Hindawi
Security and Communication Networks
Volume 2018, Article ID 9178941, 13 pages
https://doi.org/10.1155/2018/9178941

Research Article

Secure and Efficient User Authentication Scheme Based on Password and Smart Card for Multiserver Environment

Yan Zhao,[1,2] **Shiming Li**,[3] **and Liehui Jiang**[1]

[1]*State Key Laboratory of Mathematic Engineering and Advanced Computing, Zhengzhou 450002, China*
[2]*College of Physical and Electronic Information, Luoyang Normal University, Luoyang 471022, China*
[3]*College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China*

Correspondence should be addressed to Yan Zhao; yanzhao_ly@163.com

Received 15 January 2018; Revised 28 March 2018; Accepted 18 April 2018; Published 20 May 2018

Academic Editor: Kim-Kwang Raymond Choo

The rapid development of information and network technologies motivates the emergence of various new computing paradigms, such as distributed computing, cloud computing, and edge computing. This also enables more and more network enterprises to provide multiple different services simultaneously. To ensure these services can only be accessed conveniently by authorized users, many password and smart card based authentication schemes for multiserver architecture have been proposed. Recently, Truong et al. introduced an identity based user authentication scheme on elliptic curve cryptography in multiserver environment and claimed that their scheme is secure against popular attacks. However, in this paper, we point out that their scheme suffers from offline password guessing and impersonation attack and fails to achieve security requirements of this kind of authentication scheme. Moreover, we put forward a new scheme to conquer security pitfalls in the above scheme. Security analysis indicates that the proposed scheme can be free from well-known attacks. Performance discussion demonstrates that our scheme has advantages in terms of both security property and computation efficiency and thus is more desirable for practical applications in multiserver environment.

1. Introduction

The authentication and key agreement protocol is one of fundamental building blocks for securing communications over the Internet. It enables protocol participants to authenticate each other's identities and establish shared session keys subsequently used by encryption algorithms and is widely implemented in many areas, such as online-shopping, Internet banking, electronic governance, and electronic medical record system.

Roughly speaking, the above application scenarios can be abstracted to a user-server model. That is, when a user wants to remotely access the service provided by a server, he/she first registers with the service provider. Then, the service provider can ensure that the service can only be accessed by legitimate users; meanwhile, the user can believe that the service provider is legal. So far, there are many kinds of authentication scheme that are applicable to the user-server setting, such as certificate-based authentication scheme [1, 2], identity-based authentication scheme [3–5], and password-based authentication scheme [6, 7].

Among these variants of authentication scheme, password-based authentication scheme is particularly attractive due to its unique features, i.e., the password is easy to remember and the scheme is conveniently to be deployed. Specifically, in the context of this kind of authentication scheme, each user possesses a personal password, as the credential of accessing the service provider by a server. At the same time, the service provider maintains a table to verify the validity of all user's passwords such that invalid users' access request would be rejected. However, this also makes the scheme vulnerable to offline password guessing attack, especially when the verification table is disclosed. To conquer this issue, smart card is introduced into the design of password-based authentication scheme, which results in password and smart card based two-factor authentication scheme. Such an authentication provides stronger security guarantee; namely, even if the password or the smart card (not

the both) gets exposed, the scheme can remain secure. Since the introduction of this kind of two-factor authentication scheme, a lot of schemes [8–14] based on different cryptography primitives have been proposed. Particularly, these schemes are designed for the single server environment.

On the other hand, the rapid development of information and network technologies brings a number of new information systems, e.g., social networks, wireless sensor networks, and cloud computing, which can provide multiple services simultaneously. To solve the access control problem in the setting of multiple service providers, we can concurrently implement multiple instances of a password and smart card based authentication scheme designed in the single server environment. However, for a system user, this will bring tremendous workload of managing passwords and smart cards issued by different service providers. In addition, it also increases the damage of password disclosure.

To improve the usability of password and smart card based authentication scheme, researchers propose to design this kind of authentication scheme for multiserver architecture. Informally, in the improved scheme, each user just needs to register with a registration center and then can access any service provided by those servers managed by the registration center. Specifically, Yeh [15] recently proposed such authentication scheme based on RSA cryptosystem and proved its security in the random oracle model. However, Truong et al. [16] found that Yeh's scheme fails to provide mutual authentication and key agreement, which are basic security requirements of an authentication scheme. Furthermore, they proposed a new scheme to conquer these security pitfalls. Their scheme is built upon elliptic curve cryptography and is claimed to be secure against various attacks. Unfortunately, in this paper, we will demonstrate that Truong et al.'s [16] scheme cannot resist impersonation attack and offline password guessing attack, which is the most realistic and serious threat against this kind of authentication scheme. In addition, we also put forward a security enhanced password and smart card based authentication scheme in multiserver environment. The security analysis and performance discussion indicate that our scheme has advantages in terms of both security property and computation efficiency and thus are more desirable for practical applications.

1.1. Related Work. In 1981 Lamport [17] proposed the first password authentication scheme. This scheme is built upon cryptographically secure one-way hash function and has advantages of simplicity and convenience. However, it inevitably suffers from password guessing attack and the threat of the disclosure of the verification table. To enhance the security of password-based authentication scheme, Chang and Wu [18] introduced password and smart card based two-factor remote user authentication scheme. Since then, a number of such schemes [19–27] have been proposed to improve the security and efficiency of this kind of authentication scheme. In general, these schemes fall into two types, i.e., using static identity or dynamic identity. The main drawback of using static identity is that publicly transmitted identity will reveal user privacy. To conquer this issue, Das et al. [19] introduced the notion of password and smart card authentication scheme using dynamic identity and proposed a concrete protocol. However, Liao et al. [28] pointed out that this scheme cannot resist user impersonation attack and also proposed an improved scheme with mutual authentication. Subsequently, although there are various similar schemes designed to fix security pitfalls in previous schemes, most of them [20–22] are still vulnerable to offline password guessing attack when the smart card is lost.

Today, with the rapid development of information and network technologies, more and more network enterprises can simultaneously provide multiple different kinds of services. If we directly use those authentication schemes designed for the single server environment, then a user has to register with all of service providers, which will bring heavy workload for the user to manage all passwords and identities. To solve this problem, Li et al. [29] proposed a password authentication scheme based on neural network in multiserver environment and claimed that one registration enables a user to access all of services. Subsequently, Lin et al. [30] gave a new scheme based on ElGamal signature to improve the efficiency of Li et al.'s scheme. Moreover, Juang [31] further used hash function and symmetric encryption algorithm to decrease the computation cost of this kind of authentication. However, Ku et al. [32] found that Juang's scheme cannot resist insider attack and also cannot support perfect forward secrecy.

To enhance the security of the above password-based authentication schemes for multiserver environment, in 2009 Liao and Wang [33] proposed the first password and smart card based authentication scheme in the multiserver environment using dynamic identity. But Hsiang and Shih [34] immediately noted that Liao et al.'s scheme is vulnerable to inside attack, impersonation attack, and forgery attack. Although Hsiang and Shih gave an improved scheme, Sood et al. [35] found that Hsiang and Shih's scheme is susceptible to replay attack, impersonation attack, and stolen smart card attack. Recently, motivated by security requirements from different areas, a few of new two-factor authentication schemes [15, 16, 36–40] for multiserver environment have been put forward. These schemes are mainly built upon elliptic curve cryptosystem. In addition, there are several works that introduce biometrics into the design of authentication scheme for multiserver environment. For example, Odelu et al. [41] proposed a secure multiserver authentication protocol using biometric-based smart card. He and Wang [42] presented a biometrics-based three-factor authentication scheme for multiserver environment using elliptic curve cryptography. Moreover, there are a few similarly schemes [43–46] that are put forward recently. Although there have been various multifactor authentication schemes for multiserver environment, how to design a secure and efficient authentication scheme remains challenging.

