An Authentication and Key Agreement Scheme with Key Confirmation and Privacy-preservation for Multi-server Environments

Zuowen Tan
School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi Province, China
tanzyw@gmail.com

Abstract—In the internet environment, it is desirable for a user to login different servers by keying the same password and using the same smart card. This paper proposes an authentication and key agreement scheme with key confirmation for multi-server environments. Compared with the previous authentication and key agreement schemes for multi-server environments, the new scheme holds many merits. It satisfies the following properties: R1. Single registration; R2. User friendly; R3. Prevention of the replay, the password guessing without smart cards, the impersonation and the stolen-verifier attacks; R4. Resistance against server spoofing; R5. Mutual authentication; R6. Two-factor authentication; R7. Resistance against known-key attacks; R8. Perfect forward secrecy; R9. Scalability of login; R10. Anonymity of users.

Index Terms—Password, Authentication, Key agreement, Multi-server.

I. INTRODUCTION

In the internet environment, a user can access many resources which are distributed in many different places. Lamport introduces a password authentication scheme [1] for a singer server environment. However, in the password-based authentication schemes, the server often needs to store a verification table, which easily incurs the stolen-verification attacks [2]. Moreover, the password-based authentication schemes always suffer from password guessing attacks. Applying a smart card to password based authentication schemes in a singer server is a convenient way [3-6]. If one extends the password smart card based authentication methods in a single server environment to the multi-server environment, it is infeasible. Since each user needs to have different passwords with different remote servers in the multi-server environment.

Some smart card based password authentication schemes for the multi-server environment are proposed [7,8,13,14,15]. But the scheme in [7] does not fit for smart card, since one high computation cost method, the neural networks, is used. Many schemes are insecure [8,14,16-20]. For example, the scheme [13] cannot resist against impersonation attacks, and server spoofing attacks. Furthermore, it fails to provide mutual authentication [14,17,18]. Hsiang and Shih proposed a secure dynamic ID based remote user authentication scheme for multi-server environment [17]. Tan [19] demonstrated that Hsiang and Shih's scheme [17] is vulnerable to off-line password guessing attacks, impersonation attacks, server spoofing attacks. And it cannot resist against information leakage attacks. Chen, Huang and Chou [14] proposed an improvement on Hsiang and Shih's scheme. However, Tan showed that Chen et al.'s scheme [14] still suffers from impersonation attacks, password guessing attacks and server spoofing attacks. Moreover, Chen et al.'s scheme [14] does not have perfect forward security. Wang et al. proposed an authentication and key agreement scheme for multi-server environment [18]. Unfortunately, the scheme [18] is insecure against the server spoofing. In addition, Wang et al.'s scheme [18] does not hold perfect forward security.

Based on [18] and [21], we highlight the requirements of smart card based authentication and key agreement schemes with privacy-preservation for multi-server environments: R1. Single registration: A user is required to register once and can login to the eligible servers many times; R2. User friendly: A user can change his password freely; R3. Prevention of the replay, the password guessing without smart cards, the impersonation and the stolen-verifier attacks [20]; R4. Resistance against server spoofing: A server cannot impersonate other servers to cheat users or masquerade some users to cheat other servers; R5. Mutual authentication: The scheme can provide the mutual authentication between the servers and users; R6. Two-factor authentication: When only a user's smart card or the password is comprised, the scheme still can prevent the adversary from masquerading as the user and the password guessing; R7. Resistance against known-key attacks; R8. Perfect forward secrecy: Even if long secret keys are compromised, the previous session keys should not be revealed; R9. Scalability of login: After a user has finished the first time login to a certain service provider server, any interaction with the registration center is not necessarily required when the user logins to the same service provider server once again; R10. Anonymity of users.

Of the above requirements, the requirement (R9) is essential. If an authentication and key agreement scheme for multi-server environments cannot provide the scalability of login, when many logins happen at the
same time, the bottleneck will be caused. Since the registration center is required to engage in every login. It is desirable that the user logs in the same service provider server not the first time without the participation of the registration center. However, it is necessary that the service provider server can control login of the user to avoid the abuse of login.

