Improvement of One-Time Password Authentication Scheme Using Smart Cards

Narn-Yih LEE†a, Member and Jung-Chic CHEN†, Nonmember

SUMMARY In 2002, Yeh, Shen, and Hwang proposed a one-time password authentication scheme using smart cards. However, Tsuji et al. and Ku et al. showed that it is vulnerable to the stolen verifier attack. Therefore, this paper proposes an improved one-time password authentication scheme, which not only keeps the security of the scheme of Yeh-Shen-Hwang but also can withstand the stolen verifier attack.

key words: cryptography, security, one-time password, user authentication, smart card

1. Introduction

In recent years, internet applications and mobile techniques have been developed significantly. Since the internet is open and users do not see each other, how to correctly authenticate the identities of users has become very important. Password authentication scheme is the most common method to identify users. Many password authentication schemes [1]–[7] have been proposed for electronic commerce environments. In 2002, Yeh, Shen, and Hwang [8] proposed a secure one-time password authentication scheme using smart cards. The authors claimed that it can withstand many various attacks such as replay attacks, server spoofing attacks, off-line dictionary attacks, and active attacks. However, Tsuji et al. [9] and Ku et al. [10] showed that the scheme of Yeh-Shen-Hwang is still vulnerable to the stolen verifier attack. In this paper, we propose an improved one-time password authentication scheme. This improved scheme is able to prevent the stolen verifier attack and is as efficient as the scheme of Yeh-Shen-Hwang.

The rest of this paper is as follows. In Sect. 2, we review Yeh-Shen-Hwang scheme and the stolen verifier attacks of Tsuji et al. and Ku et al.. In Sect. 3, an improved scheme is introduced. In Sect. 4, we analyze the security of the improved scheme. Finally, this paper concludes in Sect. 5.

2. Review of Yeh-Shen-Hwang Scheme and the Stolen Verifier Attacks

In Sect. 2.1, some definitions and notations used in this paper are introduced. In Sect. 2.2, we review Yeh-Shen-Hwang scheme, and then the stolen verifier attacks of Tsuji et al. and Ku et al. are respectively introduced in Sects. 2.2 and 2.3.

2.1 Definitions and Notations

The definitions and notations used in this paper are described as follows.

- $U$: the user
- $S$: the server
- $E$: the attacker
- $SEED$: a pre-shared secret of $S$ and $U$
- $D_i$: a large random number generated by the server
- $K$: the user’s secret key/password
- $H(\cdot)$: a hash function
- $N$: the number of logins
- $C_i$: $C_i = N - i$
- $p_i$: $p_i = H^{i}(K \oplus SEED)$, where $N - i$ is the number of hash iterations. For example, $H^2(K \oplus SEED) = H(H(K \oplus SEED))$.

2.2 Review of Yeh-Shen-Hwang Scheme

In [8], Yeh, Shen and Hwang proposed a secure one-time password authentication scheme, which adopts smart cards to store users’ secret values. The scheme of Yeh-Shen-Hwang is divided into three stages: the registration stage, the login stage, and the authentication stage. We describe the three stages of Yeh-Shen-Hwang scheme as follows.

2.2.1 Registration Stage

1. $U \leftarrow S : SEED$
2. $U \leftarrow S : N, SEED \oplus D, H(D)$
3. $U \rightarrow S : p_0 \oplus D$

Initially, $S$ selects a large random number $SEED$, and issues a smart card containing the pre-shared secret $SEED$ to $U$. Then, $S$ selects a random number $D$ and a number $N$, and computes $H(D)$. $S$ performs the XOR operation on $SEED$ and $D$, and sends $N, SEED \oplus D, H(D)$ to $U$. Once receiving them, $U$ extracts $D$ by performing the XOR operation on $SEED \oplus D$ and the $SEED$ which is stored in the smart card. Then, $D$ is hashed one time and compared with the $H(D)$. If it is equivalent, the identity of $S$ is authenticated. Then, $U$ computes $p_0 \oplus D$ and sends it to $S$, where $p_0 = H^N(K \oplus SEED)$. 

