Turk J Elec Eng & Comp Sci, Vol.20, No.6, 2012, © TÜBİTAK
doi:10.3906/elk-1010-820

Improvement of smart card based password authentication scheme for multiserver environments

Zuowen TAN
Department of Computer Science and Technology, School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi Province-CHINA
e-mail: tanzyw@gmail.com

Received: 03.10.2010

Abstract

In multiserver (MS) environments, it is preferable for a remote user to login to different service provider servers by keying in the same password. Recently, Wang et al. proposed an improvement on the dynamic identity-based smart card authentication scheme of Liao and Wang for MS environments. Sandeep et al. improved the dynamic identity-based smart card authentication scheme of Hsiang et al. for MS architecture. However, we found that the schemes of Wang et al. and Sandeep et al. failed to provide service provider server authentication, perfect forward security, and login scalability. In addition, the scheme of Sandeep et al. was insecure against stolen verifier attacks. This paper proposes an improved smart card-based password authentication scheme for MS environments. The new scheme removes all of the abovementioned weaknesses. The proposed identity-based smart card authentication scheme satisfies the following properties: C1. User authentication; C2. Service provider server authentication; C3. Control server authentication; C4. Perfect forward security; C5. Freedom of password change; C6. Scalability of login; C7. Resistance to stolen verifier attacks; and C8. High efficiency.

Key Words: *Multiserver, password, smart card, CDH assumptions*

1. Introduction

The internet environment requires mechanisms that prevent unauthorized users from accessing resources. A user and a server must agree on a session key beforehand. The Diffie-Hellman key exchange provides a shared secret key, but no authentication [1]. The common building blocks such as nonces, certificates, the Diffie-Hellman key exchange, and encrypted or signed messages are applied to build up authentication key exchange schemes. Some derivation systems of the Diffie-Hellman key exchange are given in [1]. Lamport proposed a solution, a password-based authentication scheme, to deal with remote user access [2]. A password is one of the most acceptable and widely used authentication mechanisms [3]. Many authentication schemes are password-based (e.g., telnet and Kerberos). However, Lamport's scheme needs a password table stored in the computer system. Moreover, the low entropy of the password makes it susceptible to dictionary attacks. Recently, smart cards have been widely applied to the construction of password-based authentication schemes. Smart cards can store the sensitive data of the card holder. The smart card holder inserts his or her card into a reader and keys in the password and identity information. The smart card takes these as input, computes the login message, and sends the login message to the server. Many smart card-based remote user authentication schemes have been proposed due to their low cost and portability. Smart card-based password authentication (SCPA) is one of the most convenient and commonly used authentication techniques. However, since the smart card has a limited capacity for computation and storage, some complicated solutions are not suitable for smart card-based authentication schemes. In the literature, password authentication schemes using smart cards can be divided into 2 types, namely hash-based authentication and public-key based authentication. The first type of SCPA scheme is always more efficient than the second one.

Most of the SCPA schemes address the single-server (SS) environment. A variety of efficient SCPA-SS schemes have already been proposed [4-6].

Since each user needs to have different sets of identities and passwords with different remote servers in the multiserver (MS) environment, it is infeasible to apply the authentication methods in a SS environment to the MS environment. Different SCPA schemes have been proposed to access the resources in the MS environment [5-11]. In 2000, Ford and Kaliski [12] proposed the first SCPA-MS scheme, which splits a password among 2 or more independent servers. However, the scheme has a high computation cost and needs a prior secure authentication channel between the user and the server. In 2001, Tsaur et al. proposed a SCPA-MS scheme based on the RSA cryptosystem [13]. Li et al. applied an artificial neural network to construct an inefficient SCPA-MS scheme [5]. Tsaur et al. improved the scheme of Li et al. [14]. However, the scheme of Tsaur et al. is still insecure [15]. In 2004, Juang proposed a SCPA-MS scheme based on the hashing function and symmetric-key cryptosystem [16] Nevertheless, Juang's scheme suffers from online guessing attacks [17] and offline dictionary attacks [18]. Chang and Lee proposed an improved scheme using a symmetric encryption algorithm [17]. Nevertheless, their scheme [17] still cannot withstand the insider attack, server spoofing attack, and registration center spoofing attack [7]. Hu et al. proposed a SCPA-MS scheme, but their scheme fails to meet the security requirements of the MS environment [19]. The SCPA-MS scheme in [18] is also insecure [8]. Tsai proposed a SCPA-MS scheme based on the one-way hash function without a verification table [20]. Unfortunately, some attacks on it have been made [8-10,21].

Recently, Liao and Wang proposed a SCPA-MS scheme (LW-scheme) [8]. The LW-scheme only uses a hash function to implement a strong authentication. However, the LW-scheme is still vulnerable to insider attacks, masquerade attacks, server spoofing attacks, and registration center spoofing attacks [21,22]. Furthermore, it fails to provide mutual authentication [9,20,21]. In 2009, Hsiang and Shih proposed an improved SCPA-MS scheme (HS-scheme) [22]. However, the HS-scheme has shown [923] that it suffers from replay attacks, impersonation attacks, and stolen smart card attacks. Chen et al. [9] proposed an improvement (CHC-scheme) on the HS-scheme. Sandeep et al. [23] also presented a secure dynamic identity-based SCPA-MS scheme (SAK-scheme) and claimed that the proposed scheme resolves the aforementioned security flaws while keeping the merits of Hsiang and Shih's protocol. Recently, Wang et al. also proposed an efficient SCPA-MS scheme (WJL-scheme) [21]. In this study, we found that the CHC-scheme [9] was vulnerable to offline password guessing attacks, impersonation attacks, and server spoofing attacks. In addition, we will show that the WJL-scheme and the SAK-scheme fail to provide service provider server authentication, perfect forward security, and login scalability. Furthermore, we also demonstrate that the SAK-scheme suffers from stolen verifier attacks.

A new architecture for SCPA schemes in the MS environment was presented [24]. There, 2 levels of the servers are composed of a control server and some service provider servers, in which only the service provider servers communicate directly with the users and a control server does not interact with the users directly. The framework of the MS requires additional efforts from an adversary if the adversary launches an attack. However, the control server in [24] requires a public key for its operations. Thus far, although many SCPA-MS schemes consider the security goals, there is no common set of desirable security properties for SCPA-MS schemes. Recently proposed SCPA-MS schemes still have various security weaknesses that are being overlooked, and many of these schemes have been broken shortly after they were first proposed. A secure SCPA-MS scheme should have some stronger security properties than secure SCPA-SS schemes. In the next section, we will define a refined set of security requirements that not only captures the exact 2-factor authentication but also considers the security of MS parties.

In this paper, we apply the Diffie-Hellman key exchange technique [1] and the hash functions to construct a new SCPA-MS scheme. Upon the computational Diffie-Hellman assumptions and the assumptions of the existence of the collision resistant hash functions, the new SCPA-MS scheme satisfies all of the security requirements defined in this paper. Our scheme removes all of the weaknesses of several previous SCPA-MS schemes [8,9,20-23].

1.1. Our results

In this paper, our contributions are as follows:

1) We highlight the security attributes of a SCPA-MS scheme. These security requirements are an extension and refinement of some previously proposed security requirements of SCPA schemes. The setup of security requirements can facilitate cryptanalyses of SCPA-MS schemes. These requirements are also associated with an adversarial model.

2) We demonstrate that 2 recently proposed schemes fail to satisfy the security properties.

3) We propose a new efficient SCPA-MS scheme. We show that the proposed SCPA-MS scheme satisfies all of the security requirements.

1.2. Organization

The remainder of this paper is organized as follows: In Section 2, we propose a new set of desirable security requirements and an adversarial model of SCPA-MS schemes. In Section 3, we review the WJL-scheme [21] and the SAK-scheme [23]. The cryptanalyses show that the WJL-scheme fails to provide service provider server authentication, perfect forward security, and login scalability, and that the SAK-scheme is insecure. In Section 4, we propose a new SCPA-MS scheme. In Section 5, we analyze its security properties and performance. Finally, the conclusion will be given in Section 6.

