Security Analysis on a Timestamp-based Remote User Authentication Scheme

Zuowen Tan\textsuperscript{1,2}  
1. Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi Province, P.R. China  
2. Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong Higher Education Institutes, Guangzhou University, Guangzhou 510006, P.R.China  
tanziyw@gmail.com

Jianfeng Wang\textsuperscript{1,2}  
1. Hebei Normal University of Science & Technology, Qinhuangdao, China  
2. Information Security Center, Beijing University of Posts and Telecommunication, Beijing, China

Abstract—In recent years, many password-based remote user authentication schemes have been presented. In 2003, Shen et al. proposed a timestamp-based password authentication scheme using smart cards. In their scheme, the server does not need to maintain any verification table and only stores a secret key. However, Awasthi et al. found that Shen et al.’s scheme is vulnerable to impersonation attacks with the stolen card. Awasthi et al. proposed an improved remote user authentication scheme based smart cards. Unfortunately, the improved version is still insecure. We show that Awasthi et al.’s scheme is vulnerable to offline password guessing attacks, password compromise to the server, impersonation attack and important message leakage attacks. In addition, Awasthi et al.’s scheme has poor reparable.

Index Terms—Authentication, Smart Card, Timestamp, Impersonation Attacks, Password Guessing Attacks

I. INTRODUCTION

Smart card password-based authentication is one of the most convenient and commonly used approaches for the client-server environment over insecure channel [1, 2]. After the user has completed the registration with the server, smart card stores the registration information in it. When a user holding the smart card inputs the password, the smart card generates login message and sends it to the server. The server validates the identity of the user. Then the server sends back the response message to the smart card. The smart card authenticates the server. Thus, the remote user confirms that the communicating party is the server which he/she wants to login to, while the server applies the method to verify if the communicating user is a legal registered client. That is, not only can the server verify the user, but a user can verify the server [3].


So far, there are many various smart card password-based remote user authentication schemes [7,8] using verification tables to enhance the security. However, these password-based remote user authentication schemes using verification tables are vulnerable to stolen verification table attacks. Many smart card password-based remote user authentication schemes without verification tables have been proposed [9-12]. However, some of them have later been found to be insecure [13,14].

In 1999, Yang et al. [15] proposed a timestamp-based password user authentication scheme. In Yang et al.’s scheme, no verification table is required. Moreover, users can update the passwords freely. Unfortunately, the scheme is vulnerable to the forged login attacks [16]. Shen et al. improved Yang et al.’s scheme to remove the security flaws. However, recently Awasthi et al. [17] found that Shen et al.’s scheme is vulnerable to some deadly attacks. If an adversary has stolen smart card, the adversary can impersonate the card holder to login to the server. Awasthi et al. also proposed an improved timestamp-based password user authentication scheme. In their scheme, remote server needs no verification information for the registered users. They claim that their scheme is more secure than the original scheme. The analysis shows that their scheme does not meet the two-factor security.

The rest of the paper is organized as follows. In
Section 2, we list the desirable properties for smart card based password mutual authentication scheme. In Section 3, we review Awasthi et al.’s scheme. The detailed cryptanalysis on Awasthi et al.’s scheme [17] is given in Section 4. Finally, conclusion will be given in Section 5.

II. PROPERTIES AND TWO-FACTOR SECURITY MODEL

Typically, a smart card password-based remote user authentication scheme consists of four phases, registration, login, authentication and password change phase. For each phase, there are individual security requirements and desirable properties [18,19]. In the following, we list them.

- Mutual Authentication. After run of the protocol, the server is sure that the remote user is a valid registered client. The user is sure that the communicating party is that server which the client wants to login with.
- Server Knowing No Password. Even the server cannot have any information about the registered users’ password.
- Freedom of Password Chang. A user can freely change his password without any interaction with the server. The server is totally unaware of the change of the user’s password.
- Two-factor Security. For the security of smart card password-based remote user authentication scheme, the adversary is modeled as follows [20]:

(a) The adversary has full control over the communication channel between the users and the server during the login and authentication phase. The adversary may intercept and obtain all the messages transmitted in the channel. The adversary may delete or modify any message transmitted in the channel. The adversary even may insert message. In addition, maybe an adversary is a malicious registered user. Thus the adversary could login the server as a legitimate user.

