
Qingqi Pei  
State Key Laboratory of Integrated Service Network, Xidian University, Xi’an 710071, P.R. China  
Email: qqpei@mail.xidian.edu.cn

Jie Yu  
State Key Laboratory of Integrated Service Network, Xidian University, Xi’an 710071, P.R. China  
Email: yujie8830@gmail.com

Abstract—In the current Internet world, most of the Internet services are based on the single server model and use the password identity authentication to provide application service for the users, this means that the user must enter the identity and password, before his/her wants to login in the application service server. It is extremely hard for user to remember the different ID and password, so the single sign-on (SSO) system has been proposed to solve this problem. There many Authentication protocol proposed for the SSO system. In this paper, we first introduced the SSO system and expounded the importance of the authentication protocol in the SSO system. Then we researched on some authentication protocols which can be used in the SSO system, but there are some serious secure problems in their schemes. So we propose a secure dynamic identify based Single Sign-On authentication protocol using smart card. Our protocol can resist several kinds of attacks, such as replay attack, impersonation attack, stolen smart card attack, leak-of-verifier attack and can provide user’s anonymity. In our proposed protocol, it removes the aforementioned weaknesses of their protocols and only uses the one-way hash functions and XOR operations which make the protocol very effectively.

Index Terms—Single sign-on, Authentication, Dynamic identity, Smart card, Password

I. INTRODUCTION

With the development of the Internet and information technology, different kinds of service systems are provided via the Internet, such as online shopping, online games, electronic commerce, etc. Every system has its own security policy to authenticate the identity of remote users, the most familiar and sample authenticate mechanism is the password authentication which asking user to import his or her legal identity (ID) and password to access a service. Most of the existing password authentication protocols for these service systems are based on single-server model. With the increasing of the servers use the same one authentication protocol, it removes the aforementioned weaknesses of their protocols and only uses the one-way hash functions and XOR operations which make the protocol very effectively.

The SSO technology is a secure network access technology for the multi-server architecture. The user only has one active authentication in the SSO system, and then he/she can have all the authorization of the services in the whole SSO system. The essence of the SSO system is that users access to the entrance of a group application programs via a related authentication protocol and only need to login to the system once. In the SSO system, all the application servers use the same one authentication protocol to improve the system security strategy. So a secure authentication protocol is very important for the SSO system is very important.

In this paper, we first researched on some authentication protocols which can be used in the SSO system in section II. Then we provide a review of the SSO authentication architecture with smart card which used in this paper, in section III. In section IV, the secure dynamic identify based single sign-on authentication protocol using smart card we proposed is introduced. We discuss the security analysis of the proposed protocol and the comparison of the cost and functionality of the proposed protocol with other related protocols in section V. Finally, here is a conclusion in section VI.

II. RELATED WORKS

Since Lamport first proposed password based authentication protocol [1] in 1981, there are many different kinds of password based authentication protocols. Most of the proposed authentication protocols are based on single server architecture. But because of the popularization of the single server applications, the users feel very inconvenient to remember different IDs and passwords, and it is very complicated for the management of the servers. The SSO system is appeared in this time.

In 2000, The Novell Company taked the lead in publishing two kinds of the SSO application -- Novell(r) Single Sign-on Bundle and NDS Authentication Services 3.0, which provide good supports in Windows NT and 2000, Linux, Solaris, OS/390, NetWare, HP-UX, AIX, Free BDS, Radius, Internet Information Server, etc. The e-commerce CA also published a SSO solution -- sTrust Single Sign-on (SSO) 6.5 for electronic commerce. Tivoli Global Sign-on which is provided by IBM, used a center
server to collect all the login information in the system, and returned an application authorization list for the users after they login to the system successfully. All these solutions of the SSO system is the multi-server architecture based authentication protocol.