2. Security requirements

Consider the SCPA-MS schemes with the architecture of 2-level servers: the control server and the service provider servers, in which the users access the service of service provider servers and a control server does not interact with the users directly. In the literature, the control server is often called the registration center. In our model, the control server does not only act as the registration center but also plays a greater role during the authentication and session key exchange between users and service provider servers. In the SAK-scheme

[23], the control server and the service provider servers come from the same domain. In fact, the control server and the service provider servers are from different domains. Our SCPA-MS model only requires that the service provider servers must register the control server. One control server can manage many service provider servers from different departments. Therefore, the SCPA-MS model has more flexibility and scalability.

2.1. Skeleton of a SCPA-MS scheme

The SCPA-MS scheme involves some service provider servers S_j with identity SID_j, user U_i with identity ID_i, and control server CS. The SCPA-MS scheme is composed of the registration phase, login phase, authentication and session key agreement phase, and password change phase.

1) Registration phase: The CS securely issues a smart card to the U_i, with the smart card being personalized with respect to ID_i and an easily remembered password. The CS also executes the registration with each S_j. In this phase, all of the participants are assumed to be honest and execute the protocol according to the specification of the scheme. Registration is carried out only once for each S_j and each U_i.

2) Login phase: The U_i sends a request message to a S_j. The S_j generates a response message. The S_j then transmits the U_i's request message and its response message to the CS. The U_i can access many S_js by using their smart cards and the same passwords.

3) Authentication and session key agreement phase: The CS authenticates the S_j and the U_i. Next, the S_j and the U_i complete the mutual authentication. Finally, the S_j and U_i agree on the same session key.

4) Password change phase: The U_i can change his password freely without the presentence of the CS.

In the following, we describe the desirable security properties that a secure SCPA-MS scheme should achieve and also describe the attacks that an adversary may mount against SCPA-MS schemes.

2.2. Security goals

C1. User authentication: Both the S_j and the CS are sure that the service requestor is indeed a registered U_i as the U_i claims.

C2. Service provider server authentication: Both the U_i and the CS are sure that the S_j is indeed the S_j that the U_i attempts to access.

C3. Control server authentication: Both the U_i and the S_j are sure that the confirmation message is indeed from the CS.

C4. Perfect forward security: If secret master keys of the CS are compromised, previously established session keys should not be revealed.

C5. Freedom of password change: The password can freely be updated by the smart card holder (a registered U_i) at will, without any interaction with the S_j or the CS. The S_j or the CS can be totally unaware of the password change.

C6. Scalability of login: If a U_i has finished the first-time login to a certain S_j, any interaction with the CS is not necessarily required when the U_i logs in to the same S_j once again. If a SCPA-MS scheme does not hold the scalability of the login, when many logins happen at one time, a bottleneck will be caused. Since the CS is required to engage in every login, if the U_i logs in to the same S_j the first time and the CS does not participate in it, this will easily lead to abuse of the login by a malicious U_i. Therefore, the S_j should be able to control the login of the U_i.

C7. Resistance to the stolen verifier attacks: Neither the S_j nor the CS could get any information about a registered U_i's password or anything derived from the password. In other words, each password is exclusive, which means that the password or the derivatives of the password are only known and kept by the corresponding U_i. In addition, the CS cannot keep the verification tables for the S_j or the registered U_i.

C8. High efficiency: The password should be a short, easily remembered value in the password space. Due to the power constraints and small flash memory of smart cards, the smart card should have a low computation and communication cost. Moreover, a U_i registers with the CS once and can access all of the eligible S_js who have registered with the same CS.

Remarks: Property C4 implies the following property always mentioned in the literature: no verification or password table is stored in a server. However, the latter does not imply property C4. Property C6 implies the single registration mentioned in [16].

2.3. Adversarial model

Consider adversary A, who gets full control over both the communication channel between the U_i and the S_j and the communication channel between the S_j and the CS (except the registration phase). Thus, the A could obtain all of the messages transmitted between the U_i and the S_j and all of the messages transmitted between the S_j and the CS (except the registration message). Of all of the phases in a SCPA-MS scheme, only the registration phase requires a secure channel between the S_j and the CS and a secure channel between the U_i and the CS. In the other phases, there could be various kinds of passive and active adversaries in the communication channel between the U_i and the S_j and the communication channel between the CS and the S_j. The A can eavesdrop on and even block a transmitting message, modify messages, remove messages, or insert messages into the communication channel. Its objective is to compromise mutual authentication between the CS and the S_j, between the U_i and the S_j, and between the CS and the U_i. The A even impersonates the U_i and attempts to access the S_j, or the A impersonates the S_j and provides the U_i with false service. In the MS environment, the insider attacker is more powerful. In such an attack, the A is a malicious-but-registered U_i or S_j. To simulate the insider attack, if a U_i is under attack, the A is allowed to get the passwords and information stored on the smart cards of all of the U_is, except those of a client under attack, and is also allowed to have the registered messages of all of the S_js. If a S_j is the attack target, we allow the A to obtain the passwords and information stored on the smart cards of all of the U_is and obtain the registered messages of the other S_js. If the control server is the A's target, we allow the A to obtain the passwords and information stored in the smart cards of all of the U_is and obtain the registered messages of all of the S_js.

For a SCPA scheme, one basic security property is that the U_i is required to both have the smart card and know the password, which is often called 2-factor authentication. Since Messerges et al. [25] and Kocher et al. [26] pointed out that all existing smart cards cannot prevent the information stored in them from being extracted, for example, by monitoring their power consumption [27], the security of the SCPA scheme is always discussed in the event that the smart card is stolen. In other words, when a U_i is under attack, we also allow an A to either compromise the password or the smart card of the client under attack, but not both.

Only if the attack goal is to obtain previously established session keys can we allow an A to compromise the secret master keys of the CS.

Remarks: The security requirements and adversarial model mentioned above can eliminate the redundancies and ambiguities of previously proposed requirements for the SCPA-MS scheme. It will also simplify cryptanalyses of SCPA-MS schemes.

3. Cryptanalyses of 2 SCPA-MS schemes

In this section, we review the SAK-scheme [23] and the WJL-scheme [21]. The cryptanalyses show that the WJL-scheme and the SAK-scheme fail to provide S_j authentication, perfect forward security, and login scalability.

The notations that we will use for describing the 2 SCPA-MS schemes are given in Table 1. These notations will also be used throughout the paper.

Table 1. Notations.

U_i	ith user	x,y	Secret master key of the CS
S_j	jth service provider server	G	Generator of an elliptic curve group
CS	Control server	$\|\|$	Concatenation
ID_i	Unique identification of the U_i	$\oplus$	XOR operation
SID_j	Identification of the S_j	$\rightarrow$	Transmission of the message
PW_i	Password of the U_i	$H(\ )$	One-way hash function

3.1. Review of WJL-scheme

In the WJL-scheme [21], the CS selects 2 large prime numbers, p and q, and keeps them secret. The CS computes $n = p \times q$ and makes n public. The WJL-scheme consists of 4 phases, i.e. registration, login, authentication and session key agreement, and password change.

1) Registration phase:

Step 1. $U_i \rightarrow CS$: ID_i

Step 2. $CS \rightarrow U_i$: Smart card

The CS computes $SK_i = H(ID_i||x)$, stores (SK_i, ID_i, $H(\ )$, n) on a smart card, and issues the smart card to the U_i through a secure channel.

Step 3. $U_i \rightarrow$ Smart card: SK_i'

The U_i chooses a PW_i, computes $SK_i' = SK_i \oplus H(PW_i)$, and replaces SK_i on the smart card with SK_i'.

Similarly, the S_j registers with the CS. First, the S_j sends its identity, SID_j, to the CS. The CS sends $SK_j = H(SID_j||x)$ to the S_j over a secure communication channel.

2) Login phase:

The U_i keys in ID_i, PW_i, and SID_j. The smart card generates a nonce, N_i, and computes $Req_i = (ID_i||SK_i||SID_j||N_i)^2 \bmod n$. The smart card then sends Req_i to the S_j.

