International Journal of Security and its Applications
Vol. 2, No. 2, April, 2008

Cryptanalysis and Improvement on Lee-Chen's One-Time Password Authentication Scheme

Chun-Li Lin
Department of Computer Science and Information Engineering
Shu-Te University
cllin@mail.stu.edu.tw

Ching-Po Hung
Department of Computer Science and Information Engineering
Shu-Te University
118760@mail.csc.com.tw

Abstract

Yeh et al., in 2002, proposed a one-time password authentication scheme using smart cards. Subsequently, Tsuji et al. and Ku et al. respectively showed that Yeh et al.'s scheme is vulnerable to stolen-verifier attacks. Recently, Lee and Chen proposed an improvement of Yeh et al.'s scheme. Lee and Chen claimed that their improvement can effectively withstand the stolen-verifier attack and is as efficient as Yeh et al.'s scheme. This paper, however, will point out that Lee and Chen's improvement is still vulnerable to a masquerade attack. And, a simple improvement is given to resist the masquerade attack.

1. Introduction

User authentication is a very important security mechanism for many network applications. Traditional "static" password authentication techniques are widely used due to their convenience. However, they often suffer from eavesdropping, replaying, and guessing attacks. Moreover, many malicious programs with Trojan Horse may steal passwords from the victim's computer, no matter what applications or protocols these passwords are used for. One-time password authentication, in which every password is used only once, provides stronger user authentication via "dynamic" passwords than traditional static password authentication. Once a one-time password is used, it will be no longer valid even if it is eavesdropped, replayed, guessed or stolen. One-time password authentication usually requires a user-side device, called token (e.g. a smart card), for the generation of dynamic passwords. Even though it needs an additional token, onetime password authentication is still more cost-effective than PKI-based authentication solutions. Since Lamport [1] brought up the first one-time password authentication scheme, there have been many subsequent researches [2][3][4][5][6].

In 2002, Yeh et al. [7] proposed a one-time password authentication scheme using smart cards. This scheme is an enhancement of S/KEY [2] and was claimed to be free from any of server spoofing attack, preplay attacks, and off-line dictionary attacks. However, Tsuji et al. [8] and Ku et al. [9] later respectively showed that Yeh et al.'s scheme is vulnerable to stolen-verifier attacks. Recently, Lee and Chen [10] proposed an improvement of Yeh et al.'s scheme. Lee and Chen claimed that their improvement can effectively withstand the stolen-verifier attack and is as efficient as Yeh et al.'s scheme. This paper, however, will point out that Lee and Chen's improvement is still vulnerable to a masquerade attack. Furthermore, a simple improvement is given to resist the masquerade attack.

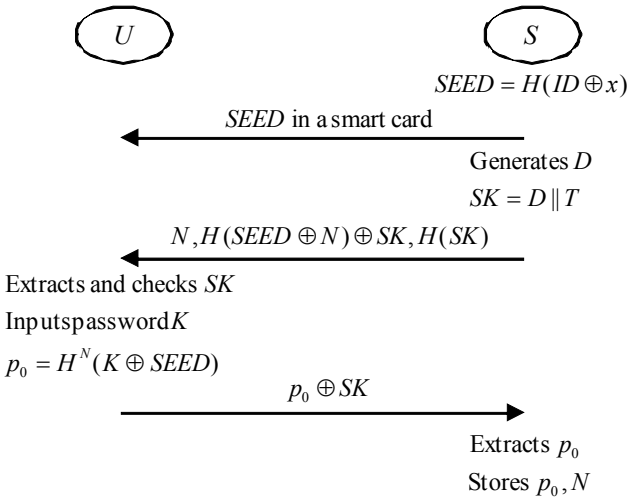

Figure 1. Registration stage of the Lee-Chen scheme

2. Review of the Lee-Chen scheme

This section reviews the Lee-Chen scheme. We first list notations used throughout this paper as follows:

U	the user
S	the server
E	the attacker
$SEED$	a pre-shared secret between U and S
D_i	a large random number generated by the server
K	the user's secret key/password
$H(\cdot)$	a secure one-way hash function
N	the maximum allowable number of login attempts
C_i	$C_i = N - i$
$\oplus$	a bit-wise exclusive-or (XOR) operation
p_i	$p_i = H^{N-i}(K \oplus SEED)$, where $N - i$ is the number of hash iterations. For example, $H^3(K \oplus SEED) = H(H(H(K \oplus SEED)))$
$\|\|$	the concatenation of two bit strings

The Lee-Chen scheme is divided into three stages: the registration stage, the login stage, and the authentication stage. We describe these three stages as follows.

2.1. Registration stage

Figure 1 shows the registration stage of the Lee-Chen scheme.

(1) $U \leftarrow S : SEED$

Firstly, S computes a value $SEED = H(ID \oplus x)$, and issues to U a smart card, in which the pre-shared secret $SEED$ is stored. ID denotes the identity of U and x is the secret of S.

(2) $U \leftarrow S : N, H(SEED \oplus N) \oplus SK, H(SK)$

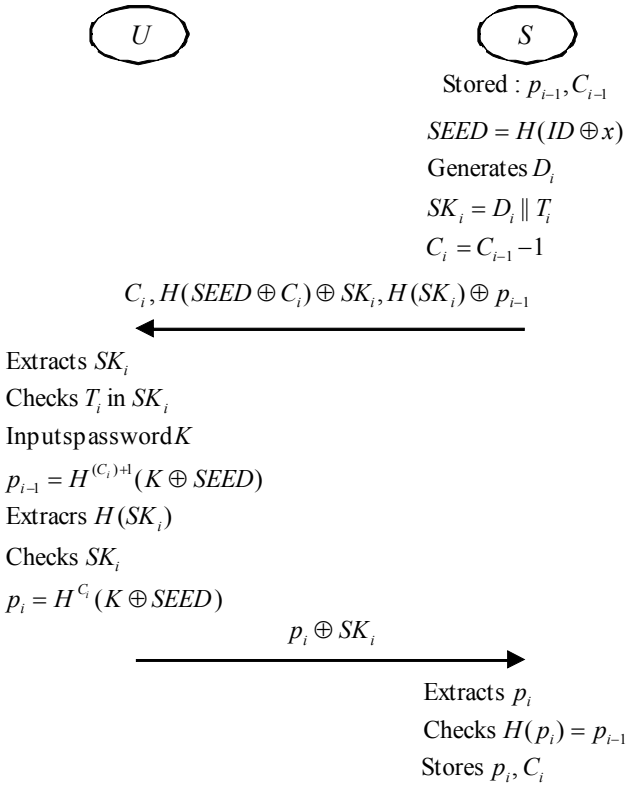

Figure 2. The *i*th login and authentication stages of the Lee-Chen scheme

Then, S generates a random number D and computes $SK = D||T$, where T is a timestamp. S also decides a number N which indicates the maximum allowable number of login attempts of U. Next, S computes $H(SEED \oplus N) \oplus SK$, $H(SK)$ and sends N, $H(SEED \oplus N) \oplus SK$, $H(SK)$ to U.

(3) $U \rightarrow S : p_0 \oplus SK$

Once receiving the data from S, U uses the $SEED$, which is stored in the smart card, to compute $H(SEED \oplus N)$ and then extracts SK by performing $[H(SEED \oplus N) \oplus SK] \oplus H(SEED \oplus N)$. U further checks the integrity of SK with the received $H(SK)$. If the check is valid, U inputs password K and computes $p_0 = H^N(K \oplus SEED)$. Then U sends $p_0 \oplus SK$ to S. Once receiving $p_0 \oplus SK$, S extracts p_0 by performing $(p_0 \oplus SK) \oplus SK$ and then stores p_0 as a verifier for later authentication of U.

