Cryptanalysis of an Efficient Password Authentication Scheme

Chiu-Shu Pan
Department of Computer Science and Information Engineering,
Asia University, Taiwan

Shyh-Chang Tsaur
Department of Electronic Engineering,
National Chin-Yi University of Technology, Taiwan

Min-Shiang Hwang
Department of Computer Science and Information Engineering,
Asia University, Taiwan

Abstract—Recently, Thandra et al. proposed an efficient password authentication scheme. Their scheme is secure, efficient, and easy to implement. However, we find some weaknesses of their scheme in this article. We will show that their scheme is vulnerable to denial of service attacks, online and offline password guessing attacks, and impersonation attacks.

I. INTRODUCTION

One of methods to verify remote users over public Internet is a remote authentication scheme. There are three types of remote authentication schemes which have been used to verify the legitimacy of remote users. The first type is based on biometric [13], [22], [23]. The second type is based on the smart card [2], [4], [12], [15], [19], [21]. The third type is based on a password [7], [16], [24], [25]. Some authentication schemes with anonymity for wireless environments are also proposed [5], [11]. Some authentication schemes are used in Multi-server Environment [1], [3], [6], [10].

In 2016, Wei et al. proposed a remote user password authentication scheme [20]. Their scheme is simple and easy to implement. However, Tsai et al. shown that their scheme is vulnerable to denial of service attacks, password guessing attack, and privileged insider attack [18]. Recently, Liu et al. proposed an efficient and secure user authentication scheme with a smart card [9]. However, Liu, Tsai, and Hwang shown that their scheme was vulnerable to the replaying attack [8].

Recently, Ramasamy et al. proposed an efficient password authentication scheme for smart cards [14]. However, Thandra et al. showed that their scheme is vulnerable to privileged insider attacks, password guessing attack, and impersonation attack [17]. Thandra et al. also proposed a modified Ramasamy et al.’s scheme to resist the above flaws existing in Ramasamy et al.’s scheme. Thandra et al. claimed that their scheme could resist privileged insider attacks, password guessing attacks, user impersonation attacks, user masquerading attacks, and replay attacks. Furthermore, Thandra et al.’s is able to update user’s password and mutual authentication. They used BAN logic to analyze the formal security of their scheme. Their scheme is easy to implement. However, we find their scheme is vulnerable to denial of service attacks, online and offline password guessing attacks, and user impersonation attack.

The rest of this paper is organized as follows. In Section 2, we briefly review Thandra et al.’s password authentication scheme. In Section 3, we analyze and show that some security flaws exist in Thandra et al.’s password authentication scheme. Finally, we present our conclusions in Section 4.

II. REVIEW OF THANDRA-RAJAN-MURTY SCHEME

In this section, we briefly review Thandra et al.’s password authentication scheme (Thandra-Rajan-Murty Scheme) with smart cards [17]. There are two participants in Thandra-Rajan-Murty’s password authentication scheme: a server (S for short) and a user (U for short). The scheme consists of four phases, namely the registration phase, the login phase, the authentication phase, and the password update phase. The notations used in this article are listed in Table I.

A. The Registration Phase

In this phase, the server S makes a smart card for a new user (U_i). The registration phase is executed as follows:

1) The user U_i chooses his/her identity ID_i and password PW_i.
2) U_i computes h = H(PW_i), where H(·) denotes a one-way hash function.
3) U_i sends (ID_i, h) to the server S.
4) S computes CID_i, w_i, and V_i as follows:

\[ CID_i = H(ID_i || d), \]
\[ w_i = CID_i g^h \mod n, \]
\[ V_i = g^{CID_i \cdot h \cdot T_i} \mod n, \]

where d denotes a private key; \( (e, n) \) denotes a public key; \{d, (e, n)\} is a RSA key pair; g denotes a primitive
Inputs \( ID_i \) and \( PW_i \)

Computes \( h = H(PW_i) \)

\[
CID_i = w_i \cdot g^{-h} \mod n \\
W_i = (CID_i || r_1) \\
X_i = g^{CID_i \cdot h \cdot r_2} \mod n \\
Y_i = (W_i \cdot V_i^{r_2 \cdot T})^e \mod n
\]

\((ID_i, X_i, Y_i, T)\)

Computes \( CID_i = H(ID_i || d) \)

\[
L = X_i^{T \cdot T_r} \\
M = Y_i^d \cdot L^{-1} \mod n \\
R = M \mod 2^{k+1} \\
O = (CID_i || R) \\
R \overset{?}{=} O
\]

Acceptance of the login request

\(5)\ S \) delivers the smart card to \( U_i \) securely. The smart card contains eight parameters, \( \{ID_i, w_i, n, e, g, V_i, T_r, H(\cdot)\} \).

