A Novel Smart Card-based Remote User Authentication Mechanism

Deborah Uwera, Dongho Won*

Department of Computer Engineering, Sungkyunkwan University, South Korea

Abstract  The past few years have seen a rapid progress of multi-user computing environments. Numerous security mechanisms have therefore been employed in a bid to ensure that sensitive information in computer systems does not get destroyed, copied or even altered by unauthorized users. Remote users attempting to login into a particular system would therefore have to authenticate themselves to the server and vice versa. This paper proposes a novel remote user authentication scheme using smart cards. Our scheme endeavors to be an efficient yet secure scheme, hence we chose to use only one-way hash functions and XOR operations, in order to avoid computationally complex operations. We also conducted a security analysis on our scheme to ensure that it is secure against possible known attacks.

Keywords  Security, Authentication, Key Exchange

1. Introduction

The main reason why schemes for user authentication are necessary is because a server needs be enabled to remotely verify the legitimacy of a user trying to login, and ensure that this user is indeed genuine and trustworthy. The user on the other hand also needs to ensure that the server is truly genuine, hence they both require to authenticate each other; Mutual authentication.

In the past decade, there has been a recent influx of various remote user authentication schemes that use biometrics, passwords and smart cards. Many of these schemes however have a number of limitations that render them inadequate to be used as authentication schemes. Cryptanalysis shows us that many of them are at risk of attacks and security breaches.

Remote user authentication scheme is a very suitable authentication scheme to deal with private data that is being transmitted over insecure channels. Since Security and privacy have proven to be one of the most important factors in today's real time applications, users are therefore required to have the proper access rights in order to be able to access resources at remote system in client/server-based service architecture, which are widely used. In this type of architecture we find that a single computer can handle a huge amount of clients who are dispersed all over the world. In daily routines, there too are many real-time applications that also require user authentication such as, e-banking, e-commerce, physical access control to computer resources. Based on various comprehensive surveys on password-based remote user authentication schemes, we see that most of the remote user authentication schemes that are password-based are impractical due to the fact that they are either very expensive in terms of computation or are susceptible to different security attacks. In [1], Das et.al proposes a dynamic ID and password-based remote user authentication scheme that uses smart cards, and incorporates the use of hash function and XOR operations. Ever since, numerous researchers have proposed improved authentication protocols. This was done in order to eliminate the weaknesses in the previous authentication protocols such as [2], [3], and [4]. These weaknesses are dealt with in the following studies; according to [5] we see a dynamic ID-based authentication scheme that has key agreement using symmetric cryptology. This scheme endeavors to deal with the security flaws and weaknesses of [2]. They incorporated a Session key in order to create a more secure channel for communication. In [6] Li et al. assert that their scheme resisted masquerading attacks and avoided the leaking of information. However, [7], pointed out that [6] was not entirely secure, since it leaked partial information about the communication party’s secret parameters and any attacker would be able to access the leaked information to deduce session keys .In [8], a secure remote user authentication scheme was proposed that is also password-based was introduced. However, their scheme uses Elliptic curve cryptography and hash functions. Due to this fact, their scheme is too costly and thus not feasible.

In our paper, we propose a Novel Smart Card-Based remote user authentication scheme using XOR operations and hash functions. The rest of this paper is organized as follows. In Section 2, we propose our new and secure Smart card-based remote user authentication scheme. In Section 3, we conduct a security analysis where we perform a security analysis of our proposed scheme. In Section 4, we have the conclusion of our paper.
2. Our Proposed Scheme

In this part, we introduce our Remote User Authentication scheme that is based on Smart Cards.

<table>
<thead>
<tr>
<th>Table 1. Our Scheme’s Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notations</strong></td>
</tr>
<tr>
<td>U&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>S&lt;sub&gt;j&lt;/sub&gt;</td>
</tr>
<tr>
<td>ID&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>PW&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>Xs</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>h(·)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>⊕</td>
</tr>
</tbody>
</table>

2.2 Proposed Scheme’s Phases

In this scheme, we have three phases;

i. The registration phase,

ii. The login phase,

iii. The authentication phase.

The registration phase is the phase where the user U<sub>i</sub> first registers to the server in order to gain access to services from the remote server S<sub>j</sub>. After the registering occurs, the server S<sub>j</sub> then issues a smart card that contains detailed parameters stored in the smart card’s memory.

The login phase, is the phase where we see that whenever the user U<sub>i</sub> needs to gain access to the services from the server S<sub>j</sub>, the user U<sub>i</sub> is required to input his/her identity and password in order to login to the server, while also using the smart card issued to them by the registration server.