1.2. Outline. The remainder of this paper is organized as follows. Section 2 briefly reviews Truong et al.'s [16] authentication scheme. Two kinds of practical attack against their scheme are provided in Section 3. We propose a security enhanced password and smart card based authentication scheme in multiserver environment in Section 4 and present

Table 1: The notations used throughout this paper.

Symbol	Description
RC	The registration center
U_i	The ith system user
S_j	The jth service provider
PW_i	The user U_i's personal password
$H_1(\cdot), H_2(\cdot)$	Secure one-way hash functions
$\oplus$	The exclusive or operation
x_i, y_j	Random integers picked by U_i and S_j
T_i, T_j	The timestamps on sides of U_i and S_j
α	The master secret key of RC
UID_i	The user U_i's identity
SID_j	The server S_j's identity
$ASID_j$	The server S_j's secret key
$[m]$	The integer set $\{1, 2, \ldots, m\}$
$\|$	The concatenation operation
$SK_{i,j}$, $SK_{j,i}$	Shared session key
ΔT	The maximum delay allowed

the corresponding security analysis in Section 5. Section 6 discusses the performance of the proposed scheme. Finally, we give the conclusion in Section 7.

2. Review of Truong et al.'s Scheme

In this section, we briefly review Truong et al.'s [16] scheme. We summarize the notations used throughout this paper in Table 1. Specifically, Truong et al.'s authentication scheme is comprised of the following four phases.

2.1. Initialization Phase. In this phase, the registration center RC is given a security parameter κ and initializes the system as follows:

(1) Select an elliptic curve $E_p(a, b)$ defined over $\mathbb{Z}_p^*$, where p is a prime number of size κ and $a, b \in \mathbb{Z}_p$. Let $\mathbb{G}$ be a cyclic group derived from $E_p(a, b)$ with prime order q and let $g \in \mathbb{G}$ be a random generator.

(2) Randomly choose $\alpha \in \mathbb{Z}_q^*$ and select two hash functions $h_1 : \{0, 1\}^* \to \{0, 1\}^n$, $h_2 : \{0, 1\}^* \to \mathbb{G}$.

(3) Publish the system public parameter as $\mathsf{pp} = \{E_p(a, b), g, h_1(\cdot), h_2(\cdot)\}$, and keep $\mathsf{msk} = \alpha$ as the secret key.

2.2. Registration Phase. This phase consists of two parts, i.e., server registration and user registration. First, when a service provider S_j intends to register with the registration center RC, as indicated in Figure 1, they interactively perform as follows:

(1) The service provider S_j chooses an identity SID_j and submits it to the registration center RC through a secure channel.

(2) After receiving S_j's registration request, the registration center RC picks a random integer $z_j \in \mathbb{Z}_q$ and computes $ASID_j = \alpha \cdot h_2(SID_j \| z_j)$ and then sends $\{z_j, SID_j, ASID_j, h_1(\cdot), h_2(\cdot)\}$ to the service provider S_j via a secure channel.

(3) Upon getting the registration center RC's response message, the service provider S_j keeps $ASID_j$ as its master secret key.

Second, as shown in Figure 2, a user U_i intending to register with the registration center RC carries out the following steps:

(1) The user U_i chooses an identity UID_i and submits it to the registration RC via a secure channel.

(2) After receiving the registration request from U_i, the registration center RC randomly selects PW_i and $r_i \in \mathbb{Z}_q^*$. Then, for each $j \in [m]$, the registration center RC computes $s_{i,j} = h_2(UID_i \| r_i) + ASID_j$ and $RPW_i = h_1(PW_i \| UID_i \| r_i)$.

(3) The registration center RC returns a smart card including the secret information $\{RPW_i, h_1(\cdot), r_i, \{s_{i,j}\}_{j=1}^m\}$ and PW_i to the user U_i.

(4) Upon getting RC's response information, the user U_i immediately updates the initial password chosen by RC.

2.3. Login-In and Authentication Phase. When a user U_i wants to access the service from a provider S_j, they need to interactively perform an authentication procedure to ensure the provided service is legally accessed. As shown in Figure 3, the details of the authentication procedure are as follows:

(1) The user U_i inserts his/her smart card into a card-reader device and inputs the identity UID_i and the password PW_i.

(2) The smart card verifies the validity of the user U_i by recomputing the secret value $RPW_i' = h_1(PW_i \| UID_i \| r_i)$ and checking if $RPW_i' = RPW_i$. If not, the smart card terminates the authentication procedure; otherwise, it randomly selects an integer $x_i \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} X_i &= x_i \cdot g, \\ X_i' &= X_i + s_{i,j}, \\ M_1 &= h_1\left(r_i \| UID_i \| X_i \| s_{i,j} \| X_i'\right). \end{aligned} \tag{1}$$

After that, the smart card sends the message $\{UID_i, X_i', M_1, r_i\}$ to the service provider S_j.

(3) Upon receiving the message from the user U_i, the service provider S_j verifies UID_i and successively computes

$$\begin{aligned} s_{i,j}' &= h_2\left(UID_i \| r_i\right) + ASID_j, \\ X_i^* &= X_i' - s_{i,j}', \\ M_1' &= h_1\left(r_i \| UID_i \| X_i^* \| s_{i,j}' \| X_i'\right). \end{aligned} \tag{2}$$

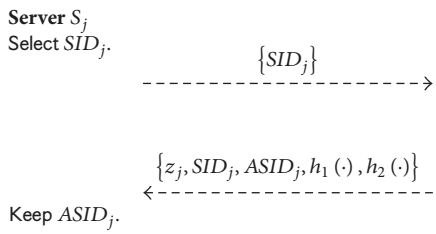

Figure 1: Server registration of Truong et al.'s scheme.

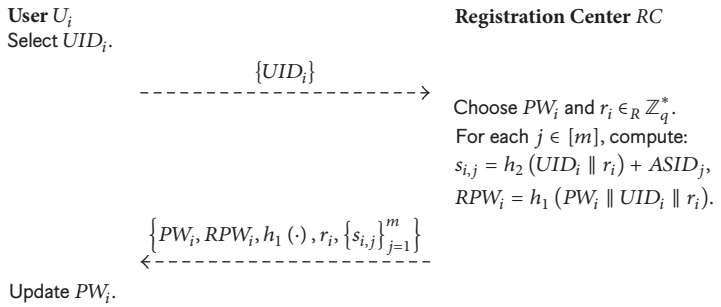

Figure 2: User registration of Truong et al.'s scheme.