3) Authentication and session key agreement phase:

Step 1. $S_j \rightarrow CS$: Req_i, SID_j, Req_j

The S_j chooses random nonce N_j, computes $Req_j = SK_j \oplus N_j$, and sends $\{Req_i, SID_j, Req_j\}$ to the CS.

Step 2. CS checks $SK_i ?= H(ID_i||x)$

The CS decrypts Req_i and obtains $ID_i||SK_i||SID_j||N_i$. The CS then checks $SK_i ?= H(ID_i||x)$. If the equality does not hold, the CS rejects the login request. Otherwise, the CS chooses a random number, N, and calculates:

$$SK_j = H(SID_j||x), N'_j = SK_j \oplus Req_j, Res_1 = H(SK_j||N'_j) \oplus N, \tag{1}$$

$$K_{SC} = H(SK_j||N'_j||N), R_{ij} = H(ID_{ij}||SID_j||N_i), Res_2 = K_{SC} \oplus R_{ij}, \tag{2}$$

$$Res_3 = H(SK_j||N_i||h(N'_j)), Res_4 = H(K_{SC}||R_{ij}||Res_3). \tag{3}$$

Next, the CS sends the message $(Res_1, Res_2, Res_3, Res_4)$ to the S_j.

Step 3. S_j checks $Res_4? = H(K_{SC}||R'_{ij}||Res_3)$

The S_j computes $N = H(SK_j||N_j) \oplus Res_1$, $K_{SC} = H(SK_j||N_j||N), R'_{ij} = K_{SC} \oplus Res_2$ and checks whether $Res_4? = H(K_{SC}||R'_{ij}||Res_3)$. If the equality does not hold, the S_j rejects the login request. Otherwise, the S_j computes $Res_{j2} = R'_{ij} \oplus H(N_j)$. Next, the S_j sends $\{Res_{j2}, Res_3\}$ to the smart card.

Step 4. Smart card checks $Res_3? = H(SK_j||N_i||H(N'_j))$

The smart card computes $H(N'_j) = Res_{j2} \oplus H(ID_j||SID_j||N_i)$ and checks whether $Res_3? = H(SK_j||N_i||H(N'_j))$. If the equality holds, the smart card computes $Res_i = H(H(N'_j) + 1)$, $SK = H(SID_j||H(N'_j)||H(ID_j||SID_j||N_i))$ and sends Res_i to the S_j.

Step 5. S_j checks $Res_i? = H(H(N_j) + 1)$

Finally, the U_i, the S_j, and the CS agree on session key $SK = H(SID_j||H(N_j)||R'_{ij})$.

3.2. Cryptanalyses of the WJL-scheme

In the following section, we demonstrate that the WJL-scheme does not satisfy security properties C2 (service provider server authentication), C4 (perfect forward security), or C6 (scalability of login).

WJL-scheme fails to provide service provider server authentication (C2)

In the WJL-scheme, the CS identifies the S_j only by the SID_j. The CS has not authenticated the S_j. Thus, any A can impersonate the S_j and generate random message Req_j. The A sends $\{Req_i, SID_j, Req_j\}$ to the CS. The CS will then believe that the communicating party, A, is a S_j with a SID_j.

WJL-scheme fails to provide perfect forward security (C4)

Suppose that the CS's secret master key x is compromised. An A mounts an attack as follows. First, the A computes $SK_j = h(SID_j||x)$. According to the adversarial model, the A can intercept all of the messages transmitted over the public communication channels. Without loss of generality, assume that the A has obtained $\{Req_j, Res_1, Res_2\}$. The A computes $N_j{*} = SK_j \oplus Req_j$, $N{*} = H(SK_j||N_j{*}) \oplus Res_1$, $K_{SC} = H(SK_j||N_j{*}||N{*})$, and $R_{ij} = Res_2 \oplus K_{SC}$. A then computes session key $SK = H(SID_j||H(N_j)||R_{ij})$.

WJL-scheme fails to provide login scalability (C6)

From the description in Section 3.1, we know that in the WJL-scheme, each time a U_i logs in to a S_j, they must submit the U_i's request message to the CS, which will authenticate the legitimacy of the U_i. In other words, the CS must interact with every login. Therefore, the WJL-scheme does not hold the scalability of login.

3.3. Review of the SAK-scheme

The SAK-scheme consists of 4 phases, i.e. registration, login, authentication and session key agreement, and password change.

1) Registration phase

The U_i registers with the *CS* by the following steps:

Step 1. $U_i \rightarrow CS$: A_i, B_i

The U_i selects a random number b; computes $A_i = H(ID_i||b)$, $B_i = H(b \oplus PW_i)$; and submits $\{A_i, B_i\}$ to the *CS* over a secure communication channel.

Step 2. $CS \rightarrow U_i$: Smart card

The *CS* selects a random number y_i and computes $F_i = A_i \oplus y_i$, $G_i = B_i \oplus H(y_i) \oplus H(x)$, and $C_i = A_i \oplus H(y_i) \oplus x$, where x is its secret master key. The *CS* stores $(y_i \oplus x,\ C_i)$ in the user database, stores $(F_i$, G_i, H( )) on the smart card, and issues the smart card to the U_i through a secure channel.

Step 3. $U_i \rightarrow$ Smart card: D_i, E_i

The U_i computes $D_i = b \oplus H(ID_i||PW_i), E_i = H(ID_i||PW_i) \oplus PW_i$ and stores them on the smart card.

When the S_j registers with the *CS*, the S_j sends its unique SK_j. The *CS* then stores $SK_j \oplus H(x||SID_j)$ in the S_j database.

2) Login phase

Step 1. Smart card checks $E_i ? = E_i*$

The U_i inserts the smart card into a card reader to log in to the S_j and keys in the ID_i*, PW_i*, and SID_j. The smart card computes $E_i* = H(ID_i*||PW_i*) \oplus PW_i*$ and verifies the legitimacy of the U_i by checking if $E_i = E_i*$.

Step 2. Smart card $\rightarrow S_j$: SID_j, Z_i, CID_i, M_i

If the equation holds, the smart card generates a random nonce N_1 and computes

$$b = D_i \oplus H(ID_i||PW_i), A_i = H(ID_i||b), B_i = H(b \oplus PW_i), \tag{4}$$

$$y_i = A_i \oplus F_i, H(x) = B_i \oplus H(y_i) \oplus G_i, Z_i = H^2(x) \oplus N_1, \tag{5}$$

$$CID_i = A_i \oplus H(y_i) \oplus H(x) \oplus N_1, M_i = H(H(x)||y_i||SID_j||N_1). \tag{6}$$

The smart card then sends the login request message $\{SID_j, Z_i, CID_i, M_i\}$ to the S_j.

3) Authentication and session key agreement phase

Step 1. $S_j \rightarrow CS$: SID_j, Z_i, CID_i, M_i, R_i

The S_j chooses a random nonce N_2, computes $R_i = SK_j \oplus N_2$, and sends $\{SID_j, Z_i, CID_i, M_i, R_i\}$ to the *CS*.

Step 2. *CS* checks $C_i ? = C_i*$

The *CS* computes $N_1 = H^2(x) \oplus C_i, N_2 = SK_j \oplus R_i, C_i* = CID_i \oplus H(x) \oplus N_1 \oplus x$ and checks whether $C_i = C_i*$. If the 2 values are equal, the *CS* goes on to Step 3. Otherwise, the *CS* rejects the login request.

Step 3. *CS* checks $M_i ? = M_i*$

The *CS* computes $M_i* = H(H(x)||y_i||SID_j||N_1)$ and checks whether $M_i ? = M_i*$. If they are not equal, the *CS* rejects the login request. Otherwise, the *CS* generates random nonce N_3 and computes:

$$K_i = N_1 \oplus N_3 \oplus H(SK_j||N_2), X_i = H(ID_i||y_i||N_1) \oplus H(N_1 \oplus N_3 \oplus N_2), \tag{7}$$

$$V_i = H(H(N_1 \oplus N_3 \oplus N_2)||H(ID_i||y_i||N_1)), T_i = N_3 \oplus N_2 \oplus H(y_i||ID_i||H(x)||N_1)). \tag{8}$$

Finally, the CS sends the message $\{K_i,\ X_i,\ V_i, T_i\}$ back to the S_j.