The authentication phase is the phase where mutual authentication occurs between the server and the user; the server S<sub>j</sub> authenticates the user U<sub>i</sub> and the user U<sub>i</sub> also authenticates the server S<sub>j</sub>. After mutual authentication between U<sub>i</sub> and S<sub>j</sub>, both U<sub>i</sub> and S<sub>j</sub> establish a secret session key shared between them so that they communicate securely using that established key in future.

2.2.1 Registration Phase.

These are the steps found in this phase.

**Step 1.** First, the user U<sub>i</sub> selects his/her own secret identity ID<sub>i</sub> and then chooses a strong password PW<sub>i</sub>

**Step 2.** U<sub>i</sub> then generates a secret number ‘K’ randomly, and makes sure to keep it secret to everyone else except from U<sub>i</sub> themselves.

**Step 3.** The user U<sub>i</sub> then uses the secret number generated ‘K’ in order to mask the password using, PW<sub>i</sub> as RPW<sub>i</sub> = h(K || PW<sub>i</sub>). And stores ‘K’ in the memory of the smart card and then proceeds to send the registration request message (ID<sub>i</sub>,RPW<sub>i</sub>) to the registration remote server S<sub>j</sub> via a secure channel.

**Step 4.** After the server S<sub>j</sub> receives the registration request message from the user U<sub>i</sub>, the server S<sub>j</sub> then generates a secret number Xs randomly, which is kept secret to S<sub>j</sub> only.

**Step 5.** S<sub>j</sub> then computes Z<sub>1</sub>= h(ID<sub>i</sub>‖RPW<sub>i</sub>) and Z<sub>2</sub>= h(ID<sub>i</sub>‖Xs) ⊕ Z<sub>1</sub>. Furthermore, S<sub>j</sub> computes AID<sub>i</sub>= ID<sub>i</sub>⊕h(ID<sub>i</sub>‖Z<sub>2</sub>‖T<sub>s</sub>). Here the AID is incorporated in order to achieve user anonymity, and it is a temporary identity for the user U<sub>i</sub>, which is used rather than the permanent identity ID<sub>i</sub>.

**Step 6.** In the Final step, S<sub>j</sub> issues the smart card SC which contains the information (AID, h(·), Z<sub>1</sub>) and sends it to the user U<sub>i</sub> via a secure channel.

---

**Scheme Phase 1: Registration**

**USER (U<sub>i</sub>)**

- ID<sub>i</sub>, PW<sub>i</sub>
- Generates ‘K’
- Compute RPW<sub>i</sub> = h(K||PW<sub>i</sub>)

**SERVER (S<sub>j</sub>)**

- (ID<sub>i</sub>, RPW<sub>i</sub>) via secure channel

- Generates a secret number Xs
- Compute Z<sub>1</sub>= h(ID<sub>i</sub>||RPW<sub>i</sub>)
- Z<sub>2</sub>= h(ID<sub>i</sub>||Xs) ⊕ Z<sub>1</sub>
- Computes AID<sub>i</sub>=ID<sub>i</sub>⊕h(ID<sub>i</sub>||Z<sub>2</sub>||T<sub>s</sub>)

---

**Smart card**

(AID, h(·), Z<sub>1</sub>)
2.2.2. Login Phase

The following steps are executed in this phase.

**Step 1.** The user $U_i$ first inserts his/her smart card SC into a card reader then. $U_i$ inputs his/her identity ID$_i$ and password PW$_i$.

**Step 2.** Then Smart Card SC computes the masked password $RPW'_i$ as $RPW'_i = h(K || PW'_i)$ using the secret number $K'$ stored in the memory of the smart card memory. SC then computes $Z'_i = h(ID_i || RPW'_i)$ and checks if the condition $Z'_i = Z_i$ holds. If this condition holds, then $U_i$ is able to pass the password verification step and the next step is executed. Otherwise, this phase has to be terminated immediately.

**Step 3.** The smart card SC computes $J_1 = Z_1 \oplus Z_2$ where these parameters are embedded within the smart card. And then the Smart Card generates a random nonce $R_{sc}$ and proceeds to compute $J_2 = (Z_1 \oplus Z_2) \oplus T_{sc}$ where $T_{sc}$ is the current system timestamp. Finally, SC sends the login request message $(J_1, J_2, T_{sc})$ to the server $S_j$ using a public channel.

2.2.3. Authentication Phase

These are the steps found in this phase.