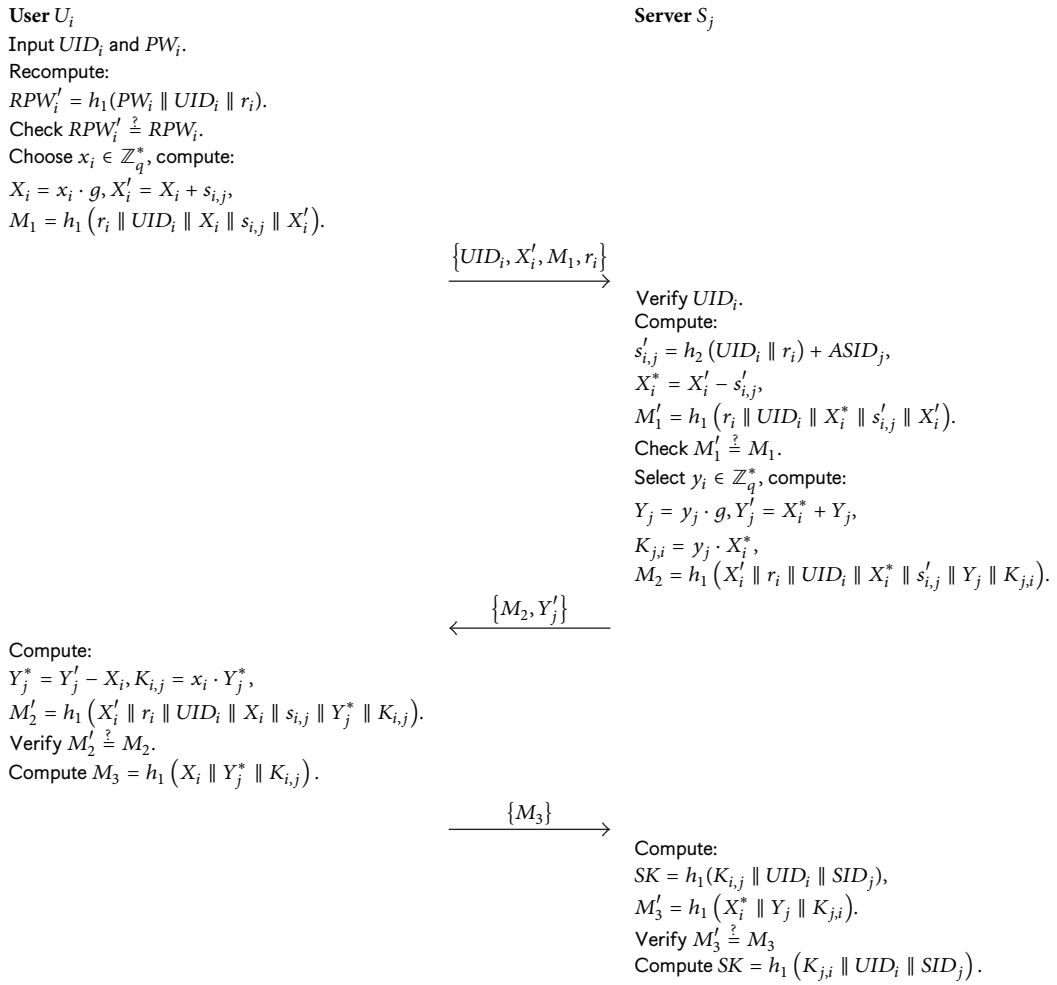

Figure 3: Authentication phase of Truong et al.'s scheme.

Then, it checks whether it holds that $M_1' = M_1$. If not, the service provider S_j also terminates the authentication procedure; otherwise, it chooses a random integer $y_j \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} Y_j &= y_j \cdot g, \\ Y_j' &= X_i^* + Y_j, \\ K_{j,i} &= y_j \cdot X_i^*, \\ M_2 &= h_1\left(X_i' \parallel r_i \parallel UID_i \parallel X_i^* \parallel s_{i,j}' \parallel Y_j \parallel K_{j,i}\right). \end{aligned} \tag{3}$$

Subsequently, S_j sends the message $\{M_2, Y_j'\}$ to the user U_i.

(4) After receiving the message from the service provider S_j, the user U_i's smart card computes

$$\begin{aligned} Y_j^* &= Y_j' - X_i, \\ K_{i,j} &= x_i \cdot Y_j^*, \\ M_2' &= h_1\left(X_i' \parallel r_i \parallel UID_i \parallel X_i \parallel s_{i,j} \parallel Y_j^* \parallel K_{i,j}\right). \end{aligned} \tag{4}$$

Then, the smart card checks if $M_2' = M_2$. If not, the authentication procedure is terminated; otherwise, the smart card successfully authenticates the service provider S_j and sends $M_3 = h_1(X_i \parallel Y_j^* \parallel K_{i,j})$ to S_j.

(5) When receiving the message M_3 from the user U_i, the service provider recomputes $M_3' = h_1(X_i^* \parallel Y_j \parallel K_{j,i})$ and checks if $M_3' = M_3$. If not, S_j terminates the authentication procedure; otherwise, the user U_i is successfully authenticated by the service provider S_j.

(6) The user U_i and the service provider S_j derive a shared session key:

$$\begin{aligned} SK &= h_1\left(K_{i,j} \parallel UID_i \parallel SID_j\right) \\ &= h_1\left(K_{j,i} \parallel UID_i \parallel SID_j\right). \end{aligned} \tag{5}$$

This completes the authentication procedure.

2.4. Password Update Phase. When a user U_i wants to update his/her password, he/she can conveniently achieve this goal by performing the following procedure:

(1) The user U_i inserts the smart card into a card-reader device and provides the corresponding identity UID_i and password PW_i.

(2) The smart card recomputes $RPW_i' = h_1(PW_i \parallel UID_i \parallel r_i)$ and checks if it holds that $RPW_i' = RPW_i$. If not, the update procedure is terminated; otherwise, the smart card requires the user U_i to input a new password PW_i^{new} and computes $RPW_i^{\text{new}} = h_1(PW_i^{\text{new}} \parallel UID_i \parallel r_i)$.

(3) Finally, the smart card replaces RPW_i with RPW_i^{new}. This completes the update procedure.

3. Cryptanalysis of Truong et al.'s Authentication Scheme

In this section, we show that Truong et al.'s [16] protocol suffers from offline password guessing attack and server impersonation attack. To this end, we first formalize the adversary's capacity. Roughly, in the literature of two-factor authentication scheme based on password and smart card, an adversary is allowed to

(i) overhear, modify, synthesize, and intercept any messages transmitted over the public channel,

(ii) obtain the user's password or the private information stored in the smart card by using the technologies introduced in [48, 49], but not both.

The above two assumptions about the adversary's capacity are widely recognised and adopted in the security analysis of password and smart card based two-factor authentication protocols, including [16]'s protocol and our scheme. Below we give the attack details.

3.1. Offline Password Guessing Attack. To launch this kind of attack against a user U_i, an adversary $\mathcal{A}$ records the message $\{UID_i, X_i', M_1, r_i\}$ appearing in some instance of the authentication procedure executed between U_i and a server S_j and then steals U_i's smart card and extracts secret values $\{RPW_i, h_1(\cdot), r_i, \{s_{i,j}\}_{j=1}^m\}$. After that, the adversary $\mathcal{A}$ performs as follows:

(1) Construct a personal password dictionary $\mathbb{D}$, and select a candidate password $PW_i^* \in \mathbb{D}$.

