A Secure Dynamic Identity based Single Sign-On Authentication Protocol

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 service systems, it is extremely hard for user to remember too many different IDs and passwords when he/she to login these remote application systems, the management of the servers is very complicated, and it is consumed huge network resource. So the single sign-on (SSO) system has been proposed to solve this problem.

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

doi:10.4304/jsw.9.1.154-161

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).

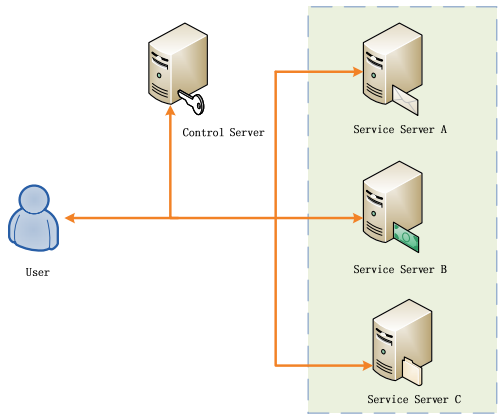
The most famous authentication in SSO system – the Kerberos [12] protocol also has some limits, like all the servers are exposed for users and two servers are both used to authenticate users. In 2006, Yang et al. [13] proposed an authentication and key exchange protocol which is similar to the Kerberos 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.



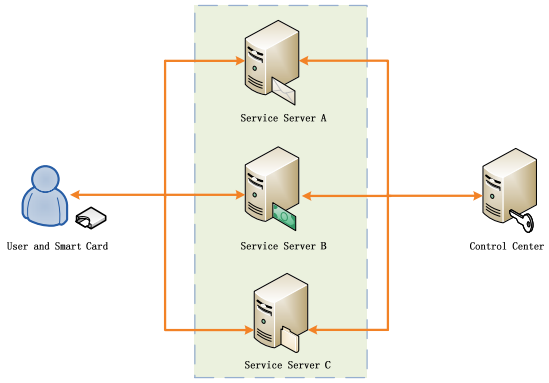
Figrue 1. Broker-Based SSO architecture

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.



Figrue 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 verity 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.

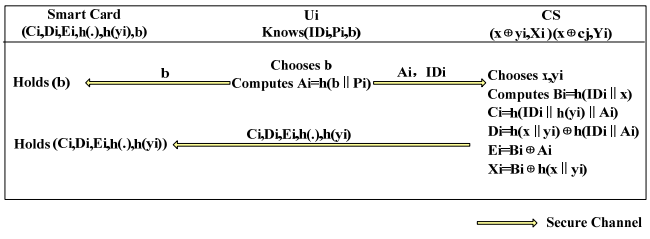
NOTATIONS USED IN THIS PAPER

Notation	*Descriptions*
U_i	The *i* th user
S_j	The *j* th service providing server
CS	The control server
ID_i	The identity of the user U_i
P_i	The password of the user U_i
SID_k	The identity of the server S_k
y_i	The random number chosen by *CS* for user U_i
x	The master secret key maintained by CS
b	A random number chosen by the user for registration
CIDi	The dynamic identity generated by the user U_i for authentication
SK	A session key shared among the user, the service providing server and the *CS*
N_{i1}	A random number generated by the user U_i's smart card
N_{i2}	A random number generated by the server S_k for the user U_i
N_{i3}	A random number generated by the *CS* for the user U_i
h(.)	A one-way hash function
⊕	Exclusive-OR operation
||	Message concatenation operation

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:



Figrue 2. User registration phase

- 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 massage (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

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

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

$D_i=h(x \| y_i) \oplus h(ID_i \| A_i)$,

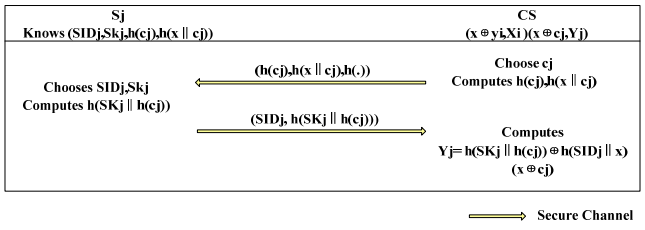
$E_i=B_i \oplus A_i$, $X_i=B_i \oplus h(x \| y_i)$.