Step 4. S_j checks $V_i ? = V_i*$

The S_j computes $N_1 \oplus N_3 = K_i \oplus H(SK_j||N_2)$, $H(ID_i||y_i||N_1) = X_i \oplus H(N_1 \oplus N_3 \oplus N_2)$, and $V_i* = H(H(N_1 \oplus N_3 \oplus N_2)||H(ID_i||y_i||N_1))$. The S_j then verifies the legitimacy of the CS by checking whether $V_i = V_i*$. If the CS is confirmed, the S_j sends (V_i, T_i) to the smart card.

Step 5. Smart card checks $V_i ? = V_i*$

The smart card calculates $V_i* = H(H(N_1 \oplus N_3 \oplus N_2)||H(ID_i||y_i||N_1))$ and checks whether $V_i = V_i*$.

Finally, the U_i, the S_j, and the CS agree on session key $SK = H(H(ID_i||y_i||N_1)||N_1 \oplus N_3 \oplus N_2)$.

Here we omit the password change phase of the SAK-scheme.

3.4. Cryptanalyses of the SAK-scheme

In this section, we will analyze the security of the SAK-scheme. We found that the SAK-scheme does not satisfy security properties C1 (user authentication), C2 (service provider server authentication), C4 (perfect forward security), C7 (resistance to stolen verifier attacks), or C8 (scalability of login).

SAK-scheme does not hold user authentication (C1)

The SAK-scheme lacks user authentication. More specifically, the SAK-scheme cannot provide 2-factor authentication. Privileged malicious user U_k, who has registered with the CS, can use his own smart card and collect information $(F_i,\ G_i, D_i,\ E_i,\ H(\ ))$ from his own smart card. The U_k can compute:

$$b_k = D_k \oplus H(ID_k||PW_k), A_k = H(ID_k||b_k), \tag{9}$$

$$B_k = H(b_k \oplus PW_k), y_k = A_k \oplus F_k, H(x) = B_k \oplus H(y_k) \oplus G_k. \tag{10}$$

If the U_i and U_k have both registered with the same CS, they have the same parameter $H(x)$. Now the U_k can intercept a valid login request message $\{SID_j,\ Z_i,\ CID_i, M_i\}$ of the U_i over the public communication channel. Since the U_k has $H(x)$, the U_k computes $N_1 = H^2(x) \oplus Z_i$. According to the requirements of 2-factor authentication, we can further assume that the U_k extracts the information from the U_i's smart card. Thus, even if the U_k does not have knowledge of the U_i's password, the U_k can still compute secret values $\{A_i, H(y_i), y_i\}$ corresponding to the U_i. The attack is described as follows. The U_k first computes $B_i = H(D_i \oplus E_i)$, $H(y_i) = G_i \oplus B_i \oplus H(x)$, $A_i = CID_i \oplus H(y_i) \oplus H(x) \oplus N_1$, and $y_i = A_i \oplus F_i$.

The U_k then obtains these secret values $\{A_i, H(y_i), y_i\}$, corresponding to the U_i, and thus the U_k can impersonate the U_i to log in to a S_j. Here we will not describe the impersonation attack.

Furthermore, since values $\{A_i, H(y_i), y_i\}$ corresponding to the U_i are kept unchanged, after the U_k intercepts the dynamic CID_i from later login request messages, the U_k first computes $N_1 = H^2(x) \oplus Z_i$ and then calculates $A_i = CID_i \oplus H(y_i) \oplus H(x) \oplus N_1$. Thus, the U_k can identify the U_i, because values $\{A_i, H(y_i), y_i\}$ are unique for each user.

The above analyses show that the SAK-scheme cannot provide 2-factor authentication and does not meet property C1. The SAK-scheme cannot resist against insider attacks or impersonation attacks. This also illustrates that the SAK-scheme has still not removed the weaknesses of the HS-scheme (see Sections 4.2 and 4.3 in [23]).

SAK-scheme fails to provide service provider server authentication (C2)

During the authentication and session key agreement phase in the SAK-scheme, when the CS receives message $\{SID_j, Z_i, CID_i, M_i, R_i\}$ sent by the S_j, the CS verifies the legitimacy of the intended U_i by checking $C_i? = C_i*$ and $M_i? = M_i*$. However, the CS does not verify the legitimacy of the S_j. This easily leads to the same security flaws as in the WJL-scheme.

SAK-scheme fails to provide perfect forward security (C4)

Suppose that the CS's x and the S_j's database are compromised. Once an A has intercepted the message transmitted over the public channel for each run of the protocol, the A can compute the session keys. First, the A computes $Z_i = H^2(x) \oplus N_1$, $N_2 = SK_j \oplus R_i$. The A then calculates $N_3 = N_1 \oplus K_i \oplus H(SK_j||N_2)$. Note that the A may not access the user database, but the A is still able to compute $H(ID_i||\mathrm{y}_i||N_1) = X_i \oplus H(N_1 \oplus N_3 \oplus N_2)$.

The A thus obtains session key $SK = H(H(ID_i||\mathrm{y}_i||N_1)||N_1 \oplus N_3 \oplus N_2)$.

SAK-scheme fails to provide login scalability (C6)

By applying similar analyses as for the WJL-scheme in Section 3.2, one can conclude that the SAK-scheme does not hold scalability of login. In fact, from the description in Section 3.4, one can easily find that, in the SAK-scheme, the CS must check every login request of the U_i.

SAK-scheme suffers from stolen verifier attacks (C7)

During the registration phase in the SAK-scheme, the CS stores $(\mathrm{y}_i \oplus x, C_i)$ to a user database. In the user database, each record corresponds to a U_i. Likewise, when a S_j registers them with the CS, the CS selects a unique secret key SK_j for each S_j and stores $(SK_j \oplus H(x||SID_j), SID_j)$, which corresponds to the S_j in its database. Therefore, the SAK-scheme is vulnerable to stolen verifier attacks.

In a similar way, we find that the CHC-scheme [9] suffers from offline password guessing attacks, impersonation attacks, and server spoofing attacks. It fails to provide perfect forward security. For space limitations, we omit the description of these attacks.

4. A new proposed SCPA-MS scheme

In this section, we propose a new SCPA-MS scheme that removes all of the above weaknesses. The notations used in this section are listed in Table 1. The CS generates the system parameters. Let G be a generator of an elliptic curve group of prime order p. RC holds 2 master keys, x and y. The new SCPA-MS scheme is composed of 4 phases, i.e. registration, login, authentication and session key agreement, and password change. However, the login phases are different for the first-time login and the subsequent logins. The authentication and session key agreement phases are not the same for the first-time login and the subsequent logins.

4.1. Registration phase

Step 1. $U_i \rightarrow CS$: ID_i

Step 2. $CS \rightarrow U_i$: Smart card

The CS computes $A_i = H(ID_i||x)$ and stores $(A_i, ID_i, H(\ ))$ to a smart card. The CS then issues the smart card to the U_i through a secure channel.

Step 3. $U_i \rightarrow$ Smart card: SK_i'

The U_i selects a PW_i, computes $H(ID_i||x) \oplus H(ID_i||PW_i)$, and replaces A_i on the smart card with it.

Similarly, the S_j registers with the CS. At first, the S_j sends its SID_j to the CS. The CS sends $B_j = H(y||SID_j)$ to the S_j over a secure communication channel.

We describe the next 2 phases in 2 cases, respectively: on the first-time login (shown in Figures 1 and 2) and on a subsequent login (shown in Figures 3 and 4).

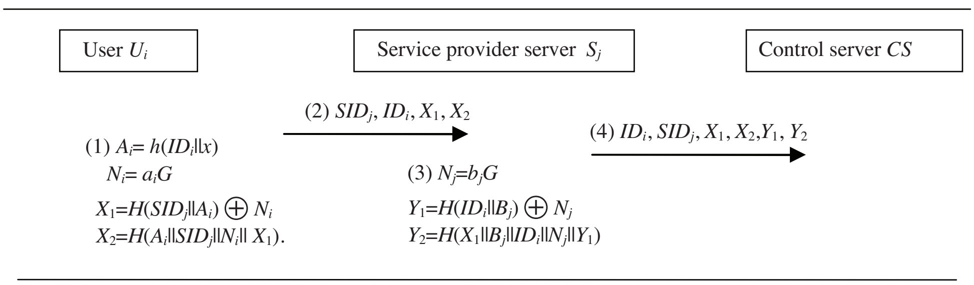

Figure 1. Login phase on the first-time login.