2.2. Login stage

When the user U wants to login the system, U and S execute the ith session of the Lee-Chen protocol. Figure 2 shows the ith login and authentication stages of the Lee-Chen scheme.

(1) $U \leftarrow S : C_i, H(SEED \oplus C_i) \oplus SK_i, H(SK_i) \oplus p_{i-1}$

For the ith login, S first computes $SEED = H(ID \oplus x)$ and generates a random number D_i. Let $SK_i = D_i||T_i$, where T_i is the timestamp. Next, S sends C_i, $H(SEED \oplus C_i) \oplus SK_i$ and $H(SK_i) \oplus p_{i-1}$ to U, where (p_{i-1}, C_{i-1}) is the stored data and $C_i = C_{i-1} - 1 = N - i$.

(2) $U \rightarrow S : p_i \oplus SK_i$

After receiving the values from S, U uses the $SEED$, which is stored in the smart card, to compute $H(SEED \oplus C_i)$ and then extracts SK_i by performing $[(H(SEED \oplus C_i) \oplus SK_i)] \oplus H(SEED \oplus C_i)$. U checks the timestamp T_i in the SK_i. If the timestamp is valid, U computes $p_{i-1} = H^{(C_i)+1}(K \oplus SEED)$. U uses p_{i-1} to extract $H(SK_i)$ from $H(SK_i) \oplus p_{i-1}$ and further checks the integrity of SK_i with $H(SK_i)$. If the check is valid, U computes $p_i = H^{C_i}(K \oplus SEED)$ and sends $p_i \oplus SK_i$ to S.

2.3. Authentication stage

Once receiving $p_i \oplus SK_i$, S obtains p_i by performing $(p_i \oplus SK_i) \oplus SK_i$. Then, S computes $H(p_i)$ and compares it with the stored verifier p_{i-1}. If they are equivalent, S replaces (p_{i-1}, C_{i-1}) with (p_i, C_i) in the database.

3. Cryptanalysis of the Lee-Chen scheme

In this section, we illustrate a masquerade attack on the Lee-Chen scheme. The masquerade attack consists of *aborting phase* and *masquerading phase*. Figure 3 shows the flow of the masquerade attack.

Aborting phase

In the aborting phase, the attacker E tries to abort the ith login stage and record messages between the user and the server.

(1) $U \leftarrow S : C_i, H(SEED \oplus C_i) \oplus SK_i, H(SK_i) \oplus p_{i-1}$

The attacker E intercepts the transmitted message, especially the item $H(SEED \oplus C_i) \oplus SK_i$.

(2) $U \rightarrow E : p_i \oplus SK_i$

In a regular situation, U will authenticate S by checking the timestamp and the integrity of SK_i. Then, U responds correct $p_i \oplus SK_i$. Now, the attacker E interrupts this transmission to S, and records the message $p_i \oplus SK_i$.

(3) $E \rightarrow S : X$

The attacker E forwards to S an arbitrary random number X with the same length of $p_i \oplus SK_i$. Obviously, the value X is invalid for the authentication to S.

Now, the ith login is aborted and the next login will remain the ith one, because the server still keeps the verifier p_{i-1}. Note that the attacker E has recorded $H(SEED \oplus C_i) \oplus SK_i$ and $p_i \oplus SK_i$.

Masquerading phase

Now, the attacker can masquerade as the user U to submit a login request.

(1) $E \leftarrow S : C_i, H(SEED \oplus C_i) \oplus SK'_i, H(SK'_i) \oplus p_{i-1}$

In this ith login, S generates a new random number D'_i , and $SK'_i = D'_i||T'_i$, where T'_i is also a new timestamp.