The remainder of this paper is organized as follows. In Section II, we proposed an authentication and key agreement scheme with key confirmation and privacy-preservation for multi-server environments. In Section III, we analyze the security of the proposed scheme and compare its performance with the previous authentication and key agreement schemes for multi-server environments. Finally, we conclude the paper in Section IV.

II. OUR PROPOSED SCHEME

In this section, we propose a new authentication and key agreement scheme with key confirmation and privacy-preservation for multi-server environments. The following notations in Table 1 will be used throughout the paper.

An authentication and key agreement scheme for multi-server environments is involved with three parties, a user set, a server (as a service provider) set and a control server (as the registration center) $RC$. In our scheme, $RC$ holds two master keys, $x$ and $y$. The public system parameters include a large field $F_p$ of prime order $p$ with $p > 2^{160}$, a base point $G$ of order $n$ in the elliptic curve $E_p$, a secure hash function $h(\cdot)$ which maps any string into an element of $Z_n$ and a secure hash function $H(\cdot)$ which maps any string into an element of the elliptic curve group.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$id_i$</td>
<td>the identity of a user $U_i$</td>
</tr>
<tr>
<td>$ind_i$</td>
<td>the blind identity of a user $U_i$</td>
</tr>
<tr>
<td>$sid_i$</td>
<td>the identity of a service provider $S_j$</td>
</tr>
<tr>
<td>$RC$</td>
<td>the registration center</td>
</tr>
<tr>
<td>$cid$</td>
<td>the identity of $RC$</td>
</tr>
<tr>
<td>$pw_i$</td>
<td>the password of a user $U_i$</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>$s, y$</td>
<td>string concatenation operation</td>
</tr>
<tr>
<td>$E_p$</td>
<td>$RC$’s master keys</td>
</tr>
<tr>
<td>$h(\cdot), H(\cdot)$</td>
<td>collision-resistant hash functions</td>
</tr>
<tr>
<td>$p$</td>
<td>a large prime number</td>
</tr>
<tr>
<td>$E_p$</td>
<td>an elliptic curve over field $F_p$</td>
</tr>
<tr>
<td>$G$</td>
<td>a base point of the elliptic curve $E_p$</td>
</tr>
<tr>
<td>$E_{p}\bigoplus$</td>
<td>a symmetric encryption with key $K$</td>
</tr>
<tr>
<td>$E_{p}\bigotimes$</td>
<td>a symmetric decryption with key $K$</td>
</tr>
</tbody>
</table>

Our scheme consists of four phases: the registration phase, the login phase, the authentication and key agreement phase, and the password change phase.

A. Registration Phase

The user $U_i$ and a service provider $S_j$ execute the following operation to complete the registration, respectively.

The service provider $S_j$ sends the identity $sid_i$ to the registration center $RC$. After $RC$ receives $sid_i$, $RC$ calculates $c_j = h(sid_i||y)G$ and sends the secret information to $S_j$ through a secure channel.

In order to provide the privacy of the user $U_i$, the user $U_i$ with a real identity $id_i$ chooses an indicator $ind_i$ as his blind identity.

1. $U_i$ sends the indicator $ind_i$ to $RC$.
2. Upon receiving $ind_i$, $RC$ computes $B_i = h(ind_i||x)G$.
3. Then $RC$ writes $(ind_i, B_i, G, E_p)$ into a smart card and issues it to the user $U_i$ through a secure channel.
4. After $U_i$ receives the smart card, $U_i$ inserts the card into a card reader and inputs a password $pw_i$ and his real identity $id_i$. The smart card computes $B_i = B_i \bigoplus h(id_i||pw_i)G$. The smart card replaces $B_i$ with $B_i$.

In the following, we describe the other phases in two cases: Case 1, for the first-time-login; Case 2, for not-the-first-time-login.