Manuscript received January 27, 2005.
†The authors are with the Information Management Department, Southern Taiwan University of Technology, Tainan County 710, Taiwan, R.O.C.
a) E-mail: nylee@mail.stut.edu.tw
DOI: 10.1093/ietcom/e88–b.9.3765
2.2.2 Login Stage

1. $U \leftarrow S : C_i, SEED \oplus D_i, H(D_i) \oplus p_{i-1}$
2. $U \rightarrow S : p_i \oplus D_i$

For the $i$th login, $S$ first generates a random number $D_i$, and then computes two values $H(D_i) \oplus p_{i-1}$ and $SEED \oplus D_i$, where $p_{i-1} = H^{N-i-1}(K \oplus SEED)$. Then, $S$ sends $C_i$, $H(D_i) \oplus p_{i-1}$ and $SEED \oplus D_i$ to $U$, where $C_i = N - i$. After receiving those values, $U$ extracts $D_i$ by performing the XOR operation on $SEED \oplus D_i$ and $SEED$. $D_i$ is hashed one time and compared with $H(D_i)$, which is extracted by $(H(D_i) \oplus p_{i-1}) \oplus p_{i-1}$. If it is equivalent, the identity of $S$ is authenticated. Then, $U$ computes $p_i$ and performs the XOR operation with $D_i$. Finally, $U$ sends $p_i \oplus D_i$ to $S$.

2.2.3 Authentication Stage

Once receiving $p_i \oplus D_i$, $S$ obtains $p_i$ by performing the XOR operation on $p_i \oplus D_i$ and $D_i$. Then, $p_i$ is hashed one time and compared with $p_{i-1}$. If it is equivalent, the identity of $U$ is authenticated. Finally, $S$ replaces $p_{i-1}$ and $C_{i-1}$ with $p_i$ and $C_i$ in the database.

2.3 Tsuji et al. Attack

In [9], Tsuji et al. showed that the scheme of Yeh-Shen-Hwang is vulnerable to the stolen-verifier attack. The attacking procedure is presented in the following.

First, an attacker $E$ eavesdrops the communications of $(i-1)$th login as follows.

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

If $H(p_{i-1}) = p_{i-2}$, $S$ replaces $p_{i-2}$ with $p_{i-1}$. Next, $E$ steals $p_{i-1}$ from $S$ by the stolen verifier attack. Then $E$ extracts $SEED$ by $(SEED \oplus D_{i-1}) \oplus (p_{i-1} \oplus D_{i-1}) \oplus p_{i-1}$. In the $i$th login, $U$ and $S$ communicate the following data.

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

$E$ eavesdrops the data of $(C_i, SEED \oplus D_i, H(D_i) \oplus p_{i-1})$ and extracts $D_i$ by $(SEED \oplus D_i) \oplus D_i$. Next, $E$ intercepts $(p_i \oplus D_i)$ and extracts $p_i$. Then, $E$ sends $p_i \oplus D_i \oplus X$ to $S$, where $X$ is an arbitrary number. After receiving $p_i \oplus D_i \oplus X$, $S$ rejects $U$. When $U$ is rejected, $U$ sends the login request again. Then, $S$ sends the following data to $U$ as a response.

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

$E$ intercepts those data, and extracts $D_i'$ by $(SEED \oplus D_i') \oplus SEED$. Next, $E$ sends $C_{i+1}, SEED \oplus D_i', H(D_i') \oplus p_i$ to $U$. After receiving them, it replies the following data.

$U \rightarrow S : p_{i+1} \oplus D_i'$

Similarly, $E$ intercepts it, and extracts $p_{i+1}$ by $(p_{i+1} \oplus D_i') \oplus D_i'$. Then, $E$ sends $p_{i+1} \oplus D_i'$ to $S$. After receiving the data, $U$ is authenticated by $S$. In the $(i+1)$th login, $E$ can impersonate $U$ to login $S$ as follows.

$E \leftarrow S : C_{i+1}, SEED \oplus D_{i+1}, H(D_{i+1}) \oplus p_i$
$E \rightarrow S : p_{i+1} \oplus D_{i+1}$

2.4 Ku et al. Attack

Recently, Ku et al. [10] also presented that Yeh-Shen-Hwang scheme is vulnerable to the stolen-verifier attack. Different from the attack of Tsuji et al., Ku et al. attack combines with stolen-verifier attack, guess attack and impersonate attack. The process is briefly described as follows.

Assume that $U$ completed the $i$th login, and $E$ has stolen $p_i$ from $S$ by the stolen verifier attack. Then, $E$ extracts $SEED$ by $(SEED \oplus D_i) \oplus (p_i \oplus D_i) \oplus p_i$. Since $p_i = H^{N-i}(K \oplus SEED)$, $E$ can guess a password/secret $K'$, and then computes $p_i' = H^{N-i}(K' \oplus SEED)$. If $p_i' = p_i$, $E$ knows that the guessing value $K'$ is a correct password. $E$ can produce $p_j$, where $i + 1 \leq j \leq N$, and impersonate as $U$ to login $S$ or impersonate as $S$ to cheat $U$.