**Step 1.** The server $S_j$ checks the validity of the timestamp $T_{sc}$ in the received message by the condition $(T_{sc} - T'_s) < \Delta t$, where $T'_s$ is the current system timestamp of $S_j$. If this condition is satisfied, $S_j$ computes $J_s = h(ID_j \parallel X_s)$, where $X_s$ is the secret number of the server. $S_j$ then verifies if $J_s = J_1$. If it does not hold, Server $S_j$ rejects the login request message and this phase terminates immediately. After that $S_j$ computes $J_s = J_1 \oplus h(ID_j) \parallel T_{sc} = h(ID_j \parallel R_{sc} \parallel T_s) \parallel h(ID_j) \parallel T_{sc} = R_{sc}$.

**Step 2.** Then the server $S_j$ generates a random nonce $R_s$ and then computes $J_6 = J_4 \oplus R_s \parallel T_s$ where $T_s$ is the current system time stamp of the server $S_j$. Then $S_j$ sends the authentication request message $(J_7, J_6, T_s, AID'_{new})$ to the user $U_i$ via a public channel.

**Step 3.** After receiving the authentication request message, Smart Card SC checks the validity of the timestamp $T_s$ in the received message with the condition $(T_s - T'_s) < \Delta t$, where $T'_s$ is the current system timestamp of SC. If this condition does not hold, the phase terminates immediately. Otherwise, SC computes $J_8 = J_5 \parallel J_7 \parallel T_{sc} = (Z_1 \parallel Z_2 \parallel R_{sc} \parallel T_s) \parallel (Z_1 \parallel Z_2) \parallel T_{sc}$, thus $J_s = R_s$. Then the Smart card SC further computes $J_9 = h(J_8 \parallel T_s \parallel R_{sc} \parallel T_{sc})$.

**Step 4.** SC then computes $J_{10} = J_9 \parallel AID'_{new}$. $S_j$ then verifies that this holds, if not, the procedures are terminated. Otherwise the Smart card computes a secret session key shared between $U_i$ and $S_j$: $SK = h(R_s \parallel T_s \parallel R_{sc} \parallel T_{sc} \parallel AID'_{new})$. Thus, after successful authentication, both $U_i$ and $S_j$ can communicate securely using the established secret session key.
3. Security Analysis of the Proposed Scheme

In this section, we first show that our scheme is secure against various known attacks.

3.1. Impersonation Attack

In this kind of attack, an adversary attempts to impersonate the remote server \( S_j \) or a legal user \( U_i \). If an attacker intercepts the login request message \( \langle J_1, J_3, T_{SC} \rangle \) during the login phase and wants to start a new session, the attacker has to modify both \( J_1 \) and \( J_3 \). However, in order to change \( J_3 \) the attacker has to know both \( I_{ID_i} \) and \( R_{sc} \), which are unknown to the attacker.

3.2. Stolen Smart Card Attack

In this kind of attack, we assume that the card \( SC_i \) is lost or stolen by an attacker.

The Attacker can then be able to extract all the information contained \( \langle AID, h(\cdot), Z_1, Z_2 \rangle \) in the smart card \( SC_i \) of the user \( U_i \) with the use of power analysis attack. However, the attacker still has no way to find out the secret information \( X_s \) of the server, therefore, since \( Z_z = h(ID_i \parallel X_s) \oplus Z_2 \) and \( AID = ID_i \oplus h(ID_i \parallel Z_z \parallel T_sc) \), this is not helpful to the attacker.

3.3. Password Guessing Attack

In this attack, we consider that the smart card \( SC_i \) of a legal user \( U_i \) is lost or stolen by an attacker. All the secret information \( \langle AID, h(\cdot), Z_1, Z_2 \rangle \) stored in the memory of the smart card \( SC_i \) is known to the attacker. Still then the attacker is not able to guess correctly the password \( PW_i \) of \( U_i \). In addition, suppose the attacker intercepts all the transmitted messages \( \langle J_1, J_3, T_{SC} \rangle \) during the login phase and \( \langle J_7, J_6, T_{S}, AID_{new} \rangle \) during the authentication phase.

None of these messages involves the password \( PW_i \) of the user \( U_i \), therefore still the attacker is unable to carry out the password guessing attack.

4. Conclusions

We have made an analysis of existing schemes for remote user authentication, and identified that they either have vulnerabilities to various known attacks or have a large cost for computation, this paper therefore proposes a novel password-based remote user authentication scheme using smart cards. Our scheme ensures efficiency and security, while upholding simplicity with the use of only one-way hash function and XOR operations. We have also conducted a security analysis on our scheme to ensure that it is secure against possible known attacks. Hence, we propose that our new scheme is both feasible and secure, making it an ideal remote user authentication scheme.

Acknowledgements

This research was supported by the Basic Science Research Program through the National Research
REFERENCES