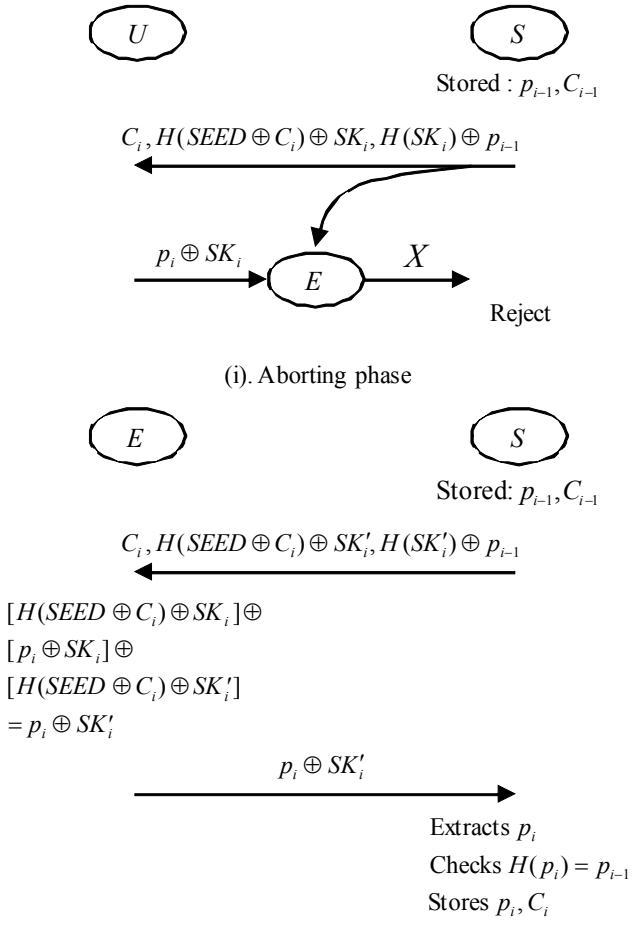

Figure 3. Masquerade attack on the Lee-Chen scheme

(2) $E \rightarrow S : p_i \oplus SK'_i$

Once receiving the message from S, the attacker E computes

$[H(SEED \oplus C_i) \oplus SK_i] \oplus [p_i \oplus SK_i] \oplus [H(SEED \oplus C_i) \oplus SK'_i] = p_i \oplus SK'_i$.

Then, E sends $p_i \oplus SK'_i$ to S. Next, S follows the procedure of authentication stage and the authentication will succeed.

Impact of this attack

The significance of this attack is that the user and the server cannot detect the attacker's masquerade. The verifiers p_i stored in the server are a hash chain of ($K \oplus SEED$). This masquerade attack does not modify any element of the chain. Moreover, the element of the chain that the user computes for the authentication to the server depends on the counter C_i from the server, not upon a synchronized value or event. Thus, the attacker can arbitrarily abort a user's login stage and then masquerade the user without detection.

4. Simple improvement

To resist the masquerade attack, we propose a simple improvement on the Lee-Chen scheme.

Registration stage

(1) $U \leftarrow S : SEED$

(2) $U \leftarrow S : N, H(SEED \oplus N) \oplus SK, \boldsymbol{H^2(SK)}$

We replace $H(SK)$ in the Lee-Chen scheme with $H^2(SK)$.

(3) $U \rightarrow S : p_0 \oplus SK, \boldsymbol{H(p_0)}$

Once receiving the data from S, U does the same operations as the Lee-Chen scheme except checks the integrity of SK with the received $H^2(SK)$. Then U sends $p_0 \oplus SK$ and $H(p_0)$ to S. We insert $H(p_0)$ for S to check the integrity of p_0 before storing it.

Login stage

(1) $U \leftarrow S : C_i, H(SEED \oplus C_i) \oplus SK_i, \boldsymbol{H^2(SK_i)} \oplus p_{i-1}$

We replace $H(SK_i) \oplus p_{i-1}$ in the Lee-Chen scheme with $H^2(SK_i) \oplus p_{i-1}$.

