weak secret K. Furthermore, the pre-shared secret $SEED$

cause the scheme not to truly achieve the strength of the S/KEY scheme that no secret information need be stored on the server, while not defending against server compromise. Therefore, it is desirable to strengthen the user secret K rather than use $SEED$ as a pre-shared secret. The user secret K can be strengthened to avoid off-line dictionary attacks in a way that it is randomly generated and stored in the smart card instead of being derived from the pass-phrase.

In practice, the difficulty of stealing a secret, such as a user secret K or a session key D, may rely on how the authentication protocol is implemented and deployed. Since we cannot guarantee that secrets will never be compromised in real systems, the threat analysis is valuable to estimate the strength of security schemes [7].

5. Conclusion

In this letter, we have shown that Yeh-Shen-Hwang's scheme is vulnerable to denial of service attacks, stolen-verifier attacks and Denning-Sacco attacks. Furthermore, some inconveniences are described.

Acknowledgments

The present research was conducted by the research fund of Dankook University in 2004.

References

[1] T.C. Yeh, H.Y. Shen, and J.J. Hwang, "A secure one-time password authentication scheme using smart cards," IEICE Trans. Commun., vol.E85-B, no.11, pp.2515–2518, Nov. 2002.
[2] N. Haller, C. Metz, P. Nesser, and M. Straw, "A one-time password system," RFC 2289, Feb. 1998.
[3] N. Haller, "The S/KEY one-time password," RFC 1760, Feb. 1995.
[4] C.J. Mitchell and L. Chen, "Comments on the S/KEY user authentication scheme," ACM Operating Systems Review, vol.30, no.4, pp.12–16, Oct. 1996.
[5] D. Denning and G. Sacco, "Timestamps in key distribution systems," Commun. ACM, vol.24, no.8, pp.533–536, Aug. 1981.
[6] S. Kim, B. Kim, S. Park, and S. Yen, "Comments on password-based private key download protocol of NDSS'99," Electron. Lett., vol.35, no.22, pp.1937–1938, 1999.
[7] W.C. Ku, C.M. Chen, and H.L. Lee, "Cryptoanalysis of a variant of Peyravian-Zunic's password authentication scheme," IEICE Trans. Commun., vol.E86-B, no.5, pp.1682–1684, May 2003.