B. Case 1: For the-first-time-login

B.1 Login Phase

When $U_i$ wants to access the resources of the server $S_j$ for the first time, $U_i$ inserts his smart card and keys his identity $id_i$ and password $pw_i$. Then the smart card, the server $S_j$ and the registration center $RC$ execute the following steps to achieve mutual authentication and agree on a session key between the smart card and the server $S_j$. The login phase for Case 1 is depicted as in Fig.1.

1. The smart card computes $B_i = B_i \bigoplus h(id_i||pw_i)G$.
2. The smart card chooses randomly $r_i \in Z_n$. Then the smart card computes $R_i = r_iG$, $X_{i1} = h(sid_i||r_iB_i)$, $X_{i2} = h(sid_i||B_i) \bigoplus R_i$.
3. The smart card sends $(ind_i, X_{i1}, X_{i2})$ to $S_j$.

B.2 Authentication and Key Agreement Phase

The phase can be subdivided into three sub-phases. We depict it in Fig.2, Fig.3 and Fig.4, respectively.

B.2.1 Authentication Server and User Phase

1. After the server $S_j$ receives $(ind_i, X_{i1}, X_{i2})$, $S_j$ chooses randomly $t_j \in Z_n$. Then $S_j$ computes $T_j = t_jG$, $Y_{j1} = h(sid_i||[ind_i||X_{i1}||X_{i2}||t_jc_j]) \bigoplus Y_{j2} = H(sid_i||ind_i||c_j) \bigoplus T_j$. Finally, the server sends $(ind_i, X_{i1}, X_{i2}, sid_j, Y_{j1}, Y_{j2})$ to $RC$.
2. $RC$ computes $T_j^* = H(sid_i||[ind_i||h(sid_i||y)G]) \bigoplus Y_{j2}$ and checks if the following holds: $Y_{j1} = h(sid_i||[ind_i||X_{i1}||X_{i2}||h(sid_i||y)T_j^*])$. If the equation holds, $RC$ continues the operation. Otherwise, $RC$ abandons it.
3. $RC$ computes $R_i^* = H(sid_i||h(ind_i||x)G) \bigotimes X_{i2}$ and checks if the following holds:
If the equation holds, $RC$ continues the operation. Otherwise, $RC$ abandons it.

Figure 1. Login Phase

(1) $T_j = t_jG$

\[ Y_{j1} = h(sid_j || ind_j || x_1 || t_j c_j) \]
\[ Y_{j2} = h(sid_j || [c_j] \oplus T_j) \]

(2) $ind_v, x_{12}, sid_v, y_{12}$

\[ ind_v = h(sid_j || [c_j] \oplus T_j) \]
\[ x_{12} = h(sid_j || x_1) \oplus T_j \]
\[ x_{12} = h(sid_j || [x_1] |x_2) \oplus T_j \]
\[ x_{12} = h(sid_j || x_1) \oplus T_j \]

Figure 2. Authentication Server and User Phase

(3) $T'_j = h(sid_j || [c_j] \oplus T_j)$

\[ V_{j1} = h(sid_j || [c_j] \oplus T_j) \]
\[ V_{j2} = h(V_{1j} || [c_j] \oplus T_j) \]

(4) $Z_{j2} = h(Z_{j1} \oplus T_j)$

\[ (5) V_{j2} = h(V_{j1} \oplus T_j) \]

Figure 3. Authentication Registration Center Phase

(1) $K = h(t_j R_i)$
\[ h_1 = h(ind_j || [sid_j] || [x] |x_2) \]
\[ h_2 = h(sid_j || [ind_j] \oplus T_j) \]
\[ A_{ij} = E_k(h_1 || h_2) \]

(2) $A_{ij}, D$

\[ (3) K = h(r_i T_j) \]
\[ D_k(A_{ij}) = h_1^* || h_2^* \]
\[ h_1^* = h(sid_j || [ind_j] || [x] |x_2) \]
\[ h_2^* = h(sid_j || [ind_j] || [x]) \]
\[ Store : (D, h_1^* \oplus h(id_j || [pwj])) \]
\[ A_{ij} = h(sid_j || [h_1^*] || D || K + 1) \]

