Efficient and Anonymous Mobile User Authentication Protocol Using Self-certified Public Key Cryptography for Multi-server Architectures

Debiao He, Sherali Zeadally, Neeraj Kumar, and Wei Wu

Abstract—Rapid advances in wireless communication technologies have paved the way for a wide range of mobile devices to become increasingly ubiquitous and popular. Mobile devices enable anytime, anywhere access to the Internet. The fast growth of many types of mobile services used by various users has made the traditional single-server architecture inefficient in terms of its functional requirements. To ensure the availability of various mobile services, there is a need to deploy multi-server architectures. To ensure the security of various mobile services applications, the Anonymous Mobile User Authentication (AMUA) protocol without online registration using the Self-Certified Public Key Cryptography (SCPKC) for multi-server architectures was proposed in the past. However, most of past AMUA solutions suffer from malicious attacks or have unacceptable computation and communication costs. To address these drawbacks, we propose a new AMUA protocol that uses the SCPKC for multi-server architectures. In contrast to existing AMUA protocols, our proposed AMUA protocol incurs lower computation and communication costs. By comparing with two of the latest AMUA protocols, the computation and the communication costs of our protocol are at least 74.93% and 37.43% lower than them respectively. Moreover, the security analysis of our AMUA protocol demonstrates that it satisfies the security requirements in practical applications and is provably secure in the novel security model. By maintaining security at various levels, our AMUA protocol is more practical for various mobile applications.

Index Terms—Authentication, bilinear pairing, mobile, multi-server architecture, security.

1 Introduction

The significant improvements in software, hardware, and wireless communication technologies have led to the emergence of a wide range of mobile devices such as PDAs, smart phones, and notebooks. These devices have become an integral part of our daily life today. According to the recent survey [1], the number of Americans owning a smart phone has increased from 35% at the end of 2011 to about 64% by the end of 2014. Wireless communication technologies along with powerful mobile devices have led to the emergence and proliferation of many different types of mobile services such as mobile banking, mobile online shopping, mobile online game, and mobile pay-TV which can be accessed from anywhere at anytime. This technological revolution in mobile computing and devices brings a lot of convenience to end-users.

The traditional single-server architecture for the mobile service system consists of a server and many mobile users. By using wireless communication technologies, the mobile user can remotely access the mobile services provided by the server. But, the computation, communication and storage capabilities of the server are limited. With an increase in the number of users and the emergence of different types of mobile devices, the traditional architecture with only one server may become a performance bottleneck for various mobile services [2]. To address this drawback, a multi-server architecture was proposed for mobile service systems. In a multi-server architecture, many servers provide various types of mobile services so that users can access these services from anywhere over different types of wireless networks. A typical multi-server architecture for the mobile service system is shown in Fig. 1.

Due to the openness of wireless networks, the adversary can easily control the communication channel and carries out many kinds of attacks. For example, the adversary can intercept, modify, replay and delay messages transmitted in the systems [3]. To prevent the adversary from accessing the mobile service, we need an efficient security protocol to

- *The work of D. He was supported in part by the National Natural Science Foundation of China under Grant 61572379, Grant 61501333, and Grant U1536204, in part by the National High-Tech Research and Development Program of China (863 Program) under Grant 2015AA016004, in part by the open fund of State Key Laboratory of Cryptology and in part by the Natural Science Foundation of Hubei Province of China under Grant 2015CFB257. The work of W. Wu is supported by National Natural Science Foundation of China under Grant 61472083, and Grant 61402110.*

- *D. He is with the State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China and the State Key Laboratory of Cryptology, Beijing, China*
  E-mail: hedebiao@163.com
- *S. Zeadally is with the College of Communication and Information at the University of Kentucky, USA.*
  E-mail: szeadally@uky.edu
- *N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala, India*
  E-mail: nehra04@yahoo.co.in
- *W. Wu (Corresponding author) is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, Fujian, China*
  E-mail: weiwu81@gmail.com

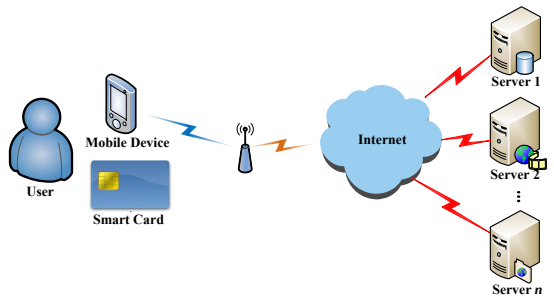

Fig. 1. A typical multi-server architecture

provide secure communications in wireless networks.