At the same time, the *CS* stores $(x \oplus y_i, X_i)$ 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 para-meters$(h(c_j), h(x \| c_j), h(.))$ to the S_j through a secure channel.
- Step2: After the server S_j receiving the massage $(h(c_j),h(.))$, S_j chooses its identity SID_j and a secret random number SK_j, then computes $h(SK_j \| h(c_j))$, and sends the security parameters$(SID_j, h(SK_j \| h(c_j)))$ to the control server *CS* via the secure channel.

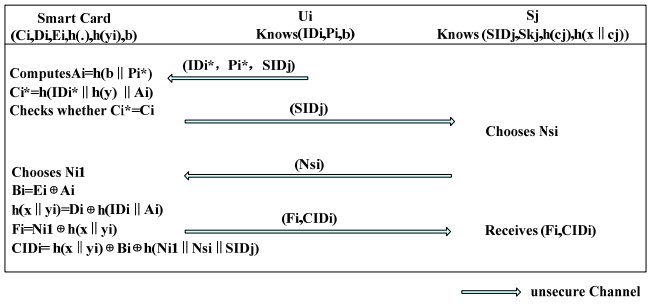


Figrue 3. Server registration phase

- Step3: When receiving the massage $(SID_j, h(SK_j \| h(c_j)))$, the control server *CS* computes $Y_j= h(SK_j \| h(c_j)) \oplus h(SID_j \| x)$. The *CS* stored the parameters $(x \oplus c_j, Yj)$ in its server database, and computes $h(x \| c_j)$. The security parameters $h(x \| c_j)$ is send to the service providing server S_j through the secure channel.

B. 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 , password P_i and the server identity SID_j. The smart card computes
$A_i=h(b \| P_i)$,
$C_i'=h(ID_i \| h(y_i) \| A_i)$,
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.
- Step2: 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 Ns_i, and sends it to the user U_i.



Figrue 4. Login phase

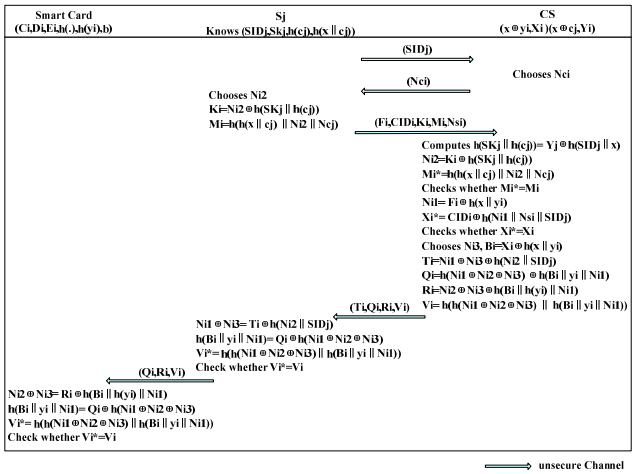
- Step3: Upon receiving the random number Ns_i from the S_j, The smart card generates a random number N_{i1} and computes:
$B_i=E_i \oplus A_i$,
$h(x \| y_i)=D_i \oplus h(ID_i \| A_i)$,
$F_i=N_{i1} \oplus h(x \| y_i)$,
$CID_i= h(x \| y_i) \oplus B_i \oplus h(N_{i1} \| Ns_i \| SID_j)$.
Then the smart card sends the login request message (F_i, CID_i) to the service providing server S_j through a public channel.

C. Authentication Phase

- Step1: When receiving the login request message from the user U_i, the service providing server S_j sends an authentication request massage (SID_j) to the control server *CS*.
- Step2: After the control server *CS* receiving the authentication request massage (SID_j), it returns a random number Nc_j.
- Step3: Upon receiving the return message Nc_j, the service providing server S_j chooses a random number N_{i2} and computes
$K_i=N_{i2} \oplus h(SK_j \| h(c_j))$,
$M_i=h(h(x \| c_j) \| N_{i2} \| Nc_j)$.
Then the login request massage $(F_i, CID_i, K_i, M_i, Ns_i)$ is send to the control server *CS* via a public channel.
- Step4: After receiving the login request massage$(F_i, CID_i, K_i, M_i, Ns_i)$, based on the identity of the service providing server S_j $-SID_j$, the control server *CS* finds the corresponding authentication massage $(x \oplus c_j, Y_i)$ in its server

database. The *CS* computes *cj* by $x \oplus c_j$, and
$h(SK_j \| h(c_j))=Y_j \oplus h(SID_j \| x)$,
$N_{i2} =K_i \oplus h(SK_j \| h(c_j))$,
$Mi'=h(h(x \| c_j) \| N_{i2} \| Nc_j)$,
and checks whether M_i' equal to the received M_i. If they are equal, that means the server S_j is a legal client, otherwise, the *CS* terminates the session.

