User Authentication Scheme with Privacy-Preservation for Multi-Server Environment

Ren-Chiu Wang, Wen-Shenq Juang, and Chin-Laung Lei, Member, IEEE

Abstract—New user authentication schemes for multipleservers environment were proposed by Liao-Wang and Tsai. In their schemes, application servers do not need to maintain a verification table and this admired merit is not addressed by previous scholarship. Besides, the privacy of users is also addressed in Liao-Wang’s scheme. In this article, we show that their schemes are not secure against the server spoofing and the impersonation attacks. Then we propose a robust user authentication scheme to withstand these attacks and keep the same merits.

Index Terms—Authentication, multi-server architecture, password, privacy, smart card.

I. INTRODUCTION

M ANY smart card-based user authentication with key agreement schemes for multiple-servers were proposed to keep the following merits [1]: (1) single registration; (2) user friendly; (3) preventing the replay, the password guessing without smart cards, the impersonation and the stolen-verifier attacks [2]; (4) keeping free from the serious time synchronization problem; (5) preventing the servers from impersonating other servers to cheat users or from masquerading some users to obtain the services of other servers.

Recently, Tsai [3] proposed a user authentication scheme for multi-server architecture. In his scheme, the application servers do not need to maintain a verification table and the computation cost is low. At the same time, Liao and Wang [4] proposed a user authentication to meet the same merits. Besides, in their scheme, the identities of users will not be traced by any users (we call it Merit 6).

In this article, we show that those schemes still suffer from the server spoofing and the impersonation attacks. Then we propose a user identification scheme to keep the privacy of users and the above merits at the same time for multi-server environment. Also, in our scheme, even if the secret information stored in a smart card is compromised [5], the off-line password guessing attack is still not successful (we call it Merit 7).

II. REVIEW OF TWO AUTHENTICATION SCHEMES

A. Notations

$U_i$ and $S_j$ denote the ith user and the jth server; $RC$ and $SC$ denote the registration center and the smart card;

B. Liao-Wang’s scheme [4]

In the registration phase, $U_i$ submits $ID_i$ and $SID_i$ to $RC$. Then $RC$ calculates $T_i = h(ID_i \parallel x_{RC})$, $H_i = h(T_i)$, $V_i = T_i \oplus h(ID_i \parallel PW_i)$ and $B_i = h(PW_i) \oplus h(x_{RC})$, and writes $(V_i, B_i, H_i, h(\cdot), y)$ into $SC$. $RC$ sends $SC$ back to $U_i$ via a secure channel. We use Figure 1 to introduce the other phases.

C. Tsai’s scheme [3]

In Tsai’s scheme, the computation cost is low since only one-way hash operations are required and the application servers do not need to maintain a verification table.

In the registration phase, $U_i$ submits $(ID_i, PW_i)$ to $RC$. $RC$ calculates $R_i = h(ID_i \parallel x_{RC})$ and $C_0 = h(PW_i) \oplus R_i$ and writes $(C_0, h(\cdot))$ into $SC$. $RC$ sends $SC$ back to $U_i$ via a secure channel.
A. Liao-Wang’s scheme

**Server spoofing attack:** When a valid user \( U_i \) wants to obtain the service of \( S_j \), other valid users \( U_{i+1} \) can impersonate \( S_j \) to cheat \( U_i \) after intercepting the login request \( \{CID_i, P_{ij}, Q_{ij}, N_i\} \). We demonstrate the attack in Figure 3.

**Impersonation attack:** After User \( U_i \) logs into Server \( S_j \), \( S_j \) can impersonate this user to obtain the services from other servers \( S_{j+1} \). We demonstrate the attack in Figure 4.

B. Tsai’s scheme

**Server spoofing attack:** A valid malicious server \( S_{j+1} \) stands in the middle of \( U_i \) and \( S_j \). When \( U_i \) logs into \( S_j, S_{j+1} \) can imitate \( S_j \) to communicate with \( U_i \). We demonstrate the attack in Figure 5. **Impersonation attack:** After \( U_i \) logs into \( S_j, S_{j+1} \) can imitate this user to get the services from other servers \( S_{j+1} \) without the knowledge of the secret random number \( N_i \). We demonstrate the attack in Figure 6.

III. SECURITY ANALYSIS OF TWO AUTHENTICATION SCHEMES

**A. Liao-Wang’s scheme**

**Mutual Verification and Session Key Agreement Phase**

- **(1)CID, P_{ij}, Q_{ij}**
- **(2)Get( , h(PW)|R)_{ij} \Rightarrow h(PW) \oplus h(x_{ij})**

**Impersonation Attack**

- **(1)Replay CID (2ID_i = P_i \oplus h(y_i \| \| \| N_i \| \| SD_i))**
- **(2)Get( , h(PW) \oplus h(x_{ij}))**

**Mutual Verification and Session Key Agreement Phase**

- **(1)CID, P_{ij}, Q_{ij}**
- **(2)Get( , h(PW)|R)_{ij} \Rightarrow h(PW) \oplus h(x_{ij})**