(b) The adversary either can extract the information by stealing the smart card or can obtain a user’s password. As Kocher et al. [21] and Messerges et al. [22] showed, the information stored in the smart card can be extracted by monitoring the power consumption. The adversary cannot do both. But the adversary is not allowed to compromise the server.

III REVIEW ON AWASTHI ET AL.’S SCHEME

In this section, we briefly review Awasthi et al.’s scheme [17]. Their scheme consists of the initialization phase, the registration phase, login phase and authentication phase. The login and authentication phases are also shown in Figure 1. Now, we list notations used throughout the paper in Table 1.

| $U_i$ | user |
| $CID_i$ | card identifier |
| $ID_i$ | identity of $U_i$ |
| $n$ | integer |
| $d$ | private key |
| $e$ | public key |
| $g$ | primitive element |
| $f()$ | one way function |
| $p$ | large primes |
| $q$ | large primes |
| $pw_i$ | password of $U_i$ |
| $T_s$ | timestamp of server |
| $T_c$ | timestamp of user |

A. Initialization Phase

The Key Information Center (KIC) generates the system parameters, computes user’s secret information and issues the smart cards to users.

1. Generate two large primes $p$ and $q$, and compute $n=pq$.
2. Choose a prime $e$ and an integer $d$, such that $e \cdot d \mod \Phi(n) = 1$,
   where $e$ is the system’s public key and $d$ is the corresponding private key provided to the server.
3. Find a primitive element $g$ of both $GF(p)$ and $GF(q)$ and select a one way function $f()$. $g$ and $f()$ are part of the system public parameters.

B. Registration Phase

A user $U_i$ submits his identifier $ID_i$ and password $pw_i$ to the KIC through a secure channel. The KIC performs the following steps:

1. Generate the smart card’s identifier $CID_i$ and compute $h_i$
   $$CID_i = f(ID_i \oplus f())$$, $h_i = g^{md} \mod n$.
2. Compute the user’s secret information
   $$S_i = CID_i \mod n$$.
3. Write $\{n, e.g., ID_i, S_i, h_i\}$ into a smart card and issue the card to $U_i$ through a secure channel.

C. Login Phase

The user logs in to the server using his smart card by the following steps:

1. The user inserts his smart card to a device reader and inputs his identity $ID$ and password.
2. The smart card chooses a random integer $r$, and calculate
   $$X_i = g^{pw_i} \mod n, Y_i = S_i \cdot h_i^{f(ID_i)} \mod n$$, where $T_i$ is the timestamp at the login device.
3. The smart card sends the login request message $M = \{ID_i, X_i, Y_i, n, e, g, T_i\}$ to the remote server $S$ over the open channel.

D. Authentication Phase

Upon receiving the login request message $M$, the server performs the following steps to verify it:

1. Verify if $ID_i$ is a valid identifier. If it is not so, then $S$ rejects the login request.
2. Check the validity of $T_i$. If $(T_i - T_c) \geq \Delta T$, then the server rejects the login request, where $T_c$ is the current timestamp at the remote server and $\Delta T$ is expected legitimate time interval for transmission delay.
(3) Compute $CID_i = f(ID_i \oplus d)$.
(4) Check if the following equation holds
$$Y'_i = CID_i X'_i \mod n.$$  
If it holds, $S$ accepts the login request.

(5) Compute $R = f(ID_i, T'_s)^y \mod n$ and send
$M' = \{R, T'_s\}$ to the user $U_i$, where $T'_s$ is the timestamp on the remote server.
After receiving the message $M'$ from the remote server $S$, $U_i$ authenticates the server as follows.