(2) Compute $RPW_i^* = h_1(PW_i^* \parallel UID_i \parallel r_i)$, and check if it holds that $RPW_i^* = PRW_i$. If yes, it implies that $\mathcal{A}$'s guess is correct; otherwise, go to the next step.

(3) Choose a new candidate password, and repeat the previous step until the correct password is recovered.

After the adversary $\mathcal{A}$ gets the correct password, it can further completely impersonate the user U_i since it simultaneously holds U_i's password and smart card. On the other hand, the computation cost of verifying a candidate password is only one hash operation, which is nearly negligible when running it on a personal computer. Thus, the entire complexity of completing the password guessing attack is linear in the size of the password dictionary $\mathbb{D}$, which is rather small in practice. This implies that the adversary $\mathcal{A}$ can recover a user's password in just a few minutes and this kind of attack is practical.

3.2. Impersonation Attack. In this kind of attack, a malicious user (adversary) tries to impersonate a legal user or a server. We demonstrate that a malicious user U_x can impersonate any legal server and user.

First, the malicious user U_x extracts secret values $\{RPW_x, h_1(\cdot), r_x, \{s_{x,j}\}_{j=1}^m\}$ stored in his/her smart card, where $s_{x,j} = h_2(UID_x \parallel r_x) + ASID_j$ for any $j \in [m]$. Intuitively, U_x can directly get $ASID_j = s_{x,j} - h_2(UID_x \parallel r_x)$ (the values $s_{x,j}, UID_x, r_x$ are all available for U_x. In Truong et al.'s

protocol, although the hash function $h_2(\cdot)$ is not stored in the user's smart card, we think it is public and is available for any one. In fact, the user can also get $h_2(\cdot)$ from a malicious server), which is the only secret value of the server S_j. Consequently, U_x can utilize it to impersonate the server S_j at any time.

Second, for any user U_i, since his/her identity UID_i and the random value r_i are transmitted over the public channel, then the malicious user U_x can directly compute the secret value $s_{i,j} = h_2(UID_i \parallel r_i) + ASID_j$ with the corrupted secret value $ASID_j$. Furthermore, by using $s_{i,j}$, the malicious user U_x can impersonate the user U_i to access the service provided by the server S_j, even if he/she does not know U_i's password. Essentially speaking, this is mainly because that the correctness of the password is locally checked by the smart card, rather than by the corresponding server.

In short, by either launching offline password guessing attack or using the above two variants of impersonation attack in a combinational way, a malicious user (adversary) can totally break the security of Turong et al.'s [16] scheme. Thus, it does not achieve the intended security requirements and is not adaptable for practical applications.

4. The Proposed Scheme

In this section, to conquer those security pitfalls in Turong et al.'s [16] protocol, we propose a security enhanced password and smart card based authentication scheme in multiserver environment. Before describing the concrete scheme, we give an overview to demonstrate our design criteria.

Note that the reason of Turong et al.'s [16] protocol suffering from offline password guessing attack mainly lies in the fact that the password correctness is *locally* verified by the smart card. As a result, when the smart card is lost, an adversary can utilize the secret value stored in the smart card to launch offline password guessing attack. In our scheme, we let the service provider check the password validity. Specifically, after a user inputs his/her password, the smart card uses it to recover a secret value, which is computable for the service provider. Then, by verifying the correctness of the secret value, the service provider can ensure whether the user holds the correct password or not. In addition, these messages transmitted over the public channel should also avoid being used to check the validity of the password.

On the other hand, to prevent a malicious user from directly recovering a server's secret key (i.e., $ASID_j$ in Turong et al.'s [16] protocol), we let the registration center first perform hash operation on each server's secret key and then use it to produce a private value for the user, rather than directly employing the server's secret key to do that as in Turong et al.'s [16] protocol. As shown in the security analysis of our scheme, this enables our scheme to be free from user/server impersonation attack. Moreover, as in the design of most authentication schemes, we call Diffie-Hellman key exchange mechanism to achieve key agreement and forward security and exploit the freshness of random numbers and timestamps to prevent replay attack.

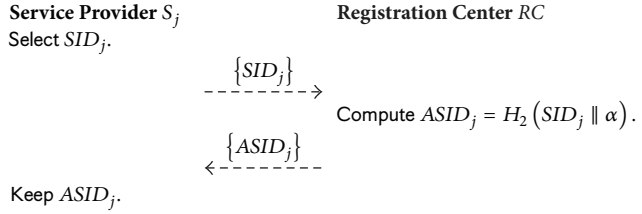

Figure 4: Server registration in our scheme.

The concrete authentication scheme is comprised of four phases: initialization phase, registration phase, authentication phase, and password update phase, which are separately specialized as follows.

4.1. Initialization Phase. In this phase, the registration center initializes the system according to a security parameter κ as follows:

(1) Choose a prime number p with the size κ and then generate an elliptic curve $E_p(a, b)$ defined over $\mathbb{Z}_p^*$, where $a, b \in \mathbb{Z}_p$. Furthermore, produce a cyclic group $\mathbb{G}$ with prime order q from $E_p(a, b)$, and randomly pick a generator $g \in \mathbb{G}$.

(2) Randomly sample an integer α from $\mathbb{Z}_q^*$, and choose two cryptographically secure hash functions $H_1 : \{0,1\}^* \to \{0,1\}^n$, $H_2 : \{0,1\}^* \to \mathbb{G}$.

(3) Publish the system public parameter as $\mathsf{pp} = \{E_p(a, b), g, H_1(\cdot), H_2(\cdot)\}$, which are available to all system users, and set $\mathsf{msk} = \alpha$ as the master secret key.

4.2. Registration Phase. In this phase, each system user U_i registers with the registration center RC to get a smart card containing several secret values, as a credential of U_i to prove his/her authenticity to service providers. Each service provider S_j registers with RC to obtain a secret key, as a credential of S_j to show its legality to system users.

Specifically speaking, to register with RC, as shown in Figure 4, service providers S_j and RC perform as follows:

(1) The service provider S_j selects a unique identity SID_j and sends it to the registration center RC via a secure channel.

(2) Upon receiving the registration request from S_j, the registration center RC directly computes $ASID_j = H_2(SID_j \parallel \alpha)$ and then sends the message $\{ASID_j\}$ to S_j through a secure channel.

(3) After obtaining the registration center RC's response message, S_j keeps $ASID_j$ as its secret key.

For a system user U_i intending to register with the registration center RC, as shown in Figure 5, they interactively conduct the following steps:

(1) The user U_i selects a unique identity UID_i and a personal password PW_i easy to remember. Moreover, he/she randomly samples an integer $r_i \in \mathbb{Z}_q^*$ and

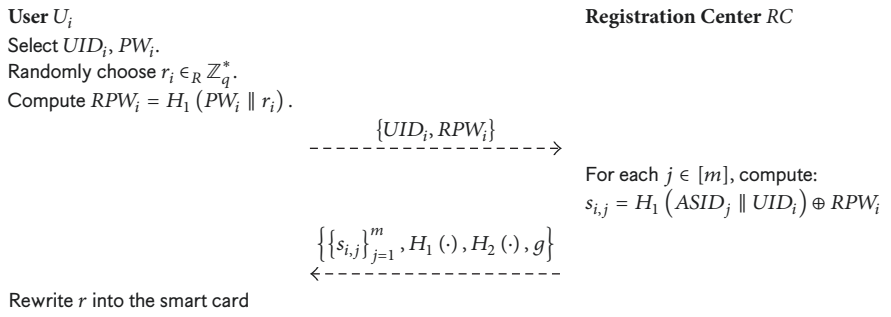

Figure 5: User registration in our scheme.

computes $RPW_i = H_1(PW_i \| r_i)$. Then, U_i sends the registration request message $\{UID_i, PRW_i\}$ to RC via a secure channel.