In 2004, Juang [9] proposed a multi-server architecture authentication protocol using smart card. In this protocol, it used the symmetric encryption algorithm and did not maintain any identity authentication lists among the servers. In the same year, Chang and Lee [10] improved Juang’s protocol and proposed a similar smart card based multi-server architecture authentication protocol. Chang and Lee’s protocol is much more efficient than Juang’s. In 2007, Hu et al.’s [11] proposed an effective password authentication key agreement protocol. This protocol is used in the multi-server architecture, and users can use a smart card and a weak password to access multi servers, the user shared a common secret session key among each server. This proposed protocol is much more efficient and user friendly than the protocol proposed by Chang and Lee (2004).

The most famous authentication in SSO system – the Kerberos [12] protocol also has some limits, like all the servers are exposed for users and two servers are both used to authenticate users. In 2006, Yang et al. [13] proposed an authentication and key exchange protocol which is similar to the Kerberos protocol. There are also two servers in the system used to authenticate users in the protocol, but only one front-end server is exposed to users directly and one control server dose not communicate with users. The methods of sharing keys and using two servers to authenticate users make attackers must to compromise two servers so that they can use the offline dictionary attack successfully. In the same year, Mackenzie et al [14] proposed a key exchange authentication protocol based on password, in which there are a group of servers and public keys used to authenticate. But the use of public key makes this protocol computation intensive. Tsai [15] proposed a smart card used authentication protocol based on multi-server architecture in 2008. Because this protocol used one-way hash function as its mathematics foundation, there is no need for servers and registration center to store any authentication form. In this protocol, there is no symmetric and asymmetric encryption algorithm, so it is much more efficient than any other protocol which is introduced above.

However, all the multi-server architecture based password authentication protocol introduced in our paper are based on static ID, it is means that there is a chance for attacker to pretend a legal user. In 2009, Liao and Wang [16] proposed a dynamic identity based remote user authentication protocol using smart card. This protocol only uses one-way hash function to implement a strong authentication and provides a secure method to update the user’s password without the help of trusted third party. But in the same year Hsiang and Shih found that Liao and Wang’s protocol cannot resist insider attack, masquerade attack, server spoofing attack, registration center spoofing attack and cannot give the mutual authentication. In Ref. [17] Hsiang and Shih proposed an enhance protocol based on Liao and Wang’s protocol. In 2011, Sood et al. found that Hsiang and Shih’s protocol still had some secure problem. They found that Hsiang and Shih’s protocol cannot resist replay attack, impersonation attack and stolen smart card attack. Furthermore, the password change phase of their protocol is wrong. Then Sood et al. in Ref. [18] proposed an enhance protocol. But Sood et al. actually is susceptible to leak-of-verifier attack, stolen smart card attack and cannot finish the mutual authentication and session key agreement. In the same year, Li Xiong et al. proposed an efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards on the basis of Sood et al.’s protocol and claimed their protocol can tackle these problems. But In our analysis, Li Xiong et al.’s protocol still has some fatal secure problems, like replay attack, Impersonation attack and stolen smart card attack.

Based on all the authentication protocols above, we know that a secure and efficient remote user authentication protocol for multi-server environment should provide mutual authentication, key agreement, secure password update, low computation requirements and resistance to different feasible attacks. So we provide a multi-server architecture authentication protocol using smart card which can be used for SSO system to satisfy all the requests.

III. SSO AUTHENTICATION ARCHITECTURE WITH SMART CARD

The protocol we proposed in this paper in based on SSO authentication architecture with smart card. This SSO authentication architecture is based on the Broker-Based SSO architecture. In the traditional Broker-Based SSO architecture in Fig. 1, there is a control server which is used to register and authenticate users and the service providing servers. The control server communicates with the users directly, so it can easily attack by the malicious users or attackers.

In allusion to solve the secure problem in Broker-Based SSO architecture, our SSO authentication
architecture with smart card puts the control server behind the service providing servers and not communicates with the users directly. In Fig.2, there are three parties in our authentication architecture, the user with smart card, the service providing server, and the control server CS.

A. The Control Server CS

The control server CS is equivalent to the registration center. It manages all the registrations with the service providing servers and the users, and stored their secret identity information for the authentication. The control server CS only communicates with all the service providing servers, and it is not directly accessible to the users and thus it is less likely to be attacked. When a user wants to get some services, the control server needs to verify not only the user, but also the service providing server which provides the services the user needs.