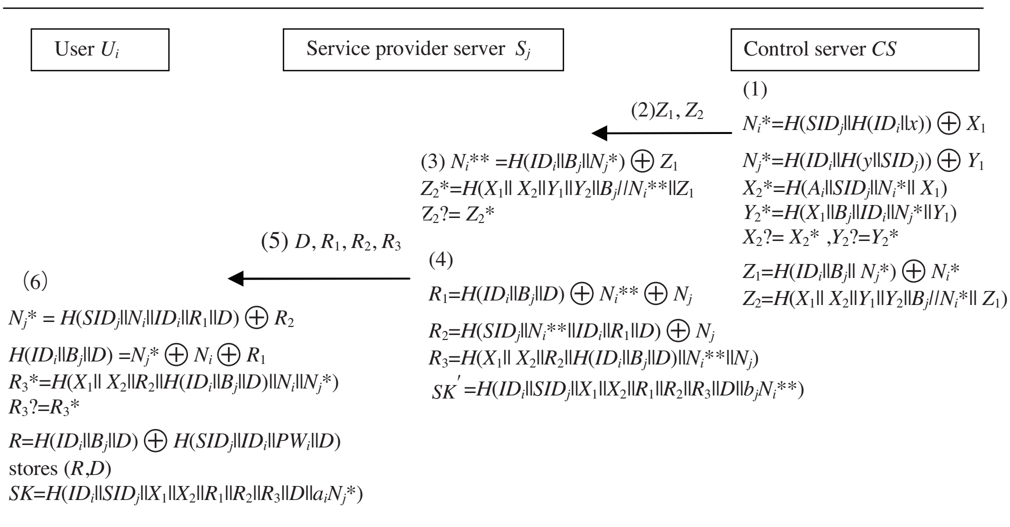

Figure 2. Authentication and session key agreement phase on the first-time login.

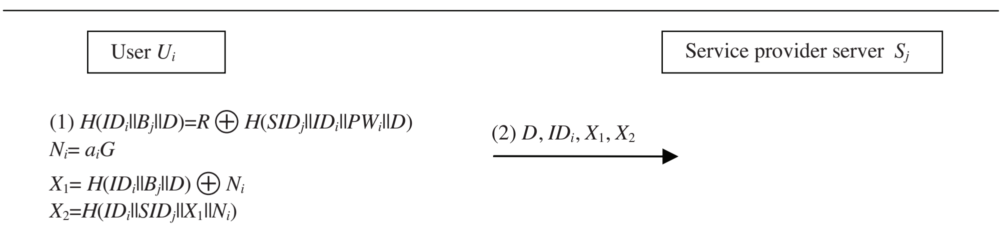

Figure 3. Login phase on a login other than the first.

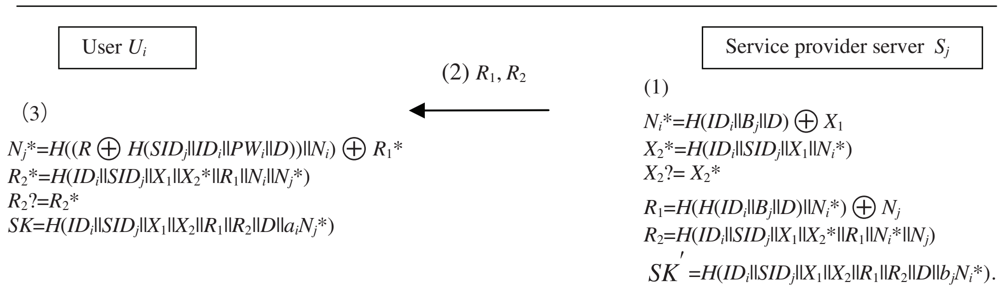

Figure 4. Authentication and session key agreement phase on a login other than the first.

4.2. Login phase on the first-time login

When the U_i logs into the S_j, the U_i inserts his smart card into a card reader and keys the ID_i, PW_i, and SID_j.

The smart card computes $A_i = (H(ID_i||x) \oplus H(ID_i||PW_i)) \oplus H(ID_i||PW_i)$. It then chooses random nonce $a_i \in Z_p^*$ and calculates:

$$N_i = a_iG, X_1 = H(SID_j||A_i) \oplus N_i, X_2 = H(A_i||SID_j||N_i||X_1). \tag{11}$$

Next, the smart card sends request message $\{SID_j, ID_i, X_1, X_2\}$ to the S_j. Upon receiving the request message, the S_j chooses random nonce $b_j \in Z_p^*$ and calculates:

$$N_j = b_jG, Y_1 = H(ID_i||B_j) \oplus N_j, Y_2 = H(X_1||B_j||ID_i||N_j||Y_1). \tag{12}$$

The S_j then transmits $\{ID_i, SID_j, X_1, X_2, Y_1, Y_2\}$ to the *CS*.

4.3. Authentication and session key agreement phase on the first-time login

Step 1. *CS* checks $X_2 ? = X_2{*}$ and $Y_2 ? = Y_2{*}$

The *CS* computes:

$$N_i{*} = H(SID_j||H(ID_i||x)) \oplus X_1, N_j{*} = H(ID_i||H(y||SID_j)) \oplus Y_1, \tag{13}$$

$$X_2{*} = H(A_i||SID_j||N_i{*}||X_1), Y_2{*} = H(X_1||B_j||ID_i||N_j{*}||Y_1), \tag{14}$$

and checks whether $X_2 ? = X_2{*}$ and $Y_2 ? = Y_2{*}$. If either of them does not hold, the *CS* rejects the login request. Otherwise, the *CS* calculates:

$$Z_1 = H(ID_i||B_j||N_j{*}) \oplus N_i{*}, Z_2 = H(X_1||X_2||Y_1||Y_2||B_j||N_i{*}||Z_1). \tag{15}$$

The *CS* then sends message (Z_1, Z_2) back to the S_j.

Step 2. S_j checks $Z_2 ? = Z_2{*}$

The S_j computes $N_i{**} = H(ID_i||B_j||N_j{*}) \oplus Z_1, Z_2{*} = H(X_1||X_2||Y_1||Y_2||B_j||N_i{**}||Z_1)$, and verifies the legitimacy of the *CS* by checking whether $Z_2 ? = Z_2{*}$. If the equation holds, the S_j determines expiration date D, and the S_j allows the U_i to log in to it without any interaction with the *CS*. Next, the S_j computes:

$$R_1 = H(ID_i||B_j||D) \oplus N_i{**} \oplus N_j, R_2 = H(SID_j||N_i{**}||ID_i||R_1||D) \oplus N_j, \tag{16}$$

$$R_3 = H(X_1||X_2||R_2||H(ID_i||B_j||D)||N_i**||N_j). \tag{17}$$

Finally, the S_j sends $\{D,\ R_1, R_2, R_3\}$ to the smart card.

Step 3. Smart card checks $R_3 ? = R_3*$

Upon receiving message $\{D,\ R_1, R_2, R_3\}$, the smart card calculates:

$$N_j* = H(SID_j||N_i||ID_i||R_1||D) \oplus R_2, H(ID_i||B_j||D) = N_j * \oplus N_i \oplus R_1, \tag{18}$$

$$R_3* = H(X_1||X_2||R_2||H(ID_i||B_j||D)||N_i||N_j*). \tag{19}$$

The smart card checks if $R_3 = R_3*$. If the above equality holds, the smart card computes $R = H(ID_i||B_j||D) \oplus H(SID_j||ID_i||PW_i||D)$ and stores (R, D) to the card. The U_i then computes the session key:

$$SK = H(ID_i||SID_j||X_1||X_2||R_1||R_2||R_3||D||a_iN_j*). \tag{20}$$

The S_j also can obtain the following session key:

$$SK' = H(ID_i||SID_j||X_1||X_2||R_1||R_2||R_3||D||b_jN_i**). \tag{21}$$

If all of the participants follow the protocol, then $a_iN_j* = b_jN_i**$. This implies that $SK = SK'$. Thus, we have confirmed the correctness of the proposed SCPA-MS scheme.