(2) After receiving the registration request from U_i, the registration center RC computes $s_{i,j} = H_1(ASID_j \| UID_i) \oplus RPW_i$ for each $j \in [m]$ and issues a smart card containing $\{\{s_{i,j}\}_{j=1}^m, H_1(\cdot), H_2(\cdot), g\}$ to the user U_i.

(3) Upon receipt of the smart card, the user U_i rewrites the random value r_i into the smart card and keeps properly.

4.3. Authentication Phase. By running an authentication procedure between a system user U_i and a service provider S_j, they can check the validity of each other and establish a secure channel. That is, S_j ensures that U_i is a registered user, and U_i believes that the service provided by S_j is legal. As shown in Figure 6, such a procedure is performed as follows:

(1) The user U_i attaches his/her smart card to a card-reader device and inputs his/her identity UID_i and the corresponding password PW_i.

(2) U_i's smart card first computes $RPW_i' = H_1(PW_i \| UID_i \| r_i)$ and $s_{i,j}' = s_{i,j} \oplus RPW_i'$. Then, it randomly selects an integer $x_i \in \mathbb{Z}_q^*$ and further calculates

$$\begin{aligned} X_i &= x_i \cdot g, \\ X_i' &= X_i + H_2\left(s_{i,j}'\right), \\ M_1 &= H_1\left(UID_i \| SID_j \| X_i' \| T_i\right), \end{aligned} \tag{6}$$

where T_i is the current timestamp. After that, the smart card sends the authentication request message $\{UID_i, X_i', T_i, M_1\}$ to the service provider S_j.

(3) Upon the receipt of the message from the user U_i, the service provider S_j checks the validity of UID_i and T_i by verifying if $|T_j^c - T_i| \le \Delta T$, where T_j^c is the current timestamp. If not, the user's authentication request would be rejected. Moreover, S_j computes $M_1' = H_1(UID_i \| SID_j \| X_i' \| T_i)$. Then, it checks whether it holds that $M_1' = M_1$. If not, the service provider S_j terminates the authentication procedure; otherwise, it chooses a random integer $y_j \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} s_{i,j}' &= H_1\left(ASID_j \| UID_i\right), \\ X_i^* &= X_i' - H_2\left(s_{i,j}'\right), \\ Y_j &= y_j \cdot g, \\ Y_j' &= X_i^* + Y_j, \\ K_{j,i} &= y_j \cdot X_i^*, \\ M_2 &= H_1\left(UID_i \| SID_j \| X_i' \| Y_j' \| K_{j,i} \| T_j\right), \end{aligned} \tag{7}$$

where T_j is the current timestamp. Subsequently, S_j sends the message $\{M_2, Y_j', T_j\}$ to the user U_i.

(4) After receiving the message from the service provider S_j, the user U_i's smart card first checks the validity of T_j by verifying if $|T_i^c - T_j| \le \Delta T$, where T_i^c is the current timestamp. After that, it computes

$$\begin{aligned} Y_j^* &= Y_j' - X_i, \\ K_{i,j} &= x_i \cdot Y_j^*, \\ M_2' &= H_1\left(UID_i \| SID_j \| X_i' \| Y_j' \| K_{i,j} \| T_j\right). \end{aligned} \tag{8}$$

Then, the smart card checks if $M_2' = M_2$. If not, the authentication procedure is terminated; otherwise, the smart card successfully authenticates the service provider S_j and sends $M_3 = H_1(X_i \| Y_j^* \| K_{i,j} \| T_i')$ to S_j, where T_i' is the current timestamp.

(5) When receiving the message M_3 from the user U_i, the service provider S_j first checks the validity of T_i' by verifying if $|T_j^c - T_i'| \le \Delta T$, where T_j^c is the current timestamp. Then, S_j recomputes $M_3' = H_1(X_i^* \| Y_j \| K_{j,i} \| T_i')$ and checks if $M_3' = M_3$. If not, S_j terminates the authentication procedure; otherwise, the user U_i is successfully authenticated by the service provider S_j.

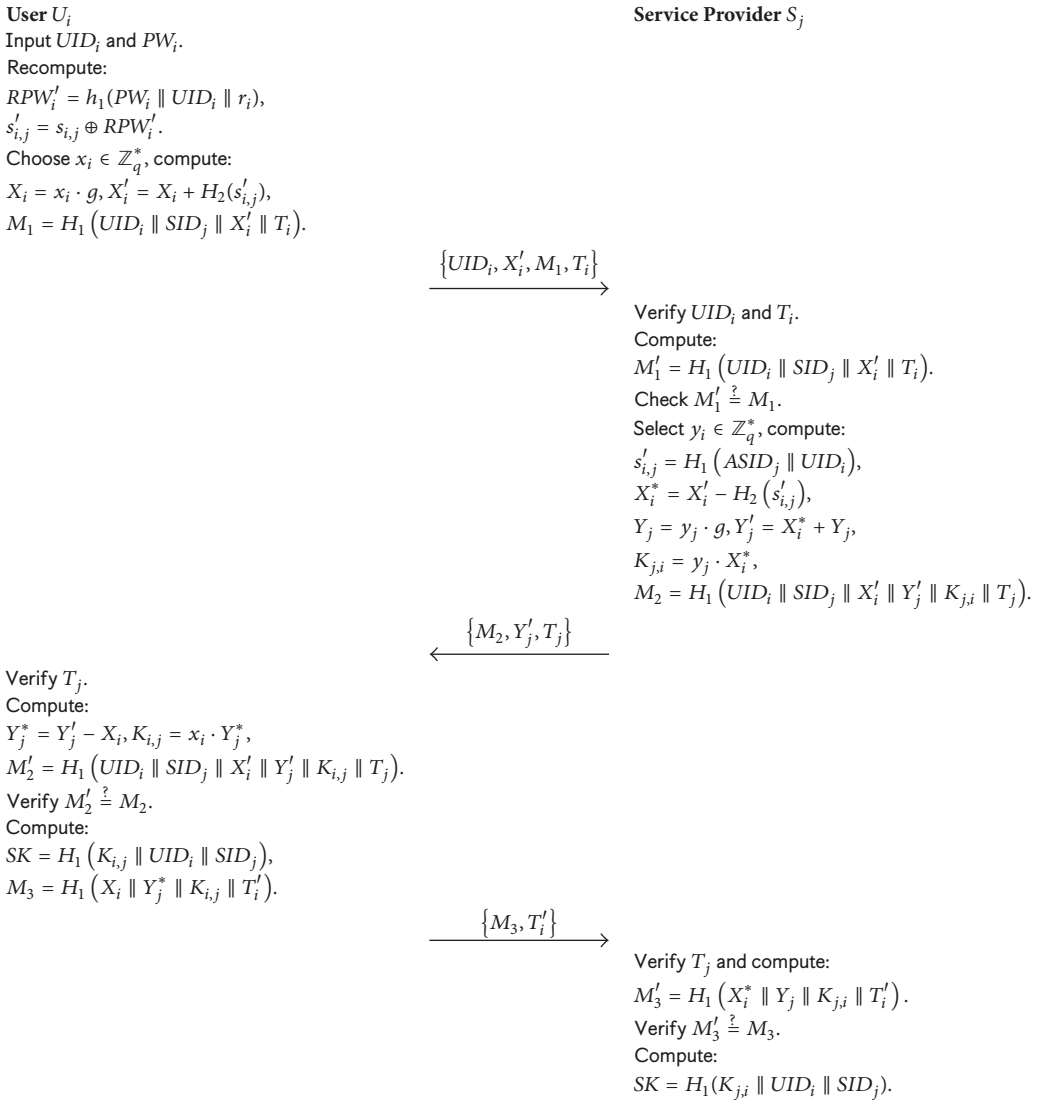

Figure 6: Authentication phase in our scheme.