The Anonymous Mobile User Authentication (AMUA) protocol is a significant security protocol which can provide confirmation for the other party's identity and preserve the users' privacy. When the AMUA protocol is executed, it generates a session key for encryption to preserve the integrity of future messages transmitted in the system. Since Lamport's work about the user authentication protocol [4], many AMUA protocols for single-server architectures have been introduced for different types of environments [5-10]. However, they are not applied in multi-server architectures because the user needs to register and store private keys generated by these servers individually. To reduce the user's burden and guarantee secure communications, it is urgent to construct AMUA protocols for multi-server architecture.

According to recent surveys [11-20], some AMUA protocols for multi-server architectures were presented in the last decade. Based on the cryptographic algorithm used, these protocols are divided into the private key cryptography-based AMUA protocols [11-25] and the public key cryptography-based AMUA protocols [26-30]. However, it has been found that the private key cryptography-based AMUA protocols have better performance. However, they cannot provide important security attributes such as perfect forward secrecy and two-factor security, which imply the protocol is still secure when one and only one of two facts (password and smart card) is lost. Hence, public key cryptography-based AMUA protocols for multi-server architectures have become more popular. Many public key cryptography-based AMUA protocols [26-33] have been proposed for different applications. However, they need the on-line registration center's help to achieve mutual authentication.

Recently, several AMUA protocols [34-37], using the Self-Certified Public Key Cryptography (SCPKC), were proposed. Compared with previous public key cryptography-based AMUA protocols [26-33], they need no on-line registration center to achieve mutual authentication and have lower communication cost. Due to this advantage such AMUA protocols have become more popular among researchers and designers as they can be applied to a wide range of applications. We summarize below some of the major drawbacks and limitations of recently proposed AMUA protocols that use the SCPKC:

- **The computation cost associated with these protocols is not acceptable for most practical applications:**

It is well known that mobile devices are resource-constrained in terms of storage and computation capabilities. However, the mobile device in these protocols has to execute the bilinear paring operation and the map-to-point hash operation which are two of the most complex operations in modern public key cryptography.

- **The security analyses of these protocols are weak:**

Most of the authors of previously proposed protocols presented only a preliminary analysis of the protocols but they did not present any evaluation with any provable security model. Hence, these protocols can have various vulnerability issues leading to a number of serious attacks.

- **Important functions or security attributes are not supported by these protocols:**

Several important functions such as key establishment and user anonymity, and security attributes such as two-factor security and no verifier table are not considered in the design of these protocols. These limited functionalities prevent their deployment in various real-time applications.

It remains a significant challenge to construct an AMUA protocol with better efficiency and security for multi-server architectures to protect the authorized users' rights for various practical mobile applications.

1.1 Organization of the paper

The rest of the paper is organized as follows. In Section II, we give a brief review previous work AMUA protocols for multi-server architectures. In Section III, we present the notations used in this paper. In Section IV, we present details of the proposed AMUA protocol. In Sections V and VI, we perform the analysis of the security and the performance of the proposed AMUA protocol to demonstrate its advantages compared to other AMUA protocols proposed in the past. Finally, we conclude the paper in Section VII.

2 Related Work and Our Research Contributions

To address the problem that traditional authentication protocols for single-server architectures cannot be directly applied in multi-server architectures, Li *et al.* [11] introduced the concept of authentication protocol for multi-server architectures and designed the first such protocol using the neural network. Due to the complexity of the neural network, Li *et al.'s* performance is not practical at all. To improve performance, Juang [12] designed a new protocol using symmetric cryptography. Unfortunately, Jung's protocol cannot resist the insider attack. To enhance security, several new protocols [13-15] using the symmetric cryptography were proposed after Juang's work.The afore mentioned protocols cannot provide user anonymity because they transmit the user's identity without any protection. To provide user anonymity, Liao and Wang [16] presented a dynamic identity-based protocol using symmetric cryptography. However, their protocol is vulnerable to three kinds of attacks [17]. To address security problems, Hsiang and Shih [17] proposed an improved protocol. Unfortunately, Lee *et al.* [18] pointed out that Hsiang and Shih's protocol cannot withstand the server

spoofing attack. Subsequently, several dynamic identity-based protocols [19-25] (using symmetric cryptography) were presented for different practical environments. Due to the performance efficiency of symmetric cryptography, these protocols yield much better performance. However, these protocols cannot provide important security attributes such as-perfect forward secrecy and the two-factor security.