**B. The Login Phase**

In this phase, a user \((U_i)\) wants to login into the server \( S \) for obtaining some services, the user first attaches his/her smart card to a device reader and inputs his/her identity \( ID_i \) and password \( PW_i \). The login phase is executed in the following and illustrated in Figure 1.

\(1)\) The user \( U_i \) sends the login request parameters, his/her identity \( ID_i \) and password \( PW_i \) to the smart card.

\(2)\) The smart card computes \( h \) and \( CID_i \) as follows:

\[
h = H(PW_i) \\
CID_i = w_i g^{-h} \mod n.
\]

\(3)\) The smart card computes \( W_i, X_i, \) and \( Y_i \) as follows:

\[
W_i = (CID_i || r_1); \\
X_i = g^{CID_i \cdot h \cdot r_2} \mod n; \\
Y_i = (W_i \cdot V_i^{r_2 \cdot T})^e \mod n.
\]

Here \( r_1 \) and \( r_2 \) are two random number selected by the smart card; \( T \) denotes the current timestamp of the smart card.

\(4)\) The smart card sends \((ID_i, X_i, Y_i, T)\) to the server \( S \).

**C. The Authentication Phase**

Upon receiving the authentication request message \((ID_i, X_i, Y_i, T)\) from \( U_i \), the server \( S \) executes this authentication phase in the following and illustrated in Figure 1.

\(1)\) The server \( S \) checks whether \( ID_i \) format and the timestamp \( T \) in valid time or not. If one of conditions does not hold, the server \( S \) rejects the login request.

\(2)\) \( S \) computes \( CID_i, L, M, R, \) and \( O \) as follows:

\[
CID_i = H(ID_i || d) \\
L = X_i^{T \cdot T_r} \\
M = Y_i^d \cdot L^{-1} \mod n \\
R = M \mod 2^{k+1} \\
O = (CID_i || R)
\]

\(3)\) \( S \) compares whether \( M \) is equal to \( O \) or not. If this condition does not hold, the server rejects the login request.

\(4)\) \( S \) sends acceptance of the login request to the user if \( M = O \).

**III. CRYPTANALYSIS OF THANDRA-RAJAN-MURTY SCHEME**

In this section, we will analyze Thandra-Rajan-Murty’s password authentication scheme [17]. Thandra et al. claimed that their scheme is resistant to privileged insider attack,
password guessing attack, user impersonation attack, server masquerading attack, and replay attack. In this section, we show that Thandra-Rajan-Murty’s user password authentication scheme is vulnerable to denial of service attack, online and offline password guessing attack, and impersonation attack.

A. Denial of Service Attack

In Thandra-Rajan-Murty’s scheme, there are two steps needed to be checked for resisting the denial of service attack.

1) The server checks whether the user’s identity \( ID_i \) and the timestamp \( T \) are in correct format and in valid time in Step 1 of the authentication phase.

2) The server checks whether \( M \) is equal to \( O \) or not in Step 3 of the authentication phase. Here \( M \) is computed from Step 2 of the authentication phase.

Next, we show that Thandra-Rajan-Murty’s scheme is also vulnerable to the denial of service attack as Figure 2. The adversary may send the modified login request message \((ID_i, X'_i, Y'_i, T')\) to server, where \( X'_i \) and \( Y'_i \) are two random numbers; \( T' \) is the current timestamp. In this case, the server checks and passes the verification of the user’s identity \( ID_i \) and the timestamp \( T \) in Step 1 of the authentication phase. Thus, the server will continuously execute Step 2 of the authentication phase. Although the server will stop the login request after executing Step 3 of the authentication phase, the server will spend more time to compute \( CID_i, L, M, R, \) and \( O \) in Step 2 of the authentication phase. Therefore, the denial of service attack might result in the more computation load the server performs.

B. Online Password Guessing Attack

Although, the authors claimed that password guessing is not possible in their scheme [17], because the adversary cannot verify \( h \) using any of the known values without knowing the server’s secret key. However, we will show that their scheme was vulnerable to the online password guessing attack. The online password guessing attack is executed in the following and illustrated in Figure 3.