(6) The user U_i and the service provider S_j derive a shared session key:

$$SK = H_1(K_{i,j} \| UID_i \| SID_j) = H_1(K_{j,i} \| UID_i \| SID_j). \quad (9)$$

This completes the authentication procedure.

4.4. Password Update Phase. When a user U_i wants to update his/her original password PW_i, the following steps are conducted:

(1) U_i randomly selects a service provider S_k, with whom U_i performs the authentication procedure.

(2) If both U_i and S_k pass through the authentication, then U_i selects a new password PW_i^{new} and lets the smart card compute

$$RPW_i^{\text{new}} = H_1(PW_i^{\text{new}} \| r_i), \quad s_{i,j}^{\text{new}} = s_{i,j} \oplus RPW_i \oplus RPW_i^{\text{new}} \quad \text{for } j = 1 \text{ to } m. \quad (10)$$

(3) The smart card replaces $s_{i,j}$ with $s_{i,j}^{\text{new}}$ $(1 \le j \le m)$.

Remark 1. Note that the password update phase in our scheme is significantly different from that in Truong et al.'s [16] scheme, in which the password update is completed in an offline way. This implies that the smart card has to check the validity of the password. As a result, when the smart card is lost, their scheme suffers from offline password guessing attack. However, in our scheme, the validity of the password

is verified by a service provider, rather than the smart card. Although the password update phase in our scheme needs more computation and communication cost (compared with Truong et al.'s [16] scheme), it provides stronger security guarantee.

5. Security Analysis

Although formal security analysis (or provable security) is more desirable, now it is difficult to achieve this requirement due to the complexity of the protocol design. In fact, there are already several similar authentication schemes that are claimed to be provably secure by defining the corresponding security model, or using the BAN logic [50]. However, subsequent works indicate that these schemes all fail to provide the required security properties. Therefore, we use informal and heuristic manner to analyze the security of the proposed scheme. We note that such a manner is widely accepted and used in the literature of multifactor authentication scheme [15, 20, 24, 33, 36, 38, 39, 51].

Specifically, we show that our scheme can resist various well-known attacks, including replay attack, impersonation attack, offline password guessing attack, and known-key attack. We also demonstrate that the proposed scheme achieves intended security goals, such as mutual authentication, key agreement, and two-factor authentication.

5.1. Replay Attack. To launch the replay attack, an adversary first needs to eavesdrop these messages $\{UID_i, X_i', T_i, M_1\}$, $\{M_2, Y_j', T_j\}$, and $\{M_3, T_i'\}$ transmitted between a user U_i and a service provider S_j and then resends one of them to U_i or S_j. Now we show how our scheme can be free from this kind of attack.

If the adversary sends $\{UID_i, X_i', T_i, M_1\}$ (the adversary needs to choose the current timestamp on the user side and recomputes M_1) to the service provider S_j, then S_j would generate a new message $\{M_2^{\text{new}}, Y_j'^{\text{new}}, T_j^{\text{new}}\}$ according to the authentication procedure. Note that the computation of $Y_j'^{\text{new}}$ involves a new random integer. However, the adversary does not know the random integer x_i originally used in the computation of X_i'. As a result, the adversary cannot produce correct response message M_3 and would be rejected by the service provider S_j. Similarly, if the adversary wants to send the message $\{M_2, Y_j', T_j\}$ to the user U_i, it also cannot pass through the verification of the user U_i, since U_i would use a new random integer to check its validity, rather than the original x_i. Thus, the proposed scheme can be free from the replay attack. Essentially speaking, we use two kinds of mechanism in a combinatorial way to avoid the replay attack, namely, timestamp and nonce (in the case that the time period in the system cannot be synchronized, the timestamp would fail to prevent the replay attack).

5.2. Insider Attack. The insider attack mainly means that a malicious insider party (e.g., the registration center or a service provider) tries to get a user's password. First, from the perspective of the registration center, it can get the registration information $RPW_i = H_1(PW_i \parallel r_i)$, where r_i is a random integer chosen by the user U_i. Thus, without the knowledge of r_i, the registration center cannot launch offline password guessing attack to recover U_i's password PW_i. Second, on the side of a service provider S_j, it also cannot get any information about the user U_i's password, since those messages transmitted between them do not involve U_i's password. Therefore, we conclude that our scheme can resist the insider attack.

5.3. Impersonation Attack. The impersonation attack against the proposed scheme falls into two classes, i.e., user personation and service provider personation. Below we show that the proposed scheme can withstand these two kinds of impersonation attack.

If an adversary $\mathcal{A}$, which might be a malicious user or a malicious service provider, wants to impersonate user U_i, then it has to correctly compute the challenge value $M_3 = H_1(X_i \parallel Y_j^* \parallel K_{i,j} \parallel T_i')$. Obviously, for this to be computable, the adversary $\mathcal{A}$ has to know the values X_i, Y_j^*, and $K_{i,j}$. Moreover, if the adversary itself produces the value $X_i = x_i \cdot g$ without (if the adversary holds U_i's password and smart card, then it will not have to impersonate U_i) using U_i's secret information (i.e., PW_i and $s_{i,j}$), then the corresponding value X_i^* computed by the service provider is not equal to X_i. As a result, it holds that $K_{i,j} \neq K_{j,i}$, which implies that the adversary $\mathcal{A}$ cannot pass through the service provider's authentication since $M_3 \neq M_3'$. Thus, the adversary $\mathcal{A}$ cannot impersonate the user U_i.

On the other hand, if the adversary $\mathcal{A}$ wants to impersonate a service provider S_j, then it also has to pass through a user U_i's authentication. This requires the adversary $\mathcal{A}$ to compute the correct value $K_{j,i} = y_j \cdot X_i^*$. We note that the adversary $\mathcal{A}$ cannot get correct $X_i^* = X_i' - s_{i,j}'$ without the secret value $ASID_j$ ($s_{i,j}' = H_1(ASID_j \parallel UID_i)$). Consequently, it cannot impersonate the service provider S_j since it cannot produce the correct value M_2.

5.4. Offline Password Guessing Attack. To launch the offline password guessing attack, an adversary $\mathcal{A}$ has to hold a value that can be used to check the validity of a candidate password. Below we demonstrate that none of those transmitted messages in the proposed scheme can be used to do this.