With increasing security requirements from users, many of the previously proposed protocols have become unsuitable for many practical applications today. To enhance security, public key cryptography (PKC) is widely employed in the design of the authentication protocols for multi-server architectures. Lin *et al.* [26] designed an authentication protocol for multi-server architectures using the PKC. However, Lin *et al.'s* protocol is vulnerable to the impersonation attack [27]. To address the security problem, Yoon and Yoo [28] proposed an improved protocol using the Elliptic Curve Cryptography (ECC). However, Kim *et al.* [29] showed that Yoon and Yoo is vulnerable to the off-line password guessing attack. To enhance security, He and Wang [30] proposed an improved protocol. However, Odelu *et al.* [31] found that their protocol cannot resist three kinds of attacks. Odelu *et al.* [31] also constructed an improved protocol to solve the security problems in He and Wang's protocol. At the same time, Zhang *et al.*[32] and Tseng *et al.* [33] also proposed two protocols for multi-server architectures using ECC. The above protocols [26, 28-33] have many security advantages than previous protocols [11-25]. However, they require that the registration center should always be online which increases communication costs and complexity.

To address the above problems, Choi *et al.* [34] constructed an AMUA protocol for multi-server architectures by using the SCPKC, and in this approach no online registration center is needed. Later, Chuang and Tseng [35] designed an improved AMUA protocol to improve the performance of Choi's protocol. However, neither Choi *et al.'s* protocol [34] nor Chuang *et al.'s* AMUA protocol [35] can provide user anonymity and two factor security. To enhance security, Liao and Hsiao [36] proposed an AMUA protocol using the SCPKC for multi-server architectures. Unfortunately, Hsieh and Leu [37] found that Liao and Hsiao's AMUA protocol cannot withstand the trace attack and suffer from the problem of updating identity-table frequently. To solve the above problems, Hsieh and Leu [37] also constructed an improved AMUA protocol. However, Amin and Biswas [38] pointed out that Hsieh and Leu's AMUA protocol suffers from the password guessing attack and the server spoofing attack. Amin and Biswas [38] also found that Hsieh and Leu's AMUA protocol cannot provide user anonymity. Although Amin and Biswas designed a new AMUA protocol to address the security problem in Hsieh and Leu's protocol, with their approach the online registration center is required to achieve mutual authentication.

2.1 Our research contributions

In this paper, we propose an AMUA protocol which yields better efficiency and security for multi-server architectures using the SCPKC. Our major contributions are summarized as follows:

- First, we present a new AMUA protocol for multi-server architectures using the SCPKC. To improve performance at the user end, neither bilinear pairing operation nor map-to-point hash operation is involved in the proposed AMUA protocol.
- Second, we perform an in-depth security analysis to show the proposed AMUA protocol is provably secure and can satisfy the security requirements of multi-server architectures.
- Finally, we analyze the performance of the proposed AMUA protocol to show that it incurs lower computation and communication costs than previously proposed AMUA protocols.

3 Background and Notations

3.1 Bilinear pairing

The bilinear pairing has been widely used in modern PKC. To enhance the readability of this paper, some background about the bilinear pairing is presented below.

Let q be a large prime number. Let G_1, G_2 and e : $G_1 \times G_1 \rightarrow G_2$ denote an additive group, a multiplicative group and a map respectively. Suppose G_1 and G_2 are generated by generators P and g separately, we say a map e is a bilinear pairing if it has the below attributes:

- **Bilinearity**: Given two elements $S, T \in G_1$ and two elements $a, b \in Z_q^*$, the equation $e(a \cdot S, b \cdot T) = e(S, T)^{a \cdot b}$ holds.
- **Nondegeneracy**: The inequation $e(P, P) \neq 1_{G_2}$ holds for at leat one element $P \in G_1$.
- **Computability**: Given any two elements $S, T \in G_1$, at least one efficient algorithm exists to compute $e(S, T)$.