(4) $A_{ij}$

Figure 4. Key Confirmation Phase

B.2.2 Authentication Registration Center Phase

1. $RC$ computes

\[ Z_{j1} = h(ind_j || [sid_j] || [x] |x_2) \oplus T_j \]
\[ Z_{j2} = h(Z_{j1} \oplus T_j) \]

Then $RC$ sends $(Z_{j1}, Z_{j2})$ to the server $S_j$.

2. $RC$ calculates

\[ V_{j1} = h(sid_j || [ind_j] \oplus T_j) \]
\[ V_{j2} = h(V_{1j} || [c_j] \oplus T_j) \]

Next, $RC$ sends $(V_{1j}, V_{j2})$ to the smart card.

3. After the server $S_j$ receives $(Z_{j1}, Z_{j2})$, $S_j$ checks whether the following equation holds:

\[ Z_{j2} = h(Z_{j1} \oplus T_j) \]

If the above holds, then $S_j$ computes

\[ R_i = h(ind_j || [c_j] \oplus T_j) \]

4. After the smart card receives $(V_{1j}, V_{j2})$, the smart card first checks whether the following equation holds:

\[ V_{j2} = h(V_{j1} || [c_j] \oplus T_j) \]

If the above holds, then the smart card computes

\[ T_j = h(sid_j || [x] |x_2) \oplus T_j \]

B.2.3 Key Confirmation Phase

After the smart card and the server have authenticated the registration, in essence, they have also authenticated each other. Next, the two parties will compute the session key and confirm it through the following steps.

1. $S_j$ computes the session key $K = h(t_j R_i)$ and sets a validation date $D$ in which the user can login to the server without the participation of the registration center. Then $S_j$ computes

\[ h_1 = h(ind_j || [sid_j] || [x] |x_2) \]
\[ h_2 = h(sid_j || [ind_j] || [x]) \]
\[ A_{ij} = E_k(h_1 || h_2) \]

Finally, $S_j$ sends $(A_{ij}, D)$ to the user $U_i$.

2. Upon receiving the message $(A_{ij}, D)$, the smart
card calculates \( K = h(r_iT_j) \) and decrypts \( A_{ij} \).

Then the smart card parses the plaintext into two parts \( h_1^* \) and \( h_2^* \) and checks whether the following equation holds

\[
(3) \quad A_{ij} = h(h_1^*|K) \quad \text{and sends} \quad A_{ij} \quad \text{to the server} \quad S_j.
\]

4. \( S_j \) validates \( A_{ij} \) by checking if it satisfies

\[
(4) \quad A_{ij} = h(h_1^*|K)\quad \text{and} \quad A_{ij} = h(h_1^*|K + 1).
\]

C. Case 2: For not-the-first-time-login

After the user stores \((D, h_2^* \oplus h(id_i||pw_i))\) to the smart card after the first time of login, \( U_i \) can obtain the service of the server \( S_j \) without the participation of RC. The login phase and authentication and key agreement phase are executed only between the user \( U_i \) and the server \( S_j \).

C.1. Login phase

When \( U_i \) wants to access the resources of the server \( S_j \), not the first time, \( U_i \) inserts his smart card and keys his identity \( id_i \) and password \( pw_i \). The login phase for Case 2 is depicted as in Fig.5.

1. The smart card computes

\[
(5) \quad h_1^* = (h_1^* \oplus h(id_i||pw_i)) \quad \text{for} \quad R_i \in H.\quad \text{Then} \quad R_i = r_iG,
\]

2. The smart card chooses randomly \( r_i \in Z_n \). Then the smart card computes

\[
X_{i1} = h(sid_i||r_ih_1^*G)\quad \text{and} \quad X_{i2} = H(sid_i||h(id_i)+R_i).
\]

3. The smart card sends \((D, ind_i, X_{i1}, X_{i2})\) to \( S_j \).

C.2. Authentication and key agreement phase

The phase can be subdivided into two sub-phases which are also depicted in Fig.6 and Fig.7, respectively.