In the authentication phase of the proposed scheme, note that a user U_i's password PW_i is only used to recover the secret value $s_{i,j}' = s_{i,j} \oplus RPW_i$, and its validity is not verified on the user side. Thus, even if the smart card is lost, those secret values stored in the smart card cannot be used to launch offline password guessing attack. Furthermore, for the first message $\{UID_i, X_i, T_i, M_1\}$, both the value $X_i' = X_i + H_2(s_{i,j}')$ and the hash value $M_1 = H_1(UID_i \parallel SID_j \parallel X_i' \parallel T_i)$ involve U_i's password. However, the computation of M_1 does not need any secret value and thus cannot be used to check the validity of a candidate password. Moreover, without the knowledge of the value X_i, the public value X_i' also cannot be used to verify the validity of a candidate password. For the second message $\{M_2, Y_j'\}$ and third message $\{M_3\}$, observe that they do not involve any information about U_i's password

Table 2: Comparisons of security properties with related works.

Security Properties	Li et al. [47]	Pippal et al. [40]	Yeh [15]	Truong et al. [16]	Ours
Insider attack	✘	✔	✔	✔	✔
Replay attack	✔	✔	✔	✔	✔
Known-key attack	✔	✔	✔	✔	✔
Mutual authentication	✔	✔	✘	✔	✔
Session key agreement	✔	✔	✘	✔	✔
Perfect forward secrecy	✘	✔	✔	✔	✔
Two-factor authentication	✘	✘	✘	✘	✔
User impersonation attack	✘	✘	✘	✘	✔
Server impersonation attack	✘	✘	✔	✘	✔
Offline password guessing attack	✘	✘	✘	✘	✔

Note. ✔ means that the scheme can provide the corresponding security property, and ✘ indicates that it cannot achieve this.

and thus naturally cannot be used to launch the offline password guessing attack. Therefore, the proposed scheme is secure against password guessing attack.

5.5. Known-Key Attack. The known-key attack means that the disclosure of a session key will affect the security of other session keys. In our scheme, a session key $SK = H_1(K_{i,j} \parallel UID \parallel SID_j)$ is derived from a fresh value $K_{i,j} = (x_i y_j) \cdot g$, where x_i and y_j are randomly sampled from $\mathbb{Z}_q^*$. This implies that all session keys are independent from each other. Thus, the disclosure of a session key has no influence on the security of other session keys, and the proposed scheme can withstand known-key attack.

5.6. Mutual Authentication and Key Agreement. In the authentication of the proposed scheme, observe that both the user U_i and the service provider S_j have to respond to the partner's challenge. Specifically, the service provider S_j uses its secret value $ASID_j$ to produce a correct response value M_2, whose validity would be checked by the user U_i. On the other hand, the user U_i also utilizes the consistent value X_i to generate a response value M_3, whose correctness would be verified by the service provider S_j. We can see that, to complete the authentication procedure, the user U_i and the service provider S_j are required to pass through each other's authentication. Thus, the proposed scheme achieves mutual authentication.

Focusing on the generation of the session key, we can see that both the user U_i and the service provider contribute to the computation of session key, namely, x_i and y_j. This implies that neither one can completely control the generation of the session key, and the session key can be sufficiently random if at least one of the participants is able to produce sufficiently random inputs. Therefore, the proposed scheme enjoys the functionality of key agreement.

5.7. Perfect Forward Secrecy. In the setting of the proposed scheme, the perfect forward secrecy requires that after a user U_i's secret information (i.e., the password and those secret values stored in the smart card) and a service provider S_j's secret key $ASID_j$ get exposed, previous session keys established and used between U_i and S_j should remain secure. By basing our scheme on the Diffie-Hellman protocol, it achieves perfect forward secrecy. Concretely, with the knowledge of U_i and S_j's secret information, an adversary can recover $X_i = x_i \cdot g$ and $Y_j = y_j \cdot g$. However, from the intractability of the computational Diffie-Hellman problem, we know that it is impossible for $\mathscr{A}$ to compute $K_{i,j} = (x_i y_j) \cdot g$, and thus $\mathscr{A}$ cannot further recover the session key $SK = H_1(K_{i,j} \parallel UID_i \parallel SID_j)$. This is why the perfect forward secrecy of the proposed scheme can be achieved.

5.8. Two-Factor Authentication. Two-factor authentication is a major security advantage of password and smart card based authentication scheme. That is, once the password or the smart card is revealed (not the both), the scheme should remain secure. Below we show that the proposed scheme is still secure in the above two cases.

In the case that an adversary $\mathscr{A}$ knows a user U_i's password PW_i but does not have U_i's smart card, the adversary $\mathscr{A}$ cannot compute $s'_{i,j} = s_{i,j} \oplus RPW_i$, where $s_{i,j}$ is stored in the smart card and $RPW_i = H_1(PW_i \parallel UID_i \parallel r_i)$. As a result, it cannot produce a correct value X_i that is consistent with the one computed by the service provider. Thus, the adversary cannot pass through the service provider's authentication. Namely, the proposed scheme remains secure in this case.

In the case that an adversary $\mathscr{A}$ holds a user's smart card but does not know the user's password, as analyzed in the offline password guessing attack, no message can be used check the validity of a candidate password. On the other hand, without the knowledge of the correct password, due to the similar reason, the adversary $\mathscr{A}$ also cannot pass through the service provider's authentication. Thus, the proposed scheme is still secure in this case.

6. Performance Discussions

In this section, we discuss the performance of the proposed scheme in terms of security property and computation efficiency via comparing it with other related works.

In Table 2, we present security properties of the listed authentication schemes. We can see that only Li et al.'s [47] scheme cannot resist insider attack since the registration center maintains a table to verify users' passwords. As

Table 3: Comparisons of computation efficiency with related works.

Computation party	Li et al. [47]	Pippal et al. [40]	Yeh [15]	Truong et al. [16]	Ours
User side	$10T_h + 6T_{\text{xor}}$	$4T_h + 3T_e$	$4T_h + 2T_e$	$5T_h + 2T_p + 2T_m$	$6T_h + 2T_p + 2T_m$
Server side	$18T_h + 15T_{\text{xor}}$	$3T_h + 4T_e$	$4T_h + 4T_e$	$6T_h + 2T_p + 3T_m$	$6T_h + 2T_p + 2T_m$
Total	$28T_h + 21T_{\text{xor}}$	$7T_h + 7T_e$	$8T_h + 6T_e$	$11T_h + 4T_p + 5T_m$	$12T_h + 4T_p + 4T_m$

Note. T_h = the running time of one hash operation. T_{xor} = the running time of one XOR operation. T_p = the running time of one point multiplication over elliptic curves group. T_m = the running time of one point addition over elliptic curves group. T_e = the running time of one modular exponentiation.

analyzed by Truong et al. [16], in Yeh's [15] scheme, the secret values computed by the user and the service provider are not consistent with each other; thus, this scheme cannot provide the security properties of mutual authentication and session key agreement. In addition, Li et al.'s [47] scheme only uses hash and XOR operations. Thus, after an adversary gets a user/server's secret keys, it can further recover any previous session keys. Consequently, it cannot provide perfect forward secrecy. Moreover, except for our scheme, other listed schemes [15, 16, 40, 47] all cannot resist offline password guessing attack, which is the most realistic threat aimed at password and smart card based authentication scheme. Even worse, after the user's smart card is lost, an adversary can correctly recover the corresponding password. This is also why these schemes [15, 16, 40, 47] suffer from user impersonation attack and cannot achieve two-factor authentication. Particularly, in Truong et al.'s [16] scheme, a malicious user can directly obtain any service provider's secret key. As a result, their scheme is vulnerable to server impersonation attack. In short, our scheme surpasses other listed schemes in terms of security property and provides stronger security guarantee for practical applications.