It is well known that there is no polynomial algorithm to get the answer of the below mathematical problems, which is the security basis of the proposed AMUA protocol.

- **Discrete Logarithm (DL) Problem**: Given an element $X \in G_1$ ($x \in G_2$), the goal of the DL problem is to calculate $\tau \in Z_q^*$ such that $X = \tau \cdot P$ ($x = g^{\tau}$).
- **Computational Diffie-Hellman (CDH) Problem**: Given two elements $a \cdot P, b \cdot P \in G_1$ ($g^a, g^b \in G_2$), the goal of the CDH problem is to calculate $(a \cdot b) \cdot P \in G_1$ ($g^{a \cdot b} \in G_2$), where a and b are two unknown elements in Z_q^*.
- **Collusion Attack Algorithm with k traitor (k-CAA) Problem**[39]: Given k elements $\alpha_1, \alpha_2, \cdots, \alpha_k \in Z_q^*$ and $k+1$ elements $\hat{\tau} \cdot P, \frac{1}{\hat{\tau}+\alpha_1} \cdot P, \frac{1}{\hat{\tau}+\alpha_2} \cdot P, \cdots, \frac{1}{\hat{\tau}+\alpha_k} \cdot P \in G_1$, the goal of the k-CAA problem is to calculate $\frac{1}{\hat{\tau}+\alpha} \cdot P$ for any $\alpha \notin \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$, where $\hat{\tau}$ is a unknown element in $\in Z_q^*$.
- **Modified Bilinear Inverse Diffie-Hellman with k value (k-mBIDH) Problem**[40]: Given k elements $\alpha_1, \alpha_2, \cdots, \alpha_k \in Z_q^*$ and $k+2$ elements $\hat{\tau} \cdot P, \eta \cdot P, \frac{1}{\hat{\tau}+\alpha_1} \cdot P, \frac{1}{\hat{\tau}+\alpha_2} \cdot P, \cdots, \frac{1}{\hat{\tau}+\alpha_k} \cdot P \in G_1$, the goal of the k-mBIDH problem is to calculate $e(P, P)^{\frac{\eta}{\hat{\tau}+\alpha}}$ for any $\alpha \notin \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$, where $\hat{\tau}$ and η are two unknown elements in $\in Z_q^*$.

3.2 Network model

According to recent proposals [34-37], the network model of the AMUA protocol for multi-server architectures is shown in Fig. 2. There are three types of participants with an AMUA protocol: a mobile user U_i, a server S_j, and the registration center RC.

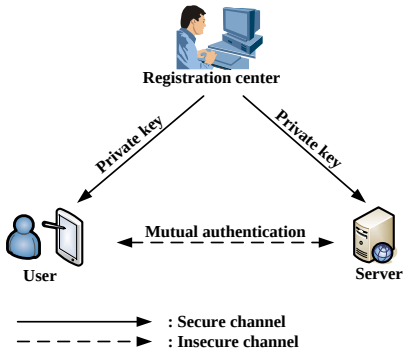

Fig. 2. Network model

- RC: It is a trusted third party and its task is generating system parameters. Besides, it is also generate private keys of U_i and S_j according to their identities.
- U_i: He/she is a mobile user who gets his/her private key from RC and uses it to prove its identity to S_j. After S_j's verification, he/she can access mobile services provided by S_j.
- S_j: It is a mobile service provider which also gets its private key from RC and uses it to show the eligibility of its identity. After verifying U_i's validity, S_j provides the corresponding mobile service according to U_i's request.

3.3 Security requirements

According to recent proposals on AMUA protocols for multi-server architectures, a protocol should satisfy the following functions and security requirements [33-37].

Single registration: To provide convenience to the user, the AMUA protocol for multi-server architectures should provide single registration, i.e., the user just needs to register withthe registration center before accessing services provided by all servers in the system.

Mutual authentication: To ensure the eligibility of participants, the AMUA protocol should provide mutual authentication.