Suppose an adversary has stolen the user’s smart card. The adversary may guess the user’s password \( PW_i \) and then observes the communication between the server and the adversary. If the guessing password is correct, the server will send the acceptance of the login request to the user in Step 4 of the authentication phase. Otherwise, the server will terminate this session in Step 3 of the authentication phase. The adversary may guess the other passwords again and repeats to observe the communication between the server and the adversary. Therefore, Thandra-Rajan-Murty’s password authentication scheme is vulnerable to the online password guessing attack.

C. Offline Password Guessing Attack

Suppose an adversary has stolen the user’s smart card and extracted the parameters \((ID_i, w_i, n, e, g, V_i, T_r)\) from the smart card. The adversary may guess the user’s password \( PW'_i \), and then compares \( V_i \) whether is equal to \( g^{w_i \cdot h'_i \cdot T_r} \mod n \) or not, where \( h'_1 = g^{-h'} \) and \( h' = H(PW') \). If it is true, the guessing password is correct; otherwise, the password is incorrect. If the adversary has guessed the password \( PW'_i \),

\[
\begin{align*}
h' &= H(PW') \\
&= H(PW_i) \\
&= h.
\end{align*}
\]

Correctness of the above statement is expressed in the following:

\[
\begin{align*}
g^{w_i \cdot h'_i \cdot T_r} \mod n &= g^{CID_i \cdot g^{h' \cdot h' \cdot T_r}} \mod n \\
&= g^{CID_i \cdot h' \cdot T_r} \mod n \\
&= V_i.
\end{align*}
\]
Therefore, Thandra-Rajan-Murty’s user password authentication scheme is vulnerable to the offline password guessing attack.

D. Impersonation Attack

Although, the authors claimed that impersonation attack is not possible in their scheme [17], because the decrypted value of $Y$ by the server will not contain proper $CID_i$ for $ID_i$. However, we will show that their scheme was vulnerable to the impersonation attack. The impersonation attack is executed in the following and illustrated in Figure 4.

Suppose an adversary has stolen the user’s smart card and extracted the parameters $(ID_i, w_i, n, e, g, V_i, T_r)$ from the smart card and obtained the password by the offline password guessing attack in Subsection III-C. Then the adversary can impersonate a valid user $U_i$ to login in the server by executing the following steps.

1) The adversary computes $X'$ and $Y'$ as follows:

$$X' = (n - 1)^2(T_r)^{-1};$$
$$CID_i = w_i \cdot g^{-H(PW_i)};$$
$$W = (CID_i||B);$$
$$Y' = W^e(n - 1)^2.$$  

where $B$ is a $k$-bit number.

2) The adversary could impersonate the user $U_i$ and sends the login request, $(ID_i, X', Y', T)$ to the server.

Upon receiving the authentication request message $(ID_i, X', Y', T)$ from the adversary, the server $S$ executes this authentication phase as follows:

1) The server $S$ checks whether $ID_i$ format and the timestamp $T$ are in valid time or not. $ID_i$ and $T$ are selected by the adversary to meet the correct $ID_i$ and $T$ in valid time.

2) $S$ computes $CID_i, L, M, R,$ and $O$ as follows:

$$CID_i = H(ID_i||d);$$
$$L = X_i^{T \cdot T_r};$$
$$M = Y_i^{rd} \cdot L^{-1} \mod n;$$

$$= Y^{ed} \cdot L^{-1} \mod n;$$
$$= W^e(n - 1)^{2d}(n - 1)^{-2};$$
$$= W(n - 1)^{2(d - 1)} \mod n;$$
$$= (CID_i||B);$$
$$R = M \mod 2^{k+1};$$
$$O = (CID_i||B).$$

3) $S$ compares whether $M$ is equal to $O$ or not.

4) $S$ sends the acceptance of the login request to the adversary because of $M = O$.

Therefore, Thandra-Rajan-Murty’s user password authentication scheme is vulnerable to the impersonation attack.
IV. CONCLUSION

In this article, we have reviewed Thandra et al.’s password authentication scheme and cryptanalyzed its security. We showed that Thandra-Rajan-Murty’s password authentication scheme [17] cannot withstand the denial of service attack, online and offline password guessing attacks, and impersonation attacks.

ACKNOWLEDGMENT

This study was supported by the National Science Council of Taiwan under grant MOST 104-2221-E-468-004.

REFERENCES