3. Improvement

In this section, we propose an improvement on Yeh-Shen-Hwang scheme to resist the attack as stated above. The proposed scheme is described as follows.

3.1 Registration Stage

1. $U \leftarrow S : SEED$
2. $U \leftarrow S : N, H(SEED \oplus N) \oplus SK, H(SK)$
3. $U \rightarrow S : p_0 \oplus SK$

Initially, $S$ computes a value $SEED = H(ID \oplus x)$, and issues a smart card containing the pre-shared secret $SEED$ to $U$, where $ID$ is the identity of $U$ and $x$ is the secret of $S$. Then, $S$ selects a random number $D$ and a timestamp $T$, and then computes $SK = D\|T$. $S$ also decides a number $N$ which indicates the login number of $U$. Next, $S$ computes $H(SEED \oplus N) \oplus SK$ and sends $N, H(SEED \oplus N) \oplus SK, H(SK)$ to $U$. Once receiving those data, $U$ performs the XOR operation on $N$ and the $SEED$ which is stored in the smart card, and hashes $SEED \oplus N$ one time. $U$ extracts $SK$ by performing the XOR operation on $H(SEED \oplus N) \oplus SK$ and $H(SEED \oplus N)$. Then, $SK$ is hashed one time and compared with the $H(SK)$. If it matches, the identity of $S$ is authenticated. Then, $U$ computes $p_0 \oplus SK$, in which $p_0 = H^0(K \oplus SEED)$, and sends $p_0 \oplus SK$ to $S$. Once receiving $p_0 \oplus SK$, $S$ extracts $p_0$ by $(p_0 \oplus SK) \oplus SK$. Then, $S$ stores $p_0$ as a verifier for authenticating $U$.

3.2 Login Stage

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

For the $i$th 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 $C_i = N - i$. After receiving those values, $U$ extracts $SK_i$ by $(H(SEED \oplus C_i) \oplus SK_i) \oplus (H(SEED \oplus C_i))$, and checks the timestamp of the
session key $S K_i$. If the timestamp is valid, $U$ computes $p_{i-1}$, and then uses the computed result to extract $H(S K_i)$ from $H(S K_i) \oplus p_{i-1}$. Then, $S K_i$ is hashed one time and compared with $H(S K_i)$. If it is equivalent, the identity of $S$ is authenticated. Then, $U$ computes $p_i$ and $p_i \oplus S K_i$. Finally, $U$ sends $p_i \oplus S K_i$ to $S$.

3.3 Authentication Stage

Once receiving $p_i \oplus S K_i$, $S$ obtains $P_i$ by performing the XOR operation on $p_i \oplus S K_i$ and $S K_i$. Then, $p_i$ is hashed one time and compared with $p_{i-1}$. If it is equivalent, the identity of $U$ is authenticated. Finally, $S$ replaces $p_{i-1}$ and $C_{i-1}$ with $p_i$ and $C_i$ in the database.

4. Security Analysis

In this section, we discuss the security of our improved scheme. Considering Tsuji et al. attack and Ku et al. attack, an attacker may steal the verifier $p_{i-1}$ from $S$ after the $(i - 1)$th login and intercept the communication between $U$ and $S$. Since it is very difficult to derive $SEED$ by $p_{i-1}$ in the login stage, unless the attacker can reverse the one-way hash function. So, the attacker cannot forge a valid login request message in the $i$th login. Thus, our improved scheme can withstand the stolen-verifier attack.

Furthermore, in the Yeh-Shen-Hwang scheme, if $S$ provides the login services for $m$ users, $S$ must store $m$ pre-shared value $SEED$ in his/her database. The probability that $SEED$ is stolen is relatively high. The improved scheme computes $SEED_t$ by $SEED_t = H(ID_t \oplus x)$, where $1 \leq t \leq m$. $S$ only needs to keep $x$ in secret.

5. Conclusions

In this paper, a simple but useful improvement on Yeh-Shen-Hwang scheme is proposed, which can effectively withstand the stolen-verifier attack. Moreover, the improvement is as efficient as the original scheme.

Acknowledgement

We would like to thank anonymous reviewers for their comments to improve this paper. This research was supported in part by National Science Council, Taiwan, Republic of China, under the contract number NSC 93-2218-E-218-012.

References