User anonymity: To preserve users' privacy, the AMUA protocol should provide the user anonymity, i.e., the adversary is unable to extract the user's real identity from the intercepted messages.

Un-traceability: To provide better protection for the user's privacy, the AMUA protocol for multi-server architectures should support un-traceability, i.e., the adversary is unable to trace the user's behavior from the intercepted messages.

Session key agreement: To ensure the security of messages transmitted in future communications, the AMUA protocol for multi-server architectures should be able to establish a shared session key between two participants, which will be used to encrypt messages.

Perfect forward secrecy: To ensure the security of messages transmitted in previous communications, the AMUA protocol for multi-server architectures should provide perfect forward secrecy, i.e., the adversary cannot get the session key generated in a previous session even if he/she can get both the private keys of two participants.

Two-factor security: To guarantee the security of the private key, the AMUA protocol for multi-server architectures should provide two-factor security, i.e., the adversary cannot extract the user's private key even if he/she could extract information stored in the user's smart card, which is issued by the registration center to store the user?s private key protected by his/her password.

No verifier table: To deduce the overhead of running the system and withstand attacks such as the stolen verifier attack related to the verifier table and the denial of service attack, the AMUA protocol for multi-server architectures should provide no verifier table, i.e., no verifier table should be needed to achieve mutual authentication.

No online registration center: To deduce the communication overhead, the AMUA protocol for multi-server architectures should provide no online registration center, i.e., no online registration center is needed to achieve mutual authentication.

Resistance of various attacks: To withstand various attacks prevalent in the mobile service system, the AMUA protocol for multi-server architectures should provide resistance of various attacks, i.e., the protocol should withstand the insider attack, the off-line password guessing attack, the user impersonation attack, the server spoofing attack, the modification attack, the stolen card attack, the stolen verifier table attack, the replay attack, and the man-in-the-middle attack.

4 The Proposed AMUA Protocol

We describe the proposed AMUA protocol in this section. The proposed AMUA protocol consists of five phases: the setup phase, the user registration phase, the server registration phase, the mutual authentication phase, and the password change phase.

4.1 Setup phase

By executing the following steps, the registration center (RC) selects the system private key and the system parameters.

1) RC chooses two elements G_1, G_2 of the same prime order q and a bilinear paring $e : G_1 \times G_1 \rightarrow G_2$. RC also selects a generator P of G_1.

2) RC chooses two random numbers $\tau, \hat{\tau} \in Z_q^*$ as the system private keys and calculate $g = e(P, P)$, $g_{pub} = g^{\tau}$ and $P_{pub} = \hat{\tau} \cdot P$.

3) RC chooses seven secure hash functions $h_0 : \{0,1\}^* \rightarrow Z_q^*$, $h_1 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$, $h_2 : \{0,1\}^* \rightarrow Z_q^*$, $h_3 : G_1 \times G_2 \times G_2 \rightarrow Z_q^*$, $h_4 : \{0,1\}^* \times G_1 \times G_2 \times G_2 \rightarrow Z_q^*$, $h_5 : \{0,1\}^* \times \{0,1\}^* \times G_2 \times G_2 \times G_2 \rightarrow Z_q^*$ and $h_6 : G_2 \rightarrow \{0,1\}^*$.

4) RC publishes the system parameters $params = \{G_1, G_2, e, q, P, g, g_{pub}, P_{pub}, h_0, h_1, h_2, h_3, h_4, h_5, h_6\}$

4.2 User registration phase

The user U_i registers with the registration center RC and gets his/her private key in this phase. As shown in Fig. 3, the steps below are carried out between U_i and RC.

1) U_i freely selects his/her identity ID_{U_i} and password PW_{U_i}. U_i also chooses a nonce b_i and sends the message $\{ID_{U_i}, h_0(ID_{U_i}, PW_{U_i}, b_{U_i})\}$ to RC.

2)RC generates a random number $w_{U_i} \in Z_q^*$, computes $g_{U_i} = g^{w_{U_i}}$, $\xi_{U_i} = h_1(ID_{U_i}, g_{U_i})$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$, $\psi_{U_i} = \tau_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$, and $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$. At last, RC sends $\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}\}$ to U_i using a secure channel.