- Step5: After verification, the control server *CS* computes
  $N_{i1}= F_i \oplus h(x \| y_i)$,
  $X_i'= CID_i \oplus h(N_{i1} \| Ns_i \| SID_j)$.
  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.
- Step6: After the control server *CS* accepts the login request of the user U_i, it generates a random number N_{i3},and computes:
  $B_i=X_i \oplus h(x \| y_i)$,
  $T_i=N_{i1} \oplus N_{i3} \oplus h(N_{i2} \| SID_j)$,
  $Q_i=h(N_{i1} \oplus N_{i2} \oplus N_{i3}) \oplus h(B_i \| y_i \| N_{i1})$,
  $R_i=N_{i2} \oplus N_{i3} \oplus h(B_i \| h(y_i) \| N_i1)$,
  $V_i= h(h(N_{i1} \oplus N_{i2} \oplus N_{i3}) \| h(B_i \| y_i \| N_{i1}))$.
  Then, the control server *CS* sends the mutual authentication message (T_i, Q_i, R_i, V_i) to the server S_j.



Figrue 5. Authentication phase

- Step7: When receiving the message (T_i, Q_i, R_i, V_i), the service providing server S_j computes:
  $N_{i1} \oplus N_{i3}= T_i \oplus h(N_{i2} \| SID_j)$,
  $h(B_i \| y_i \| N_{i1})= Q_i \oplus h(N_{i1} \oplus N_{i2} \oplus N_{i3})$,
  $V_i' = h(h(N_{i1} \oplus N_{i2} \oplus N_{i3}) \| h(B_i \| y_i \| N_{i1}))$.
  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).
- Step8: When the user U_i receives the message (Q_i, R_i, V_i) from the server S_j, the smart card computes:
  $N_{i2} \oplus N_{i3}= R_i \oplus h(B_i \| h(y_i) \| N_{i1})$,
  $h(B_i \| y_i \| N_{i1})= Q_i \oplus h(N_{i1} \oplus N_{i2} \oplus N_{i3})$,
  $V_i' = h(h(N_{i1} \oplus N_{i2} \oplus N_{i3}) \| h(B_i \| y_i \| N_{i1}))$,
  and checks whether The S_j 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 massage (T_i, Q_i, R_i, V_i) send 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 $\{B_i, y_i, N_{i1}, N_{i2}, N_{i3}\}$ can calculate the session key *SK*, and the server S_j uses the massage (T_i, Q_i, R_i, V_i) to calculate the session key *SK*, and the user uses the massage (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:

$$SK = h\left(h\left(B_i \| y_i \| N_{i1}\right) \| \left(N_{i1} \oplus N_{i2} \oplus N_{i3}\right)\right).$$

E. Password Change Phase

When the user U_i changes his/her password P_i to a new password P_i^{new}, 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:
$A_i=h(b \| P_i)$,
$B_i=E_i \oplus A_i$,
$h(x \| y_i)=D_i \oplus h(ID_i \| A_i)$,
$C_i'=h(ID_i \| h(y_i) \| A_i)$,
and checks whether $C_i'=C_i$. If they are equal, the user U_i asks to submit a new password P_i^{new}. Then the smart card computes
$A_i^{new}=h(b \| P_i^{new})$,
$C_i^{new}=h(ID_i \| h(y) \| A_i^{new})$,
$D_i^{new}= h(x \| y_i) \oplus h(ID_i \| A_i^{new})$,
$E_i^{new} =Ai^{new} \oplus B_i$,
and stores the new parameters C_i^{new}, D_i^{new}, E_i^{new} 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_{i1}, N_{i2}, N_{i3} 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 Ns_i 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 Nc_j to the S_j. These random numbers N_{i1}, N_{i2}, N_{i3}, Ns_i and Nc_j 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 Ns_i and Nc_j 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, CID_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 CID_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 $h(x \| y_i)$ 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 $(C_i, D_i, E_i, h(\bullet), h(y), b)$. 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 $h(x \| y_i)$. 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 $CID_i = h(x \| y_i) \oplus B_i \oplus h(N_{i1} \| Ns_i \| SID_j)$ 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 II.