Fig. 2. SSO authentication architecture with smart card

B. The Service Providing Server

The service providing server is the bridge between the user and the control server CS, when the user wants to verify him/her. In the architecture, there are many service providing servers to provide different kinds of services only to all the users. When a service providing server wants to provide services to the user, it must register itself to the control server CS.

C. The User with Smart Card

Every user has his/her own smart card to login the service providing servers with the identity and the password. The smart card can store some secret information of the user in the authentication, and perform some cryptographic operations to verify the authenticity of the user.

IV. SINGLE SIGN-ON AUTHENTICATION PROTOCOL USING SMART CARD

In this section, we propose secure dynamic identify based single sign-on authentication protocol using smart card. In Table I, the notations used in this section are listed. This protocol consists of four phases, the registration phase, the login phase, the authentication phase, the session key agreement phase and the password change phase, which are summarized in Fig. 2, Fig.3 and Fig. 4. In our proposed protocol, we only use one-way hash function to provide a strong mutual authentication, and use various different random numbers to achieve dynamic identify. When our protocol used in SSO system, it only needs one login phase when user login to the system in first time.

TABLE I.

NOTATIONS USED IN THIS PAPER

<table>
<thead>
<tr>
<th>Notation</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U_i)</td>
<td>The i th user</td>
</tr>
<tr>
<td>(S_j)</td>
<td>The j th service providing server</td>
</tr>
<tr>
<td>(CS)</td>
<td>The control server</td>
</tr>
<tr>
<td>(ID_i)</td>
<td>The identity of the user (U_i)</td>
</tr>
<tr>
<td>(P_i)</td>
<td>The password of the user (U_i)</td>
</tr>
<tr>
<td>(SID_j)</td>
<td>The identity of the server (S_j)</td>
</tr>
<tr>
<td>(y_i)</td>
<td>The random number chosen by (CS) for user (U_i)</td>
</tr>
<tr>
<td>(x)</td>
<td>The master secret key maintained by (CS)</td>
</tr>
<tr>
<td>(b)</td>
<td>A random number chosen by the user for registration</td>
</tr>
<tr>
<td>(CID_i)</td>
<td>The dynamic identity generated by the user (U_i) for authentication</td>
</tr>
<tr>
<td>(SK)</td>
<td>A session key shared among the user, the service providing server and the (CS)</td>
</tr>
<tr>
<td>(N_{i1})</td>
<td>A random number generated by the user (U_i)'s smart card</td>
</tr>
<tr>
<td>(N_{i2})</td>
<td>A random number generated by the server (S_i) for the user (U_i)</td>
</tr>
<tr>
<td>(N_{i3})</td>
<td>A random number generated by the (CS) for the user (U_i)</td>
</tr>
<tr>
<td>(h(.))</td>
<td>A one-way hash function</td>
</tr>
<tr>
<td>(\oplus)</td>
<td>Exclusive-OR operation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A. Registration Phase

There are two parts in the registration phase, the one is the user registration, and the other is the server registration.

When the user \(U_i\) wants to access the services legally, the user must register himself/herself to the \(CS\) server with the identity and the password. The details of the user registration phase are as follow:

\[
\begin{align*}
\text{Choose a} & \text{ random secret key } x, \text{ and a random secret key } y_i, \\
\text{A} & = h(b||P_i), \\
\text{Compute } & A_i = h(b||P_i), \\
\text{Send } & (ID_i, A_i) \text{ to the control server } CS \text{ via a secure channel, which guarantee the security of the user identity to avoid network attack like impersonation attack.}
\end{align*}
\]