3) U_i stores $\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}\}$ and b_{U_i} into his/her smart card and finishes the registration.

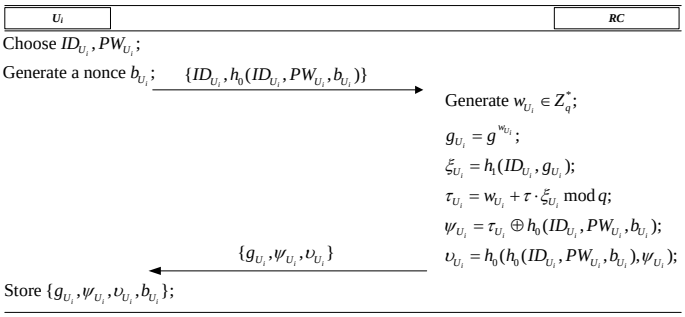

Fig. 3. The user registration phase

4.3 Server registration phase

The server S_j registers with the registration center RC and gets his/her private key in this phase. As shown in Fig. 4, the steps below are carried out between S_j and RC.

1) S_j freely chooses his/her identity ID_{S_j} and transmits it to RC.

2) RC computes $D_{S_j} = \frac{1}{\hat{\tau} + h_2(ID_{S_j})} \cdot P$ and transmits it back to S_j using a secure channel.

3) S_j stores D_{S_j} secretly and finishes the registration.

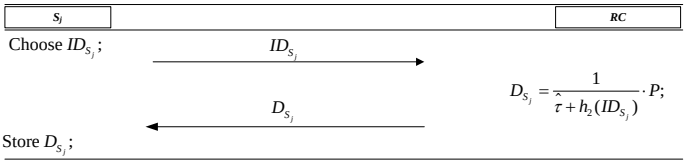

Fig. 4. The server registration phase

4.4 Mutual authentication phase

U_i and S_j authenticate each other and generate a session key for future secure communications in this phase. As shown in Fig. 5, the steps below are carried out between them.

1) U_i inputs his/her identity ID_{U_i} and password PW_{U_i}. The smart card checks whether the equation $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$ holds. If not, it stops the request; otherwise, it randomly selects a number $r_{U_i} \in Z_q^*$ and calculates $R_{U_i} = r_{U_i} \cdot (P_{pub} + h_2(ID_{S_j}) \cdot P)$, and $x = g^{r_{U_i}}$. At last, U_i sends R_{U_i} to S_j.

2) S_j generates a random number $r_{S_j} \in Z_q^*$, computes $x = e(R_{U_i}, D_{S_j})$, $y = g^{r_{S_j}}$ and $\alpha_{S_j} = h_3(R_{U_i}, x, y)$. At last, S_j sends $\{y, \alpha_{S_j}\}$ to U_i.

3) U_i checks whether the equation $\alpha_{S_j} = h_3(R_{U_i}, x, y)$ holds. If not, U_i rejects the session; otherwise, U_i calculates $\theta_{U_i} = h_4(ID_{U_i}, R_{U_i}, x, y)$, $\tau_{U_i} = \psi_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$, $\alpha_{U_i} = \tau_{U_i} + \theta_{U_i} \cdot r_{U_i}$, the session key $sk_{U_i} = h_5(ID_{U_i}, ID_{S_j}, x, y, y^{r_{U_i}})$ and $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha_{U_i})$. At last, U_i sends C_{U_i} to S_j.

4) S_j computes $(ID_{U_i}, g_{U_i}, \alpha_{U_i}) = h_6(x) \oplus C_{U_i}$, $\xi_{U_i} = h_1(ID_{U_i}, g_{U_i})$ and $\theta_{U_i} = h_4(ID_{U_i}, R_{U_i}, x, y)$. S_j checks whether the equation $g^{\alpha_{U_i}} = g_{U_i} \cdot g_{pub}^{\xi_{U_i}} \cdot x^{\theta_{U_i}}$. If not, S_j stops the request; otherwise, S_j computes the session key $sk_{S_j} = h_5(ID_{U_i}, ID_{S_j}, x, y, x^{r_{S_j}})$.