4.4. Login phase on a subsequent login

The U_i inserts the smart card into a card reader and keys the ID_i, PW_i, and SID_j. The smart card then carries out the following steps:

1) Compute $H(ID_i||B_j||D) = R \oplus H(SID_j||ID_i||PW_i||D)$.

2) Select random nonce $a_i \in Z_p^*$ and calculate

$$N_i = a_iG, X_1 = H(ID_i||B_j||D) \oplus N_i, X_2 = H(ID_i||SID_j||X_1||N_i). \tag{22}$$

3) Send request message $\{D,\ ID_i, X_1, X_2\}$ to the S_j.

4.5. Authentication and session key agreement phase on a subsequent login

After receiving request message $\{D,\ ID_i, X_1, X_2\}$, the S_j executes the following steps:

1) Check if the login has expired via D.

2) Compute $N_i* = H(ID_i||B_j||D) \oplus X_1, X_2* = H(ID_i||SID_j||X_1||N_i*)$.

3) Check $X_2 ? = X_2*$. If the equation holds, the S_j continues.

4) Compute

$$R_1 = H(H(ID_i||B_j||D)||N_i*) \oplus N_j, R_2 = H(ID_i||SID_j||X_1||X_2*||R_1||N_i*||N_j). \tag{23}$$

5) Send $\{R_1, R_2\}$ to the smart card.

Upon receiving message $\{R_1, R_2\}$, the smart card calculates:

$$N_j* = H((R \oplus H(SID_j||ID_i||PW_i||D))||N_i) \oplus R_1*, \tag{24}$$

$$R_2* = H(ID_i||SID_j||X_1||X_2*||R_1||N_i||N_j*). \tag{25}$$

The smart card checks if $R_2 = R_2*$. If the above equality holds, the smart card computes the session key:

$$SK = H(ID_i||SID_j||X_1||X_2||R_1||R_2||D||a_iN_j*). \tag{26}$$

The service provider server computes the following session key:

$$SK' = H(ID_i||SID_j||X_1||X_2||R_1||R_2||D||b_jN_i*). \tag{27}$$

It is easy to confirm the correctness of the proposed SCPA-MS scheme.

4.6. Password change phase

If the U_i wants to change his PW_i, the U_i inserts the smart card and keys in the ID_i and PW_i. The U_i carries out the following steps:

1) Select a new password, PW_i'.
2) Compute $(H(ID_i||x) \oplus H(ID_i||PW_i)) \oplus H(ID_i||PW_i) \oplus H(ID_i||PW_i')$.
3) Replace $(H(ID_i||x) \oplus H(ID_i||PW_i))$ with the above result on the smart card.

5. Analyses of the proposed SCPA-MS scheme

5.1. Security analyses

In the following section, we analyze our SCPA-MS scheme according to the security requirements given in Section 2.2. We also demonstrate that our scheme can resist against some well-known security threats.

Theorem 1. The proposed SCPA-MS scheme holds user authentication (C1).

Proof. In the proposed scheme, since the CS can authenticate the U_i by checking the validity of information $\{ID_i, X_1, X_2\}$, the undetectable online password guessing attack will not work. The password is not applied to compute any authentication messages. Therefore, no A can guess the password successfully from the authentication message.

On the other hand, a malicious S_j can get $\{N_i, X_1, X_{2,}, Z_1, Z_2\}$, but $\{N_i, X_1, X_{2,}, Z_1, Z_2\}$ does not contain any information about the U_i's password and malicious server S_k cannot guess the U_i's password from this information. Hence, our improved scheme can resist against the offline guessing attack by a S_k.

Without loss of generality, assume that an A obtains a smart card and the password of the card owner is unknown to the A. The A can then extract the information stored on the smart card. The password is protected on the card as $h(ID_i||x) \oplus h(ID_i||PW_i)$ while $h(ID_i||x)$ is contained in $\{X_1, X_2, Z_2\}$, but X_1 and X_2 are provided with a random nonce, N_i. Z_2 is mixed with random nonce N_i. It is infeasible to extract $h(ID_i||x)$ from $\{X_1, X_2\}$ or Z_2. Therefore, an A cannot obtain a verification function about the password from the stored information $h(ID_i||x) \oplus h(ID_i||PW_i)$ or the transmitted information.

The above analyses show that the proposed scheme can resist against password guessing attacks and provide 2-factor security.

We now consider the impersonation attack in the worst case: a S_k attempts to impersonate a U_i who has never accessed the S_j. In order to impersonate the U_i to log in other $S_j s$, the S_k computes request message $\{X_1, X_2\}$. However, the message depends on A_i. The S_j could compute T_i only through 2 approaches. One approach is to compute it from the U_i's past request message, but the S_j cannot obtain $h(ID_i||x)$ since $h(ID_i||x)$ is protected by random number N_i. The other approach is to compute it from the past confirmation message, $\{Z_1, Z_2\}$, sent by the *CS*. However, the A still cannot obtain $h(ID_i||x)$, because $h(ID_i||x)$ is mixed with random nonces N_i and N_j.

Therefore, the proposed scheme can resist impersonation attacks. It provides strong user authentication.

Theorem 2. The proposed SCPA-MS scheme provides service provider server authentication (C2).

Proof. Assume that a S_k tries to cheat a U_i or *CS* by masquerading as a S_j. Although the A has its secret value B_k, since $H(\ )$ is a collision-resistant hash function, the S_k still cannot compute secret key B_j of the S_j. Thus, the A cannot produce valid pair $\{Y_1, Y_2\}$ (on the first-time login) or $\{R_1, R_2\}$ (on a subsequent login). In other words, the A cannot fool the *CS* or U_i by masquerading as a S_j.

Next, the A also cannot compute N_i from $H(ID_i||B_j||N_j) \oplus Z_1$. It is thus infeasible to compute $\{R_1, R_2, R_3\}$ such that $R_3 = H(X_1||X_2||R_2||H(ID_i||B_j||D)||N_i||N_j{}^*)$ holds.

Hence, the server spoofing attacks fail. The proposed SCPA-MS scheme provides service provider server authentication.

Theorem 3. The proposed SCPA-MS scheme provides control server authentication (C3).

Proof. We first show that the proposed scheme can prevent any A from obtaining the *CS*'s secret master key. The secret key x is hashed in the form $h(ID_i||x)$ and the secret key y is hashed into the form $h(y||SID_j)$. During the transmission of the above values, they are hashed by adding some random integers and other information. Upon the assumptions of collision-resistant hash functions, an A cannot extract x or y from $h(ID_i||x)$ or $h(SID_j||y)$.

Next, it is infeasible that an A cheats a U_i or a S_j by masquerading as a *CS*. Since the A does not have x or y, the A cannot compute A_i or B_j. Therefore, it is infeasible to compute N_i from $H(SID_j||H(ID_i||x)) \oplus X_1$ or N_j from $H(ID_i||H(y||SID_j)) \oplus Y_1$. Thus, the A cannot generate valid pair $\{Z_1, Z_2\}$. When an A sends a forged authentication message, the S_j will find that the information is not from the *CS*. Likewise, the smart card can identify that the message is forged, since verification equation $R_3 = H(X_1||X_2||R_2||H(ID_i||B_j||D)||N_i||N_j)$ will not hold. Therefore, the proposed scheme can resist control server spoofing attacks.

Theorem 1, Theorem 2, and Theorem 3 imply that the proposed SCPA-MS scheme can also resist against man-in-the-middle attacks.

Theorem 4. Upon the computational Diffie-Hellman assumptions, the proposed SCPA-MS scheme holds perfect forward security (C4).

Proof. Assume that the *CS*'s master secret keys x and y are disclosed. The A attempts to learn a used session key. The A can intercept all of the messages transmitted among the U_i, the S_j, and the *CS*.