(6) Check the validity of $T'_s$. If $(T'_e - T'_s) \geq \Delta T$,
then the user rejects the server, where $T'_e$ is the current timestamp when the user received the response message.

(7) Compute $R = R' \mod n$. If $R = f(ID_i, T'_e)$ holds, $U_i$ authenticates $S$. Otherwise, $U_i$ refuses the response message.

### Fig. 1. Login-Authentication Phase

<table>
<thead>
<tr>
<th>User $U$</th>
<th>Server $S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose $r_i$</td>
<td>$X_i = g^{pn_i} \mod n$</td>
</tr>
<tr>
<td>$Y_i = S_i \cdot h_i^{f(ID_i, T)} \mod n$</td>
<td>${ID_i, X_i, Y_i, n, e, g, T_i}$</td>
</tr>
<tr>
<td>$R = R' \mod n$</td>
<td>${R, T'_s}$</td>
</tr>
<tr>
<td>Check: $(T'_e - T'_s) \leq \Delta T$</td>
<td>$R = f(ID_i, T'_e)$</td>
</tr>
<tr>
<td>Accept the login request</td>
<td>Accept the response message</td>
</tr>
</tbody>
</table>

### IV CRYPTANALYSIS ON AWASTHI ET AL.’S SCHEME

Assume that Eve is an adversary as described in Section 2. We will show that Awasthi et al.’s scheme is vulnerable to offline password guessing attacks, password compromise to the server, impersonation attack and important message leakage attacks. In addition, Awasthi et al.’s scheme has poor reparability.

#### A. Offline Password Guessing Attack

For password-based authentication schemes, the password guessing attack is one of the basic menaces which must be considered. This is because the client tends to choose a short and easy-to-remember password. However, the convenience of these human easy-to-remember passwords may potentially suffer from password guessing attacks. An adversary would try to guess the client’s password and then impersonate the client. The password guessing attacks can be classified into two categories: online password guessing attacks and offline password guessing attacks. If the adversary guesses the client’s password and checks their validity online, the type of attacks is called online password guessing attacks. If the adversary first intercepts some login message or even extracts the message stored in the smart cards and then guesses the password and verifies the guesses through the known password-related message, the type of attacks is called offline password guessing attacks (also known as offline dictionary attack).

For Awasthi et al.’s scheme, an adversary could mount offline password guessing attacks by performing the following operations.

Assume that $U_i$’s smart card is compromised by an adversary Eve. The adversary Eve first extracts the message $\{n, e, g, ID_i, h_i\}$ stored in the smart card by monitoring the power consumption [21] or by analyzing the leaked information [22]. From the description of registration phase, note that $h_i = g^{pn_i} \mod n$. Thus, the adversary would guess the passwords and verify if the guess is right by checking the equation: $h'_i = g^{pn_i} \mod n$. If the above equation holds, Eve has mounted password
guessing attacks successfully. If the above equation does not hold, the adversary repeats the guess-then-verify process until the adversary finds the correct password. Since the attack is launched offline and the password space is small, the password guessing is always completed in a short time.

Therefore, Awasthi et al.’s scheme cannot resist the offline password guessing attack.

B. Password Compromised to the Server

As described in Section 3, when a user needs to register with the server, the user sends password \(pw_i\) to the server. Then the server obtains the user’s password. It is undesirable, which violates the security property “Server Knowing No Password” mentioned in Section 2.

C. Impersonation Attack

Suppose that the user’s smart card is lost, an adversary Eve who holds the lost smart card can obtain the message \(\{n, e, g, ID_i, S_i, h_i\}\). Thus, Eve can impersonate the user \(U_i\) to login to the server by performing the procedure.

1. Compute \(h = h_i^e \mod n\).
2. Choose a random integer \(r_i\) and calculate \(X_i = h_i^r \mod n\), \(Y_i = S_i \cdot h_i^{\epsilon f(\delta T_c)} \mod n\), where \(T_c\) is the timestamp at the login.
3. Send the login request message \(M = \{ID_i, X_i, Y_i, n, e, g, T_c\}\) to the remote server \(S\).