- Step1: The user \(U_i\) chooses his/her own identity \(ID_i\) and password \(P_i\), and chooses a random number \(b\). Then \(U_i\) computes \(A_i = h(b||P_i)\), and transforms \(ID_i\) and \(A_i\) to the control server \(CS\) via a secure channel, which guarantee the security of the user identity to avoid network attack like impersonation attack.
- Step2: When the control server \(CS\) receiving the message \((ID_i, A_i)\) from the user \(U_i\), \(CS\) chooses a master secret key \(x\), and a random secret key \(y_i\)
which is unique for user $U_i$. Then the control server $CS$ calculates

$$B_i=h(ID_i \parallel x),$$

$$C_i=h(ID_i \parallel h(y_i) \parallel A_i),$$

$$D_i=h(x \parallel y) \oplus h(ID_i \parallel A_i),$$

$$E_i=B_i \oplus A_i, \quad X_i=h(x \parallel y_j).$$

At the same time, the $CS$ stores $(x \parallel y_j, X_j)$ in its client database, and stores the security parameters $(C_i, D_i, E_i, h(y_i), h(.))$ in the smart card of the user $U_i$, then transforms the smart card to the $U_i$ via a secure channel.

- Step3: After the receiving the smart card, the user $U_i$ enters the random number $b$ in his/her smart card. Finally, the smart card contains security parameters as $(C_i, D_i, E_i, h(y_i), h(.), b)$.

In the server registration phase, when a service providing server $S_j$ wants to provide service in this system, it must to register itself to the control server $CS$, the details of the registration phase are as follow:

- Step1: After the receiving the registration request from the service providing server $S_j$, the control server $CS$ chooses a unique secret key $c_j$ for the $S_j$ and computes $h(c_j)$, then sends the security parameters $(h(c_j), h(x \parallel c_j), h(.))$ to the $S_j$ through a secure channel.

- Step2: After the server $S_j$ receiving the message $(h(c_j), h(.)), S_j$ chooses its identity $SID_j$ and a secret random number $SK_j$, then computes $h(SK_j \parallel h(c_j))$, and sends the security parameters $(SID_j, h(SK_j \parallel h(c_j)))$ to the control server $CS$ via the secure channel.

### Authentication Phase

- Step3: When receiving the authentication request message $(SID_j)$ from the $S_j$, the smart card generates a random number $N_{sid}^j$ and computes:

$$B_j=E_j \oplus A_j,$$

$$h(x \parallel y_j)=D_j \oplus h(ID_j \parallel A_j),$$

$$F_j=N_{sid}^j \oplus h(x \parallel y_j),$$

$$CID_j=h(x \parallel y_j) \oplus B_j \oplus h(N_{sid}^j \parallel N_{sid} \parallel SID_j).$$

Then the smart card sends the login request message $(F_j, CID_j)$ to the service providing server $S_j$ through a public channel.

$$CID_j=h(x \parallel y_j) \oplus B_j \oplus h(N_{sid}^j \parallel N_{sid} \parallel SID_j).$$

Then the smart card sends the login request message $(F_j, CID_j)$ to the service providing server $S_j$ through a public channel.

### Login Phase

- Step1: When the user $U_i$ wants to login to the service providing server $S_j$, the user $U_i$ inserts his/her smart card into the card reader, and enters his/her identity $ID_i$ and password $P_i$.

- Step2: When receiving the login request message $(SID_i)$ from the user $U_i$, the service providing server $S_j$, based on the identity of the service providing server $S_j$ and the server identity $SID_j$, the smart card computes $A_i=h(b \parallel P_i), C_i'=h(ID_i \parallel h(y_i) \parallel A_i)$, and checks whether $C_i'=C_i$. If they are equal, that means the user $U_i$ is a legal client, otherwise, the user $U_i$ enters correct identity and password again.

- Step3: After verification, the smart card sends the authentication request message $(SID_j)$ to the service providing server $S_j$ through a public channel. After receiving the request message, the server $S_j$ chooses a random number $N_{sid}$, and sends it to the user $U_i$.
Step 5: After verification, the control server CS computes
\[ N_{id} = F_i \oplus h(x \parallel y_i), \]
\[ X_i' = CID_i \oplus h(N_{id} \parallel N_i \parallel SID_i). \]
Then finding the corresponding \( X_i \) to compare with \( X_i' \). If the value of \( X_i' \) does not match with any value of \( X_i \) in its client database, the CS rejects the login request and terminates this session, otherwise, the CS accepts the login request of the user \( U_i \).