**Impersonation Attack**

- **(1)Replay CID (2ID_i = P_i \oplus h(y_i \| \| \| N_i \| \| SD_i))**
- **(2)Get( , h(PW) \oplus h(x_{ij}))**

**Mutual Verification and Session Key Agreement Phase**

- **(1)CID, P_{ij}, Q_{ij}**
- **(2)Get( , h(PW)|R)_{ij} \Rightarrow h(PW) \oplus h(x_{ij})**

**Impersonation Attack**

- **(1)Replay CID (2ID_i = P_i \oplus h(y_i \| \| \| N_i \| \| SD_i))**
- **(2)Get( , h(PW) \oplus h(x_{ij}))**

**Mutual Verification and Session Key Agreement Phase**

- **(1)CID, P_{ij}, Q_{ij}**
- **(2)Get( , h(PW)|R)_{ij} \Rightarrow h(PW) \oplus h(x_{ij})**

**Impersonation Attack**

- **(1)Replay CID (2ID_i = P_i \oplus h(y_i \| \| \| N_i \| \| SD_i))**
- **(2)Get( , h(PW) \oplus h(x_{ij}))**
We analyze that well-known security threats cannot work in our scheme and compare the satisfaction of the merits with Liao-Wang [4] and Tsai’s schemes [3] in Table 1.

**Server spoofing attack:** (1) No valid malicious $S_j$ can imitate users $U_i$ to create a new request without knowing $SK_k$. (2) Malicious $S_{j+1}$ replays the used request $Res_i$ to $S_j$ and will get $S_j$’s response ($Res_{s2y}$, $Res_{SCS}$). $S_{j+1}$ cannot derive $h(N_j)$ and send the response $Res_i$ without $R_{si}$. (3) After $U_i$ logs into $S_j$, $S_j$ cannot employ the used $Req_i$ to launch any well-known attacks. (a) If $S_j$ replays $Res_i$ to other servers $S_{j+1}$, $RC$ will deny it since $SID_{j+1}$ is not included in $Req_i$. (b) $S_{j+1}$ cannot imitate $S_j$ to cheat $U_i$ since he does not have the capability to forge the response ($Res_{s2y}$, $Res_{SCS}$) from the tapped messages ($Res_{RC1}$, $Res_{RC2}$, $Res_{RC3}$, $Res_{RC4}$) without knowing ($SK_k$, $N_j$). **Impersonation attack:** No adversary can imitate $U_i$ to pass the verification of $RC$ without the secret key $SK_k$, and then to obtain services of the application servers $S_j$. **Off-line password guessing attack with smart cards:** Assume that the adversary has tapped the communication channel and can extract the information ($ID_i$, $SK_k'$, $h(n)$) stored in the smart card. Then the adversary guesses a candidate password $h(PW_i')$ and extracts $SK''_k = h(PW_i') + SK_k'$. The adversary will try to verify $SK''_k$ using the tapped messages ($Req_i$, $Res_i$, $Res_{RC1}$, $Res_{RC2}$, $Res_{RC3}$, $Res_{RC4}$, $Res_{s2y}$, $Res_{SCS}$) in local computer. If he can do that, the secret key $SK_k$ is known and the password $PW_i$ is guessed. (1) Under the concept of the quadratic residue, since the adversary does not know the secret key $(p, q)$ of the system, the adversary cannot derive the content of $Req_i$. If the adversary wants to verify if the guessed $SK''_k$ is correct, the adversary must guess the high entropy random number $N_i$. The adversary cannot correctly guess $N_i$ in a polynomial time. (2) Since $SK_k$ is unknown to the adversary and $NN_i$ and $Nh_i$ are high entropy random numbers, the adversary is also hard to guess or to derive them from ($Req_i$, $Res_{RC1}$, $Res_{s2y}$, $Res_{SCS}$) in a polynomial time. (3) Based on the same reason, without the knowledge of $SK_k$, $N_i$, $NR_i$ and $Nh_i$, the adversary is still hard to verify if the guessed $SK''_k$ is valid from ($Res_{RC2}$, $Res_{RC3}$, $Res_{RC4}$, $Res_{s2y}$).

### Table I

**Comparisons of satisfaction of the criteria**

<table>
<thead>
<tr>
<th><strong>Criteria</strong></th>
<th>Our</th>
<th>Liao-Wang</th>
<th>Tsai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single registration</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>User friendly</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Preventing the various attack$^1$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No serious time synchronization problem</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Preventing the server’s impersonation attack$^2$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>User anonymity</td>
<td>Yes</td>
<td>Yes</td>
<td>No$^*$</td>
</tr>
<tr>
<td>Off-line password guessing attack with smart cards</td>
<td>Yes</td>
<td>No$^*$</td>
<td>No$^*$</td>
</tr>
</tbody>
</table>

$^*$: Even if the scheme follows our idea, the scheme still suffers from the off-line password guessing attack. 1: The attacks include the replay, the password guessing without smart cards, the impersonation and the stolen-verifier attacks. 2: Preventing the servers from impersonating other servers to cheat users or from masquerading some users to obtain the services of other servers.

---

**References**