Upon receiving the login request message, the server will believe that the login message is from the user with an identity \(ID_i\). Since \(h = h_i^e \mod n\), then \(Y_i = S_i \cdot h_i^{\epsilon f(\delta T_c)} = S_i \cdot g^{\epsilon f(\delta T_c) n \cdot p w_i}\). Since \(h = h_i^e \mod n\), we have \(Y_i = S_i \cdot h_i^{\epsilon f(\delta T_c)} \mod n\). Thus, the verification equation holds:

\[
Y_i = (S_i \cdot h_i^{\epsilon f(\delta T_c)})^e = S_i^e \cdot h_i^{\epsilon f(\delta T_c)} \mod n.
\]

Therefore, Awasthi et al.’s scheme cannot resist the impersonation attack.

D. Important Message Leakage Attack

Suppose that the important message \(S_i\) of the user \(U_i\) is compromised to an adversary Eve. Without knowledge of password \(pw_i\), Eve could impersonate the user \(U_i\) to issue the login message.

Case 1: Impersonation using any password

We further suppose that Eve is a malicious client. If it is not so, Eve can register with the server and becomes a registered user. With knowing \(S_i\), Eve uses any password and can impersonate the user \(U_i\) to login to the server by performing the procedure.

1. Use his own password \(pw_i\) and the message \(h_i\) stored in his own smart card and compute \(t = h_i^{\epsilon p w_i} = g^t \mod n\).
2. Choose a random integer \(r_i\) and randomly chooses \(pw_i\) as \(U_i\)’s password. Then Eve computes

\[
Y_i = (S_i \cdot h_i^{\epsilon f(\delta T_c)})^e = S_i^e \cdot h_i^{\epsilon f(\delta T_c)} \mod n = C I D X_i^{\epsilon f(\delta T_c)} \mod n.
\]

Thus, the server believes that the login request
message is sent by the user $U_i$.

E. Poor Reparability

Suppose that the user has found or suspected that the adversary has performed some above described attacks. The user may abandon his smart card and re-register with the server.

The user will choose a new identity $ID_i$ and a new password $PW_{i}$. However, it is also impractical to change identity, which is always banded with the user in the real world according to the identity-based cryptography. Therefore, the new registration will have the same identity as the previous smart card. If the identity of the new registration keeps unchanged, the secret information $S_i$ still keeps the same. However, as shown in the above subsection, the compromise of secret $S_i$ will lead to important message leakage attacks against the new registration. In other words, the adversary with the information of the used smart card can still mount impersonation attacks against the new smart card.

V. CONCLUSION

In this paper, we have analyzed that Awasthi et al.’s scheme suffers from offline password guessing attacks, impersonation attack and important message leakage attack. Additionally, we have shown that Awasthi et al.’s scheme lacks the property: Server Knowing No Password. Moreover, Awasthi et al.’s scheme provides poor reparability. How to design the smart card password-based remote user authentication schemes with strong security in the formal model is interesting and useful. In addition, identity-based smart card password-based remote user authentication protocols with a tighter security reduction without random oracles are practical and challenging. This will be our future research.

ACKNOWLEDGMENT

The author would like to thank the reviewers for their useful suggestions. This work is supported by the National Science Foundation of China under Grant No.61163053, Natural Science Foundation of Jiangxi Province (20122BAB201035), the Open Project Program of Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong Higher Education Institutes, Guangzhou University (No. 2012-02-02-01) and Foundation of Jiangxi Educational Committee under Grant GJJ13301.

REFERENCES


He is currently a professor at Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance & Economics. He has published over 60 papers on information security in international conferences and journals. His current research interests include e-commerce security, information security and cryptography.

Dr. Tan is the committee members of some international conferences and reviewers on international Journals.