C.2.1 Authentication User Phase

1. \( S_j \) computes

\[
R_i = H(sid_i||h(id_i)||sid_i||D||c_j) \quad \text{and} \quad X_{i2}.
\]

2. \( S_j \) checks if the following equation holds:

\[
X_{i1} =? h(sid_i||h(id_i)||sid_i||D||c_j)R_i.
\]

If it holds, \( S_j \) authenticates the user.

C.2.2 Authentication Server and Key Confirmation Phase

1. The server \( S_j \) chooses randomly \( T_j \in Z_n \) and computes \( T_j = t_jG \), and \( K = h(t_jR_i) \).

2. \( S_j \) computes

\[
V_{ij} = H(X_{i1}||X_{i2}||h(T_j)||R_i) \quad \text{and} \quad A_{ij} = h(V_{ij}||K).
\]

Then \( S_j \) sends \((A_{ij}, V_{ij})\) to the user \( U_i \).

3. Upon receiving the message \((A_{ij}, V_{ij})\), the smart card calculates

\[
(6) \quad A_{ji} = h(A_{ij}|h(T_j)|K + 1).
\]

4. Finally, the smart card issues \( A_{ji} \) to \( S_j \).

5. \( S_j \) confirms the session key by checking

\[
(7) \quad A_{ji} =? h(A_{ij}|h(id_i)||sid_i||D||c_jT_j||K + 1).
\]

D. Password Change Phase

If \( U_i \) wants to change his password, \( U_i \) inserts his smart card into a card reader and keys his identity \( id_i \), old password \( pw_i \) and new password \( pw_{\text{new}} \). Then the smart card computes

\[
B = B_i \oplus h(id_i||pw_i)\quad \text{and} \quad B_{\text{new}} = B_i \oplus h(id_i||pw_{\text{new}}).\quad \text{Finally, the smart card replaces} \quad B_i \quad \text{with} \quad B_{\text{new}}.
\]

III. ANALYSIS ON THE PROPOSED SCHEME

In the following, we demonstrate that the proposed scheme satisfies all the requirements of smart card based authentication and key agreement schemes for multi-server environments, which is listed in Section 1.

Theorem 1. Upon the assumption of the difficulty of the CDH problems, there is no any adversary who could
mount the user impersonation attacks.

**Proof.** Suppose that an adversary $A$ could impersonate a user $U_i$ to generate login message $(\text{ind}_i, X_{i1}, X_{i2})$ for the-first-time-login or $(\text{ind}_i, D, X_{i1}, X_{i2})$ for not-the-first-time-login and key confirmation message $A_{ji}$. Then, the login message must satisfy

$$X_{i1} = h(\text{sid}_i || h(\text{ind}_i || x) R_{i1}^*)$$

and

$$X_{i2} = h(\text{sid}_i || r_i h_i^* G).$$

Since $A$ has no knowledge of $x$ or $y$, $A$ computes $h(\text{ind}_i || x)$ or $h(\text{ind}_i || \text{sid}_i || D || c_i)$ only by guessing the hash value. By $q_h$, we denote the total number of the hash query. The probability of $A$’s success is $\frac{q_h}{2^n}$ or $A$ generates $h_1^*$ and $(K + 1)$. Without the knowledge of $y$, $A$ obtains $h_1^*$ with the probability at most $\frac{q_h}{2^n}$. If $A$ generates $K$, $A$ could solve an instance $(R_i, T_j, r_i T_j)$ of CDH problems with the probability at least $\delta - \frac{q_h}{2^n}$. □

**Theorem 2.** Upon the assumption of the difficulty of the CDH problems, there is no adversary who could mount the server impersonation attacks.