1) First-time login

With knowledge of x, the A can compute:

$$A_i = H(ID_i||x), N_i = H(SID_j||H(ID_i||x)) \oplus X_1. \tag{28}$$

Likewise, the A uses y to compute:

$$B_j = H(y||SID_j), N_j = H(ID_i||B_j) \oplus Y_1. \tag{29}$$

However, the A still cannot work out:

$$SK = H(ID_i||SID_j||X_1||X_2||R_1||R_2||R_3||D||a_iN_j). \tag{30}$$

This is because the A has to work out a_iN_j or b_jN_i. The A will be confronted with an instance, (N_i, N_j, a_iN_j) or (N_i, N_j, b_jN_i), of computational Diffie-Hellman problems.

2) Subsequent login

With knowledge of y, the A can compute:

$$N_i = H(ID_i||H(y||SID_j)||D) \oplus X_1, N_j = H(H(ID_i||B_j||D)||N_i) \oplus R_1. \tag{31}$$

However, in order to calculate the session key,

$$SK = H(ID_i||SID_j||X_1||X_2||R_1||R_2||D||a_iN_j), \tag{32}$$

the A will still be confronted with a computational Diffie-Hellman problem, (N_i, N_j, a_iN_j) or (N_i, N_j, b_jN_i).

Therefore, the proposed scheme provides perfect forward security.

The proof of Theorem 4 also shows that in the proposed SCPA-MS scheme, both the U_i and the S_j agree on the session key. Meanwhile, on the computational Diffie-Hellman assumptions, the probability that the A computes other session keys from one session key is still negligible. Next, the proposed scheme is secure against replay attacks. Even if the *CS* confirms that the replay message is valid, on the assumptions of Diffie-Hellman, the A still cannot compute a fresh session key.

It is easily confirmed that our proposed scheme satisfies the following properties, C5-C8.

Freedom of password change (C5)

In the proposed SCPA-MS scheme, the U_i can change the password freely without any interaction with the S_j or the *CS*.

Scalability of login (C6)

In the proposed SCPA-MS scheme, when a U_i has finished the first-time login to a S_j, the U_i stores (R, D) to the card. The next time that the U_i logs in to the S_j before the D, the U_i interacts only with the S_j to agree to a new session key, without interaction with the *CS*. To some extent, D helps the S_j control the login of the U_i. In addition, the S_j can also introduce the number of logins to control the login of the U_i, without any interaction with the *CS*.

Resistance to the stolen verifier attacks (C7)

Since the proposed SCPA-MS scheme does not maintain a user verification table or a password table in the *CS* or the S_j, and does not maintain a S_j database in the *CS*, no U_i or S_j verifiable information can be obtained from the *CS*. Thus, the proposed scheme can prevent the stolen verifier attack.

High efficiency (C8)

In the proposed SCPA-MS scheme, if a U_i has registered with the CS once, the U_i can access all of the eligible S_js registered with the same CS. The proposed protocol is efficient because only the one-way hash functions, exclusive-OR (XOR) operations, and 2 scalar multiplications in the elliptic curve group are required by the smart card. The detailed efficiency analyses will be shown in Section 5.2.

5.2. Performance and functionality analyses

Due to the resource constraints of the smart card, the SCPA scheme must take efficiency into consideration. In this section, we will evaluate the performance of the proposed SCPA-MS scheme and make comparisons with some SCPA-MS schemes in Table 2. We evaluate the efficiency in terms of both communication cost and computation cost. Assume that ID_i, SID_j, x and y, D, PW_i, the nonce values, and the timestamp are 128 bits long; the large prime in the modular operation in the elliptic curve group is 160 bits long, and the large prime in the modular operation in other groups is 1024 bits long, as in practical implementation. Moreover, we also assume that both the size of the output of the secure one-way hashing function $H(\ )$ and the block size of the secure symmetric cryptosystem are 128 bits. To analyze the computational complexity of the schemes, we define the notation T_h, T_m, T_e, and T_{sy} as the time cost for 1 hash operation, 1 scalar multiplication in the elliptic curve group, 1 modular exponentiation, and 1 symmetric encryption/decryption, respectively. Because the XOR operation requires very few computations, its computational cost is usually neglected. Let T_M and T_{inv} be the time cost of 1 scalar multiplication and 1 inversion operation in Z_p, respectively.

In our scheme, the parameters stored in the smart card are $\{A_i \oplus H(ID_i || PW_i),\ ID_i,\ H(\ )\}$ and $\{R, D\}$ (on logins other than the first), so the memory needed (E1) on the smart card is, at most, 640 (=128 × 5) bits. The communication cost includes the capacity of transmitting the message involved in the login phase and the authentication and session key agreement phase. The U_i is required to transmit $\{SID_j, ID_i, X_1, X_2\}$ (on the first-time login) or $\{D, ID_i, X_1, X_2\}$ (on any subsequent login). The S_j needs to transmit $\{ID_i, SID_j, X_1, X_2, Y_1, Y_2, D, R_1, R_2, R_3\}$ (on the first-time login) or $\{R_1, R_2\}$ (on any subsequent login). The CS is required to transmit (Z_1, Z_2) to the S_j only on the first-time login. Hence, the communication

Table 2. Communication cost and computation cost comparison.

	E1	E2	E3	E4	E5	E6
Ours (case 1)	384 bits	2048 bits	$3T_h$	$8T_h + 2T_m$	$8T_h + 2T_m$	$4T_h$
Ours (case 2)	640 bits	768 bits	$3T_h$	$3T_h + 2T_m$	$5T_h + 2T_m$	0
Sandeep et al. [23]	640 bits	1920 bits	$5T_h$	$10T_h + T_e$	$5T_h$	$9T_h + T_e$
Wang et al. [21]	3200 bits	10496 bits	$3T_h$	$6T_h + T_e$	$6T_h$	$8T_h + T_{sy}$
Chen et al. [9] (case 1)	256 bits	11904 bits	$3T_h$	$5T_h + 3T_e$	$4T_h + 3T_e$	$6T_h$
Chen et al. [9] (case 2)	1280 bits	4480 bits	$3T_h$	$3T_h + 2T_e$	$3T_h + 2T_e$	0
Hsiang et al. [22]	768 bits	2176 bits	$7T_h$	$11T_h$	$9T_h$	$5T_h$
Liao et al. [8]	640 bits	896 bits	$5T_h$	$9T_h$	$6T_h$	0
Tsai [20]	384 bits	1664 bits	$2T_h$	$5T_h$	$7T_h$	$6T_h$
Juang [16]	256 bits	1152 bits	$2T_h + 2T_{sy}$	$3T_h + 3T_{sy}$	$3T_h + 6T_{sy}$	$T_h + 2T_{sy}$
Lin et al. [28]	5120 bits	7424 bits	$5T_e + 4T_M + 2T_{inv}$	$2T_e$	$7T_e + T_M$	0

Note: E2 denotes the communication cost of the login phase and the authentication and session key agreement phase. E3 denotes the computation cost of a U_i and one S_j.

cost (E2) of the login phase and the authentication and session key agreement phase is a total of 2048 (=128 $\times$ 16) bits on the first-time login or 768 (=128 $\times$ 6) bits for subsequent logins.

The computation cost of registration is defined as the total time of various operations in the registration phase. In our SCPA-MS scheme, the computation cost (E3) of registration is $3T_h$. We discuss the E3 in 2 cases: on the first-time login, the computation costs of the smart card (E4), the service provider server (E5), and the control server (E6) are $8T_h + 2T_m$, $8T_h + 2T_m$, and $4T_h$, respectively; and on subsequent logins, the E4 and the E5 are $3T_h + 2T_m$ and $5T_h + 2T_m$, respectively.