Step 6: After the control server CS accepts the login request of the user \( U_i \), it generates a random number \( N_{id} \), and computes:
\[ B_i = X_i \oplus h(x \parallel y_i), \]
\[ T_i = N_{id} \oplus N_i \oplus h(N_{id} \parallel SID_i), \]
\[ Q_i = h(N_{id} \parallel N_i \parallel N_{id}) \oplus h(B_i \parallel y_i \parallel N_{id}), \]
\[ R_i = N_{id} \oplus N_i \oplus h(B_i \parallel y_i \parallel N_{id}), \]
\[ V_i = h(h(N_{id} \parallel N_i \parallel y_i) \oplus h(B_i \parallel y_i \parallel N_{id})). \]
Then, the control server CS sends the mutual authentication message \( (T_i, Q_i, R_i, V_i) \) to the server \( S_j \).

Step 7: When receiving the message \( (T_i, Q_i, R_i, V_i) \), the service providing server \( S_j \) computes:
\[ N_{i1} \oplus N_{i2} = T_i \oplus h(N_{id} \parallel SID_i), \]
\[ h(B_i \parallel y_i \parallel N_{id}) = Q_i \oplus h(N_{id} \oplus N_i \oplus N_{id}), \]
\[ V_i' = h(h(N_{id} \parallel N_i \parallel y_i) \parallel h(B_i \parallel y_i \parallel N_{id})). \]
The \( S_j \) checks whether \( V_i' = V_i \). If they are not equal, the \( S_j \) terminates the session. Otherwise, the server \( S_j \) sends the mutual authentication message \( (Q_i, R_i, V_i) \).

Step 8: When the user \( U_i \) receives the message \( (Q_i, R_i, V_i) \) from the server \( S_j \), the smart card computes:
\[ N_{i2} \oplus N_{i3} = R_i \oplus h(h(y_i \parallel N_{id})), \]
\[ h(B_i \parallel y_i \parallel N_{id}) = Q_i \oplus h(N_{id} \parallel N_i \parallel N_{id}), \]
\[ V_i' = h(h(N_{id} \parallel N_i \parallel y_i) \parallel h(B_i \parallel y_i \parallel N_{id})), \]
and checks whether \( V_i' = V_i \). If they are not equal, the \( S_j \) terminates the session. Otherwise, the user \( U_i \) is a legal client.

D. Session Key Agreement Phase

After the user \( U_i \), the server \( S_j \) and the server CS finish the mutual authentication, all of them can calculate the session key from the message \( (T_i, Q_i, R_i, V_i) \) sent by the server CS. This session key is used for the encryption of the communication between the user \( U_i \) and server \( S_j \).

In the session key agreement phase, the server CS bases on the computed authentication parameters \( \{B_i, y_i, N_{id}, N_{i2}, N_{i3}\} \) can calculate the session key \( SK \), and the server \( S_j \) uses the message \( (T_i, Q_i, R_i, V_i) \) to calculate the session key \( SK \), and the user uses the message \( (Q_i, R_i, V_i) \) to calculate the session key \( SK \).

Finally, the user \( U_i \), the server \( S_j \) and CS agree on the common session key:
\[ SK = h(h(B_i \parallel y_i \parallel N_{id}) \parallel (N_{id} \parallel N_i \parallel N_{id})). \]

E. Password Change Phase

When the user \( U_i \) changes his/her password \( P_i \) to a new password \( P_i' \), there is no need for the control server to join in. The user \( U_i \) inserts his/her smart card and enters the identity \( ID_i \) and \( P_i \), the smart card computes:
\[ A_i = h(b \parallel P_i), \]
\[ B_i = E_i \oplus A_i, \]
\[ h(x \parallel y_i) = D_i \oplus h(ID_i \parallel A_i), \]
\[ C_i = h(ID_i \parallel h(y_i) \parallel A_i), \]
and checks whether \( C_i = C_i \). If they are equal, the user \( U_i \) asks to submit a new password \( P_i' \). Then the smart card computes:
\[ A_i' = h(b \parallel P_i'), \]
\[ C_i' = h(ID_i \parallel h(y_i) \parallel A_i'), \]
\[ D_i' = h(x \parallel y_i) \oplus h(ID_i \parallel A_i'), \]
\[ E_i' = A_i' \oplus B_i, \]
and stores the new parameters \( C_i', D_i', E_i \) into the smart card to replace \( C_i, D_i, E_i \) to finish the password change phase.