(2) $U \rightarrow S : p_i \oplus \boldsymbol{H(SK_i)}$

After receiving the values from S, U does the same operations as the Lee-Chen scheme except extracts $H^2(SK_i)$ from $H^2(SK_i) \oplus p_{i-1}$ and further checks the integrity of SK_i with $H^2(SK_i)$. For the message that U sends to S, we replace $p_i \oplus SK_i$ in the Lee-Chen scheme with $p_i \oplus H(SK_i)$.

Authentication stage

Once receiving $p_i \oplus H(SK_i)$, S obtains p_i by performing $[p_i \oplus H(SK_i)] \oplus H(SK_i)$. Then, S does the same operations as the Lee-Chen scheme.

Examination of the masquerade attack

According to the masquerade attack described in Section 3, the attacker E will computes

$[H(SEED \oplus C_i) \oplus SK_i] \oplus [p_i \oplus H(SK_i)] \oplus [H(SEED \oplus C_i) \oplus SK'_i]$

$= SK_i \oplus p_i \oplus H(SK_i) \oplus SK'_i$.

Obviously the result is not equivalent to the expected value $p_i \oplus H(SK'_i)$, hence the server will reject this login request.

5. Conclusions

In this paper, we demonstrate that the Lee-Chen one-time password authentication scheme is still vulnerable to a masquerade attack, though Lee and Chen claimed that their improvement on the Yeh-Shen-Hwang scheme can withstand the stolen-verifier attack. By aborting one regular login session and recording those transmitted messages, an attacker can successfully masquerade as the victim user to login without detection. We further propose a simple improvement to resist the masquerade attack.

6. References

[1] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, 24(11):770–772, November 1981.

[2] N.M. Haller, "The S/KEY (TM) one-time password system", Proc. Internet Society Symposium on Network and Distributed System Security, 151–158, 1994.

[3] M. Sandirigama, A. Shimizu, and M.T. Noda, "Simple and secure password authentication protocol (SAS)", IEICE Transactions on Communications, E83-B(6):1363–1365, June 2000.

[4] C.L. Lin, H.M. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication", IEICE Transactions on Communications, E84-B(9):2622–2627, September 2001.

[5] C.M. Chen and W.C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols", IEICE Transactions on Communications, E85-B(11):2519–2521, November 2002.

[6] T. Tsuji and A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA", IEICE Transactions on Communications, E86-B(7):2182–2185, July 2003.

[7] T.C. Yeh, H.Y. Shen, and J.J. Hwang, "A secure one-time password authentication scheme using smart cards", IEICE Transactions on Communications, E85-B(11):2515–2518, November 2002.

[8] T. Tsuji and A. Shimizu, "Cryptanalysis on one-time password authentication schemes using counter value", IEICE Transactions on Communications, E87-B(6):1756–1759, June 2004.

[9] W.C. Ku, H.C. Tsai, and M.J. Tsaur, "Stolen-verifier attack on an efficient smartcard-based one-time password authentication scheme", IEICE Transactions on Communications, E87-B(8):2374–2376, August 2004.

[10] N.Y. Lee and J.C. Chen, "Improvement of one-time password authentication scheme using smart cards", IEICE Transactions on Communications, E88-B(9):3765–3767, September 2005.

Authors

Chun-Li Lin was born in Pingtung, Taiwan, ROC, in 1967. He received his B.S. degree in Information Science from Tunghai University, Taiwan, in 1990, his M.S. degree in Information Engineering from National Cheng Kung University, Tainan, Taiwan, in 1992, and his Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University in 2003. He is presently an assistant professor in Department of Computer Science and Information Engineering, Shu-Te University. His research interests include Cryptography and Network Security.

Ching-Po Hung was born in Tainan, Taiwan, ROC, in 1959. He received his B.S. degree in Industrial Engineering from Tunghai University, Taiwan, in 1982, and his M.S. degree in Information Engineering from Shu-Te University, Kaohsiung, Taiwan, in 2006. He is presently first year of phd study in Department of Information Engineering, I-Shou University, and an employee of China Steel Co. Ltd., Kaohsiung, Taiwan. His research interests include Cryptography and Network Security.