Of all of the previous SCPA-MS schemes in the literature, the LW-scheme [8] has higher efficiency and does not require any interaction of the *CS* each time the U_i logs in to the S_j. However, all of the S_js know the y of the *CS*. Moreover, after the U_i logs in to a S_j, the S_j can obtain the U_i's private information, $H(ID_i||x)$. These flaws will lead to a lack of user authentication and control server authentication. In essence, the LW-scheme suffers from control server spoofing and service provider server spoofing [21]. The Lin et al. scheme [28] does not require the interaction of the *CS* for the U_i to log in, but the use of public keys makes its E2 and E3 very high. Table 2 shows that our SCPA-MS scheme is very efficient. Especially compared with the SCPA-MS schemes that keep the U_i's login free of interaction of the *CS*, our scheme requires less computations.

We summarize the proposed scheme and make comparisons with some SCPA-MS schemes. The comparison results for security requirements are summarized in Table 3. These demonstrate that our schemes can achieve the essential requirements for a secure SCPA-MS scheme. The WJL-scheme [21] requires a smart card with a large memory and high E2. The CHC-scheme [9] also needs a high E2. Therefore, the 2 schemes cannot provide property C8. In addition, the HS-scheme [22], Tsai scheme [20], and Juang scheme [16] require that the CS participate in every U_i login. Therefore, none of these schemes can provide property C6. The analyses of other properties of these schemes can be found in [8,16-23].

Table 3. Functionality comparison.

	C1	C2	C3	C4	C5	C6	C7	C8
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sandeep et al. [23]	No	No	Yes	No	Yes	No	No	Yes
Wang et al. [21]	Yes	No	Yes	No	Yes	No	Yes	No
Chen et al. [9]	No	No	Yes	No	Yes	Yes	Yes	No
Hsiang et al. [22]	No	No	Yes	Yes	No	No	Yes	Yes
Liao et al. [8]	No	No	No	No	Yes	Yes	Yes	Yes
Tsai [20]	No	Yes	No	No	Yes	No	No	Yes
Juang [16]	No	Yes	Yes	Yes	No	No	Yes	Yes
Lin et al. [28]	No	No	Yes	Yes	No	Yes	Yes	No

6. Conclusion

SCPA in MS environments is important in the real user-server world. In this paper, we presented the cryptanalyses of 2 identity-based smart card authentication schemes for MS environments. We found that the schemes of Wang et al. and Sandeep et al. failed to provide service provider server authentication and perfect forward security. Moreover, they lacked login scalability. In addition, the scheme of Sandeep et al. suffers from stolen verifier attacks. An improved SCPA-MS scheme was proposed. The proposed SCPA-MS scheme inherits the merits of the schemes of Wang et al. and Sandeep et al. It removes their weaknesses. Security analyses demonstrate that the proposed SCPA-MS protocol can withstand various possible attacks and satisfies all of the security requirements. The functionality comparison confirms the advantages of our scheme in contrast to the previous SCPA-MS schemes. Our SCPA-MS scheme is practical and efficient. Dynamic identity-based authentication protocols can provide anonymity and protection of the login user. However, the WJL-scheme and the SAK-scheme have some security defects. Future work must be done to design secure dynamic identity SCPA-MS schemes in which the verification message is not required in the *CS*.

Acknowledgment

This work was partially supported by a grant from the National Natural Science Foundation of China (10961013 and 61163053), 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 and the editors.

References

[1] W. Diffie, M.E. Hellman "New directions in cryptography", IEEE Transactions on Information Theory, Vol. 22, pp. 644654, 1976

[2] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, Vol. 24, pp. 28-30, 1981.

[3] K. Bıçakcı, N. Baykal, "Improving the security and flexibility of one-time passwords by signature chains", Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 11, pp. 223-236, 2003.

[4] M.S. Hwang, L.H. Li, "A new remote user authentication scheme using smart cards", IEEE Transactions on Consumer Electronics, Vol. 46, pp. 28-30, 2000.

[5] L. Li, I. Lin, M. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks", IEEE Transactions on Neural Networks, Vol. 12, pp. 1498-1504, 2001.

[6] M.L. Das, A. Saxena, V.P. Gulati, "A dynamic ID-based remote user authentication scheme", IEEE Transactions on Consumer Electronics, Vol. 50, pp. 629-31, 2004.

[7] L.L. Hu, X.X. Niu, Y.X. Yang, "An efficient multi-server password authenticated key agreement scheme using smart cards", International Conference on Multimedia and Ubiquitous Engineering, pp. 903-907, 2007.

[8] Y.P. Liao, S.S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards & Interfaces Vol. 31, pp. 24-29, 2009.

[9] Y.L. Chen, C.H. Huang, J.S. Chou, "A novel multi-server authentication scheme", Cryptology ePrint Archive, pp. 161-190, 2009.

[10] E.J. Yoon, K.Y. Yoo, "Robust multi-server authentication scheme", Sixth IFIP International Conference on Network and Parallel Computing, pp. 197-203, 2009.

[11] W.B. Lee, C.C. Chang, "User identification and key distribution maintaining anonymity for distributed computer network", Computer System Science, Vol. 15, pp. 211-214, 2000.

[12] W. Ford, B.S. Kaliski, “Server-assisted generation of a strong secret from a password”, Proceedings of IEEE 9th International Workshop Enabling Technologies, pp. 176-180, 2000.

[13] W.J. Tsaur, C.C. Wu, W.B. Lee, “A flexible user authentication scheme for multi-server Internet services”, Lecture Notes in Computer Science, Vol. 2093, pp. 174-183, 2001.

[14] W.J. Tsaur, C.C. Wu, W.B. Lee, “A smart card-based remote scheme for password authentication in multi-server internet services”, Computer Standards & Interfaces, Vol. 27, pp. 39-51, 2004.

[15] J.S. Chou, C.H. Huang, C.C. Ding, “Security weaknesses in two multi-server password based authentication schemes”, Nanhua University White Paper, available at eprint.iacr.org/2009/338.pdf.

[16] W.S. Juang, “Efficient multi-server password authenticated key agreement using smart cards”, IEEE Transactions on Consumer Electronics, Vol. 50, pp. 251-255, 2004.

[17] C.C. Chang, J.S. Lee, “An efficient and secure multi-server password authentication scheme using smart cards”, International Conference on Cyber Worlds, pp. 417-422, 2004.

[18] J.H. Lee, D.H. Lee, “Efficient and secure remote authenticated key agreement scheme for multi-server using mobile equipment”, Proceedings of the International Conference on Consumer Electronics, pp. 1-2, 2008.

[19] M.H. Lim, S. Lee, H. Lee, “An efficient multi-server password authenticated key agreement scheme revisited”, Third International Conference on Convergence and Hybrid Information Technology, pp. 396-400, 2008.

[20] J.L. Tsai, “Efficient multi-server authentication scheme based on one-way hash function without verification table”, Computers & Security, Vol. 27, pp. 115-121, 2008.

[21] R.C. Wang, W.S. Juang, C.L. Lei, “User authentication scheme with privacy-preservation for multi-server environment”, IEEE Communications Letters, Vol. 13, pp. 157-159, 2009.

[22] H.C. Hsiang, W.K. Shih, “Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment”, Computer Standards & Interfaces, Vol. 31, pp. 1118-1123, 2009.

[23] K.S. Sandeep AK. Sarje, K. Singh, “A secure dynamic identity based authentication protocol for multi-server architecture”, Journal of Network and Computer Applications, Vol. 34, pp. 609-618, 2011.

[24] Y. Yang, R.H. Deng, F. Bao, “A practical password-based two-server authentication and key exchange system”, IEEE Transactions on Dependable and Secure Computing Vol. 3, pp. 105-114, 2006

[25] T.S. Messerges, E.A. Dabbish, R.H. Sloan, “Examining smart-card security under the threat of power analysis attacks”, IEEE Transactions on Computers, Vol. 51, pp. 541-552, 2002.

[26] P. Kocher, J. Jaffe, B. Jun, “Differential power analysis”, Lecture Notes in Computer Science, Vol. 1666, pp. 388-397, 1999.

[27] K.A. Bayam, B. Örs, “Differential power analysis resistant hardware implementation of the RSA cryptosystem”, Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 18, pp. 129-140, 2010.

[28] C. Lin, M.S. Hwang, L.H. Li, “A new remote user authentication scheme for multi-server architecture”, Future Generation Computer Systems, Vol. 1, pp. 13-22, 2003.