**Proof.** Suppose that an adversary $A$ impersonates a remote server generating $(\text{ind}_i, X_{i1}, X_{i2}, \text{sid}_j, Y_{j2}, Y_{j2})$ and $(A_{ji}, D)$ for the first-time-login or $(A_{ji}, V_{ji})$ for not-the-first-time-login to cheat a user $U_j$ or $RC$ with a non-negligible probability $\epsilon$.

Under that random oracle model, it implies that $A$ could compute $h(\text{sid}_j || y) T_j$ and $K = h(r_i T_j)$ with a non-negligible probability. The former shows that $A$ knows the value $y$, which contradicts that $y$ is secret. The latter demonstrates that $A$ can compute $r_i T_j$ from $(R_i, T_j)$. That is, $A$ solves an instance of CDH problems with a non-negligible probability $\epsilon - \frac{q_h}{2^n}$. □

The theorem implies that the proposed scheme can resist against the server spoofing.

**Theorem 3.** Upon the assumption of the difficulty of the CDH problem under the random oracle model, the proposed scheme achieves mutual authentication between the user and the registration center, between the user and the server, and between the server and the registration center.

**Proof.** Theorem 1 and Theorem 2 imply that the server and the user are authenticated. Now, we show that the registration center will be authenticated by the server and the user, respectively.

The proposed scheme can prevent an attacker from obtaining the registration center’s secret keys $(x, y)$ or $(h(\text{ind}_i || x) h(\text{sid}_i || y))$. During the authentication, $h(\text{ind}_i || x) G$ and $h(\text{sid}_i || y) G$ are hashed into some values. One will be faced with a discrete logarithm problem if he wants to obtain $h(\text{ind}_i || x) (h(\text{sid}_i || y) G)$ or $(h(\text{sid}_i || y) G)$. In addition, $(x, y)$ are protected in a hashed form.

Suppose that an adversary $A$ impersonates a registration center to issue $(Z_{j1}, Z_{j2})$ to the server $S$ and $(V_{i1}, V_{i2})$ to the user with a non-negligible probability $\epsilon$. There must be the following equations:

$$Z_{j2} = h(Z_{j1} || c_i d || \text{sid}_i || h(\text{sid}_i || y) T_j),$$

$$V_{i2} = h(V_{i1} || c_i d || \text{sid}_i || h(\text{ind}_i || x) R_{i1}).$$

Under the random oracle model, $A$ could obtain $h(\text{sid}_i || y) G$ or $h(\text{ind}_i || x) G$ with a non-negligible probability $\epsilon - \frac{q_h}{2^n}$. □

**Theorem 4.** The proposed scheme achieves two-factor authentication.

**Proof.** Assume that an adversary obtains the smart card and extracts the information $(\text{ind}_i, B_i, G, E_i)$ or $(D, h_1^* \oplus h(\text{ind}_i || p w_1))$ stored in the card.

If the adversary has no password of the device owner, the adversary cannot compute $B_i$ or $h_1^*$, the former of which is used to compute the first-time login message and the latter of which is used to compute the not-first-time login message.

The proposed scheme can resist against the undetectable on-line password guessing attack. Since $RC$ can authenticate the user by checking the validity of the information $(X_{i1}, X_{i2})$. The password is not contained in the verification equation $X_{i1} = h(\text{sid}_i || h(\text{ind}_i || x) R_{i1}^*)$. In addition, the off-line password guessing attacks will also fail in our proposed scheme. Since the password is protected in the smart card as $B_i = B_i \oplus h(\text{ind}_i || p w_1) G$.

**Theorem 3** proves that $B_{ci}$ is protected. $h_1^*$ is shared only by the user $U_i$ and the server $U_j$. In the proposed scheme, the identity $id_i$ is anonymous. Therefore, the proposed scheme can resist offline password guessing attacks. Our scheme holds two-factor security. □

**Theorem 5.** Upon the assumption of the difficulty of the CDH problem under the random oracle model, the proposed scheme provides the perfect forward secrecy.