V. PROTOCOL ANALYSIS

In this section, we discuss the security analysis of the proposed protocol within replay attack, impersonation attack, stolen smart card attack, leak-of-verifier attack, and user’s anonymity. And then we talk about the performance and functionality between the proposed protocol and other related multi-server architecture authentication protocols.

A. Security Analysis

1) Replay attack

In this type of attack, an attacker may try to pretend a legal user to login the server \( S_j \) with the message which is
send before by a legal user. In each phase of the proposed protocol, the user $U_i$, the service providing server $S_j$ and the control server $CS$ choose the different random numbers $N_{ij}$, $N_{ij}$, $N_{ij}$ to compute and verify the identity authenticate message. When the user $U_i$ verifying his/her own identity to the service providing server $S_j$, the $S_j$ send a random number $N_{ij}$ to the $U_i$. When the service providing server $S_j$ verifying his/her own identity to the control server $CS$, the $CS$ send a random number $N_{ij}$ to the $S_j$. These random numbers $N_{ij}$, $N_{ij}$, $N_{ij}$, $N_{ij}$ and $N_{ij}$ guarantee that the authenticate messages transmitted in a public channel are different and legal only in every session of the protocol. The control server $CS$ via random numbers $N_{ij}$ and $N_{ij}$ to ensure the login request messages of the user $U_i$ and the server $S_j$ are fresh, it is effective to prevent the replay attack.

2) Impersonation attack

An attacker or a malicious user using the previously eaves-dropped message or the information obtained from the lost smart card, to forge a legal login request message $F_i, C_i$ to pretend a valid user. However, in our proposed protocol, the attacker and malicious user $U_k$ cannot compute the legal identity message $A_i$, $B_i$, and $C_i$, from the previous login request message. If the malicious user $U_k$ has his/her own smart card, he/she can calculate these information $h(y_i)$ and $h(x_i \parallel y_i)$ related to the control server $CS$, but every legal user has different $y_i$, so the malicious user $U_k$ cannot compute the effective identity information to pretend a valid user since he/she cannot get $A_i$, $B_i$, $E_i$. Because the identity information of the valid users are stored in the client database of the control server $CS$, the malicious user $U_k$ cannot guess $A_i$ and $B_i$ to forge a login request message to start an impersonation attack.

3) Stolen smart card attack

We assume that the user $U_i$’s smart card has been lost or stolen, then the attacker can get the information stored in the smart card $(C_i, D_i, E_i, h(\ast), h(y_i), b)$. Since the attacker cannot get the information $x$ and $y_i$, he/she cannot guess the real identity $ID_i$ and password $P_i$ from the breached information, and cannot get or refresh the user $U_i$’s password $P_i$. In addition, if the attacker gets both the $U_i$’s smart card and the previous legal login request message, he/she also cannot compute $A_i$ and $B_i$ through the information above, since the attacker has no way to get $h(x_i \parallel y_i)$. So our protocol can prevent the stolen smart card attack.

4) Leak-of-verifier attack

For the user part, in our protocol, even though there some secret information related to the authentication is stored in the client database of the control server $CS$, but the attacker also cannot compute the user’s identity information $B_i$, and the secret parameter $y_i$ from the leaked information of the control server $CS$’s database, since the CS has the master secret key $x$ which is supposed to be safe. So our protocol can resist the Leak-of-verifier attack for the user.

For the service providing server part, if the database of the control server $CS$ is leaked, the attacker also cannot get any effective information from $S_j$, because of the safety master secret key $x$. So our protocol can resist the Leak-of-verifier attack for the service providing server.