In Table 3, we summarize the computation cost of the authentication procedure of these schemes. By running the main operations (i.e., hash operation, XOR operation, modular exponentiation, point multiplication, and point addition) involved in these schemes by calling the library MIRCAL (https://libraries.docs.miracl.com/), we have that $T_e \gg T_p > T_m > T_h \gg T_{\text{xor}}$. Thus, we can see that Li et al.'s [47] scheme is the most efficient one since it only uses the hash operation and XOR operation, but it cannot provide the required security guarantee. On the other hand, our scheme is more efficient than other schemes [15, 16, 40]; meanwhile, it is also more secure than them. Therefore, from the perspective of both security and efficiency, the proposed authentication is more desirable for practical applications.

7. Conclusion

In this paper, we study the design and analysis of password and smart card based authentication scheme for multiserver architecture. Specifically, we pointed out that Truong et al.'s [16] scheme is vulnerable to offline password guessing attack and user/server impersonation attack. We also analyzed why their scheme fails to achieve the intended security goal. Moreover, we proposed a security enhanced and cost-effective authentication scheme to secure communications in the setting of multiple service providers. Security analysis and performance discussion show that our scheme has advantages in terms of security and efficiency and thus is more desirable for practical applications.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards viable certificate-based authentication for the Internet of Things," in *Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy (HotWiSec '13)*, pp. 37–42, ACM, Budapest, Hungary, April 2013.

[2] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

[3] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5931, pp. 157–166, 2009.

[4] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.

[5] D. He, S. Zeadally, B. Xu, and X. Huang, "An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[6] M. S. Farash and M. A. Attari, "An efficient client–client password-based authentication scheme with provable security," *The Journal of Supercomputing*, vol. 70, no. 2, pp. 1002–1022, 2014.

[7] C.-C. Lee, C.-T. Li, and C.-W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 125–132, 2013.

[8] D. He, D. Wang, Q. Xie, and K. Chen, "Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation," *Science China Information Sciences*, vol. 60, no. 5, Article ID 052104, 2017.

[9] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals

Are Beyond Attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[10] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 71–77, 2015.

[11] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[12] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56–75, 2016.

[13] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-Health clouds," *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849, 2016.

[14] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.

[15] K.-H. Yeh, "A Provably Secure Multi-server Based Authentication Scheme," *Wireless Personal Communications*, vol. 79, no. 3, pp. 1621–1634, 2014.

[16] T.-T. Truong, M.-T. Tran, A.-D. Duong, and I. Echizen, "Provable Identity Based User Authentication Scheme on ECC in Multi-server Environment," *Wireless Personal Communications*, vol. 95, no. 3, pp. 2785–2801, 2017.

[17] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[18] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proceedings Part E Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[19] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[20] Y.-Y. Wang, J.-Y. Liu, F.-X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[21] K.-H. Yeh, C. Su, N. W. Lo, Y. Li, and Y.-X. Hung, "Two robust remote user authentication protocols using smart cards," *The Journal of Systems and Software*, vol. 83, no. 12, pp. 2556–2565, 2010.

[22] M. K. Khan, S.-K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[23] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2015.

[24] F. Wu, L. Xu, S. Kumari et al., "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *Journal of Network and Computer Applications*, vol. 89, pp. 72–85, 2017.

[25] S. Kumari, S. A. Chaudhry, F. Wu, X. Li, M. S. Farash, and M. K. Khan, "An improved smart card based authentication scheme for session initiation protocol," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 92–105, 2017.

[26] S. Kumari, X. Li, F. Wu, A. K. Das, V. Odelu, and M. K. Khan, "A user anonymous mutual authentication protocol," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 9, pp. 4508–4528, 2016.

[27] J. Wei, X. Hu, and W. Liu, "An improved authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3597–3604, 2012.

[28] I.-E. Liao, C.-C. Lee, and M.-S. Hwang, "Security enhancement for a dynamic ID-based remote user authentication scheme," in *Proceedings of the International Conference on Next Generation Web Services Practices, NWeSP 2005*, pp. 437–440, kor, August 2005.

[29] L. Li, I. Lin, and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 12, no. 6, pp. 1498–1504, 2001.

[30] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[31] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.

[32] W.-C. Ku, H.-M. Chuang, and M.-H. Chiang, "Cryptanalysis of a multi-server password authenticated key agreement scheme using smart cards," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 11, pp. 3235–3238, 2005.

[33] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[34] H.-C. Hsiang and W.-K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[35] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[36] J. Wei, W. Liu, and X. Hu, "Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 77, no. 3, pp. 2255–2269, 2014.

[37] X. Li, J. Niu, S. Kumari, J. Liao, and W. Liang, "An Enhancement of a Smart Card Authentication Scheme for Multi-server Architecture," *Wireless Personal Communications*, vol. 80, no. 1, pp. 175–192, 2015.

[38] V. Odelu, A. K. Das, and A. Goswami, "An Effective and Robust Secure Remote User Authenticated Key Agreement Scheme Using Smart Cards in Wireless Communication Systems," *Wireless Personal Communications*, vol. 84, no. 4, pp. 2571–2598, 2015.

[39] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[40] R. S. Pippal, C. D. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[41] V. Odelu, A. K. Das, and A. Goswami, "A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[42] D. He and D. Wang, "Robust Biometrics-Based Authentication Scheme for Multiserver Environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[43] S. Kumari, X. Li, F. Wu, A. K. Das, K.-K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[44] A. Irshad, S. A. Chaudhry, Q. Xie et al., "An Enhanced and Provably Secure Chaotic Map-Based Authenticated Key Agreement in Multi-Server Architecture," *Arabian Journal for Science and Engineering*, vol. 43, no. 2, pp. 811–828, 2018.

[45] S. Kumari, A. K. Das, X. Li et al., "A provably secure biometrics-based authenticated key agreement scheme for multi-server environments," *Multimedia Tools and Applications*, pp. 1–31, 2018.

[46] R. Amin, S. K. H. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, "A two-factor RSA-based robust authentication system for multiserver environments," *Security and Communication Networks*, vol. 2017, Article ID 5989151, 15 pages, 2017.

[47] X. Li, Y.-P. Xiong, J. Ma, and W.-D. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[48] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the Advances in Cryptology–CRYPTO 1999*, pp. 789-789, Springer, 1999.

[49] T. S. Messerges, E. A. Dabbish, and R. . Sloan, "Examining smart-card security under the threat of power analysis attacks," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[50] M. Burrows, M. Abadi, and R. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[51] J. Wei, W. Liu, and X. Hu, "Secure control protocol for universal serial bus mass storage devices," *IET Computers & Digital Techniques*, vol. 9, no. 6, pp. 321–327, 2015.