**Proof.** Assume that an adversary $A$ corrupts the registration center and obtains master keys $x$ and $y$. The adversary can recover $T_j$ and $R_i$ after $A$ has intercepted the transmitted messages. However, the adversary will have to be faced with an instance of CDH problems, when $A$ tries to compute $r_i T_j$ or $t_j R_i$ from $(T_j, R_i)$. □
of the not-first-time-login, \( R_i \) is included in \( X_{12} \), while \( T_j \) is included \( V_j \). Only those ones who know \( h_i \) can obtain them. So the adversary cannot calculate the new session key. Theorem 5 demonstrates that even the registration center could not mount the known-key attacks.

Therefore, the proposed scheme can provide known-key security. \( \square \)

In addition, we briefly review the security properties:

1. **Resistance to Replay Attacks**
   In the proposed scheme, the freshness of the messages transmitted and the shared session key are provided by the random nonces \( R_i \) and \( T_j \). Theorem 5 Theorem 6 shows that only the user, the server, and the registration center can recover and verify the random nonces \( R_i \) and \( T_j \). Therefore, the proposed scheme can resist replay attacks.

2. **Resistance to Stolen Verifier Attacks**
   From Section 2, we know that the proposed scheme does not need to maintain a verification table or password table in the servers or the registration center. Therefore, nobody obtain any verifiable information from the servers and \( RC \). The proposed scheme can resist against the stolen verifier attacks.

### TABLE II

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Ours [12]</th>
<th>[13]</th>
<th>[14]</th>
<th>[17]</th>
<th>[18]</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R4</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>R5</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R6</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R7</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R8</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>R9</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>R10</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R11</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Note: R3 [12-14,17] suffer from password guessing attacks and impersonation attacks. R11 refers to the function: key confirmation.

3. **Scalability of Login**
   In the proposed scheme, after a user \( U_i \) has finished the login to a server \( S_j \) once, \( U_i \) stores \( (D, h_1 \oplus h(id_i||pw_i)) \) in the smart card. When \( U_i \) logsins to the server \( S_j \) before the expiration \( D, U_i \) sends an initiation message \( (D, \text{ind}_i, X_{11}, X_{12}) \) to \( S_j \). Then \( S_j \) uses \( h_1 \) to authenticate \( U_i \) straightly without interaction with the registration center.

From Section II, one can find easily that our proposed scheme satisfies the requirements R1, R2 and R9 mentioned in Section I. We compare the satisfaction of some merits with the previous smart card based authentication and key agreement schemes [12-14,17,18] for multi-server environments in Table II.

### IV. CONCLUSION

Authentication and key agreement for multi-server environments is useful in the network. In the paper, we proposed a smart card based authentication and key agreement scheme for multi-server environments. The proposed scheme removes the weaknesses of the authentication and key agreement schemes for multi-server environments in the literature. Security analyses demonstrate that the proposed protocol can withstand various possible attacks and satisfies all the security requirements. The functionality comparison confirms the advantages of our scheme in contrast to the previous authentication and key agreement schemes for multi-server environments. Especially, our scheme can provide the session key confirmation and privacy preservation. Thus, our scheme can provide anonymity and protection of the login users.

### ACKNOWLEDGMENT

This work is partially supported by a grant from the National Natural Science Foundation of China (10961013), the Science Research Fund of Jiangxi Province Educational Department (GJJ11418) and the National Natural Science Foundation of Jiangxi Province (2010GZS0047). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

### REFERENCES


Zuowen Tan, born in 1967. He received his M.S. degree in Fundamental Mathematics from Xiangtan University in 2002 and his Ph.D. degree in Applied Mathematics from Institute of Systems Science, Academy of Mathematics and System Science, CAS in 2005. He is currently an associate professor at Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance & Economics. He has published over 50 papers on information security in international conferences and journals. His current research interests include information security and cryptography.

Dr. Tan was a committee member of some international conferences, a reviewer of some international journals.