5) User’s anonymity

A secure channel between the user and the control server $CS$ protects the identity information not to be published in the registration phase. In this phase, the user sends a masked identity $CID_i= h(x_i \parallel y_i) \oplus B_i \oplus h(N_{ij} \parallel NS_i \parallel SID_i)$ to the service proving server $S_j$ and the control server as a substitute for the real identity $ID_i$ for its authentication. The method proposed by authentication and session key agreement is based on computing the secret information $B_i$ and $y_i$, but not the real identity $ID_i$. So when the user logging in the system, the dynamic authentication $CID_i$ is different in every phase and the attacker cannot distinguish the differences among the different phases. So we can say our protocol can provide the user’s anonymity.

B. Performance and Functionality Analysis

In this section, we discuss the performance and functionality of our proposed protocol and then make comparisons with some related multi-server authentication protocols. We analyze our protocol and some related protocol in two ways, the one is the computational complexity, and the other is security and functionality properties.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Login phase</th>
<th>Verification phase</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed protocol</td>
<td>4 $T_h$</td>
<td>21 $T_h$</td>
<td>25 $T_h$</td>
</tr>
<tr>
<td>Li X et al (2011)</td>
<td>7 $T_h$</td>
<td>21 $T_h$</td>
<td>28 $T_h$</td>
</tr>
<tr>
<td>Sood et al. (2011)</td>
<td>7 $T_h$</td>
<td>18 $T_h$</td>
<td>25 $T_h$</td>
</tr>
<tr>
<td>Hsiang and Shih (2009)</td>
<td>7 $T_h$</td>
<td>17 $T_h$</td>
<td>24 $T_h$</td>
</tr>
<tr>
<td>Liao and Wang (2009)</td>
<td>6 $T_h$</td>
<td>9 $T_h$</td>
<td>15 $T_h$</td>
</tr>
</tbody>
</table>

In the analysis of the computational complexity of the protocols, we define the notation $T_h$ as the time complexity for the hashing function $h(\cdot)$. Because XOR operation requires very few computations, it is usually negligible considering its computation cost. In Table II, it shows the computational complexity of our protocol and other related protocols, we first consider for two parts: login phase, and authentication and session key agreement phase, because in these two phases, there is a strict request for the running time. From Table II, we can see that our protocol does not have an obvious predominance in the computational complexity, but it is worth to achieve these security and functionality properties with several additional hash operations.
In the analysis of the security and functionality properties of the protocols, we consider about User's anonymity, Computation cost, Single registration, No time synchronization, Resist replay attack, Resist impersonation attack, Resist leak-of-verifier attack, Resist stolen smart card attack. In Table III, we can find that our proposed protocol has the advantage over the other related protocols.

VI. CONCLUSION

In this paper, we first described the application condition of SSO system in the network, and the importance of the authentication protocol in the SSO system. Then we introduced some authentication protocols which can be used in the SSO system. And we abstracted a SSO authentication architecture using smart card to solve the authentication problem in SSO system. In this architecture there are three actors: the control server CS, the service providing server SJ and the user UI with smart card. Then secure dynamic identity based single sign-on authentication protocol using smart card based on the architecture talked above. Finally, we make a secure analysis of our proposed protocol, and some performance and functionality comparisons with our proposed protocol and some related multi-server authentication protocols. In conclusion, our proposed protocol keeps the efficiency and is more secure. So our proposed protocol is suitable for the SSO system applications.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No.61172068, 61003300; the Fundamental Research Funds for the Central Universities (Grant No.K50511010003), the Key Program of NSFC-Guangdong Union Foundation under Grant No. U0835004 and Program for New Century Excellent Talents in University(Grant No. NCET-11-0691).

REFERENCES


Qingqi Pei received his BEng, MEng and Ph.D. degrees in Computer Science and Cryptography from Xidian Univ, in 1998, 2005 and 2008, respectively. He is now an associate professor and member of the State Key Laboratory of Integrated Services Networks, also a Professional Member of ACM and Member of IEEE, Senior Member of Chinese Institute of Electronics and China Computer Federation. His research interests focus on digital contents protection and wireless networks and security.

Jie Yu is a Master in cryptography. Her research interests focus on networks and security.