COMPUTATIONAL COMPLEXITY COMPARISONS

Protocols	Login phase	Verification phase	Total
Proposed protocol	4 T_h	21 T_h	25 T_h
Li X et al (2011)	7T_h	21 T_h	28 T_h
Sood et al. (2011)	7 T_h	18 T_h	25 T_h
Hsiang and Shih (2009)	7 T_h	17 T_h	24 T_h
Liao and Wang (2009)	6 T_h	9 T_h	15 T_h

In the analysis of the computational complexity of the protocols, we define the notation T_h as the time complexity for the hashing function $h(\sim)$. 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.

TABLE III.

FUNCTIONALITY COMPARISONS

Functionalities	Proposed protocol	Li X et al (2011)	Sood et al. (2011)	Hsiang and Shih (2009)	Liao and Wang (2009)
User's anonymity	Yes	Yes	Yes	Yes	Yes
Computation cost	Low	Low	Low	Low	Low
Single registration	Yes	Yes	Yes	Yes	Yes
No time synchronization	Yes	Yes	Yes	Yes	Yes
Resist replay attack	Yes	No	Yes	No	No
Resist imperson-ation attack	Yes	No	No	No	No
Resist leak-of-verifier attack	Yes	Yes	No	Yes	Yes
Resist stolen smart card attack	Yes	No	No	No	No

In the analysis of the security and functionality properties of the protocols, we consider about User's anonymity, Computation cost, Single registration, No time synchroni-zation, Resist replay attack, Resist impersonation attack, Resist leak-of-verifier attack, Resist stolen smart card attack, Correct password update, Correct mutual authentication, Correct session key agreement. 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 S_j and the user U_i with smart card. Then secure dynamic identify 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

[1] Lamport L. Password authentication with insecure communication. Communications of the ACM 1981; 24(11):770–2.

[2] Ford W, Kaliski BS. Server-assisted generation of a strong secret froma password. In: Proceedings of IEEE 9th international workshop enabling technologies, June 2000, p. 176–80.

[3] Jablon DP. Password authentication using multiple servers. In: Proceedings of the RSA security conference, April 2001, p. 344–60.

[4] Lee WB, Chang CC. user identification and key distribution maintaining anonymity for distributed computer network. Computer System Science 2000; 15(4): 211–4.

[5] Li L-H, Lin L-C, Hwang M-S. A remote password authentication scheme for multi-server architecture using neural networks. IEEE Transactions on Neura Networks, vol. 12(6), pp.1498–504, 2001.

[6] Lin IC, Hwang MS, Li LH. A new remote user authentication scheme for multi-server architecture. Future Generation Computer System 2003; 19(1):13–22.

[7] Raimondo MD, Gennaro R. Provably secure threshold password-authenticated key exchange. In: Proceedings of the advances in cryptology (Eurocrypt' 03), p. 507–23, May 2003.

[8] Brainard J, Juels A, Kaliski B, SzydloM. A new two-server approach for authentication with short secrets. In: Proceedings of the USENIX security symposium, August 2003, p. 201–14.

[9] Juang W-S. Efficient multi-server password authenticated key agreement using smart cards. IEEE Transaction on Consumer Electronics, vol. 50(1), pp. 251–5, 2004.

[10] Chang C-C, Lee J-S. An efficient and secure multi-server password authentication scheme using smart cards. In: Proceedings of the third international conference on cyber worlds, pp417–22. November 2004.

[11] Hu L, Niu X, Yang Y. An efficient multi-server password authenticated key agreement scheme using smart cards. In: Proceedings of the international conference on multi media and ubiquitous engineering (MUE'07), April 2007, p. 903–07.

[12] J Steiner, C Nevman, JI Schillier. Kerberos: an Authentication Service for Open Network Systerms[c]. In: Proc of Winter Usenix Conference, Dallas, 2005.

[13] Yang Y, Deng RH, Bao F. A practical password-based two-server authentication and key exchange system. IEEE Transactions on Dependable and Secure Computing 2006; 3(2): 105–14.

[14] Mackenzie P, Shrimpton T, Jakobsson M. Threshold password-authenticated key exchange. Journal of Cryptology 2006;19(1):27–66.

[15] Tsai J-L. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers & Security, vol. 27(3-4), pp.115-21, 2008.

[16] Liao Y-P, Wang S-S. A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces, vol. 31(1), pp.24-9, 2009.

[17] Hsiang H-C, Shih W-K. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces, vol. 31(6), pp.1118-23, 2009.

[18] Sood S-K, Sarje A-K, Singh K. A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network and Computer Applications, vol. 34(2), pp.609-18, 2011.

[19] Li X, et al. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications 2011.

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.