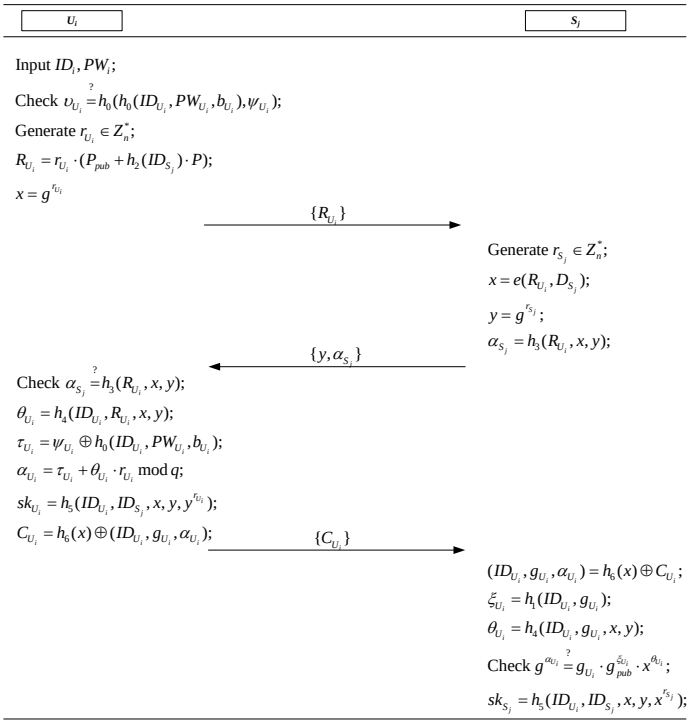

Fig. 5. The mutual authentication phase

Due to $g_{pub} = g^{\tau}$, $P_{pub} = \hat{\tau} \cdot P$, $g_{U_i} = g^{w_{U_i}}$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$, $D_{S_j} = \frac{1}{\hat{\tau} + h_2(ID_{S_j})} \cdot P$, $R_{U_i} = r_{U_i} \cdot (P_{pub} + h_2(ID_{S_j}) \cdot P)$, $x = g^{r_{U_i}}$, $y = g^{r_{S_j}}$ and $\alpha_{U_i} = \tau_{U_i} + \theta_{U_i} \cdot r_{U_i}$, then we have the following equations:

$$\begin{aligned} & e(R_{U_i}, D_{S_j}) \\ &= e(r_{U_i} \cdot (P_{pub} + h_2(ID_{S_j}) \cdot P), \tfrac{1}{\hat{\tau} + h_2(ID_{S_j})} \cdot P) \\ &= e(r_{U_i} \cdot (\hat{\tau} \cdot P + h_2(ID_{S_j}) \cdot P), \tfrac{1}{\hat{\tau} + h_2(ID_{S_j})} \cdot P) \\ &= e(r_{U_i} \cdot (\hat{\tau} + h_2(ID_{S_j})) \cdot P, \tfrac{1}{\hat{\tau} + h_2(ID_{S_j})} \cdot P) \\ &= e(P, P)^{r_{U_i} \cdot (\hat{\tau} + h_2(ID_{S_j})) \cdot \frac{1}{\hat{\tau} + h_2(ID_{S_j})}} \\ &= e(P, P)^{r_{U_i}} = g^{r_{U_i}} = x \end{aligned} \tag{1}$$

and

$$\begin{aligned} g^{\alpha_{U_i}} &= g^{\tau_{U_i}+\theta_{U_i}\cdot r_{U_i}} \\ &= g^{w_{U_i}+\tau\cdot\xi_{U_i}+\theta_{U_i}\cdot r_{U_i}} \\ &= g^{w_{U_i}}\cdot g^{\tau\cdot\xi_{U_i}}\cdot g^{\theta_{U_i}\cdot r_{U_i}} \\ &= g^{w_{U_i}}\cdot (g^{\tau})^{\xi_{U_i}}\cdot (g^{r_{U_i}})^{\theta_{U_i}} \\ &= g_{U_i}\cdot g_{pub}^{\xi_{U_i}}\cdot x^{\theta_{U_i}} \end{aligned} \quad (2)$$

Therefore, the correctness of the proposed AMUA protocol is proved.

4.5 Password change phase

U_i changes his/her password to a new one in this phase. The following steps are executed between U_i and his/her smart card.

1) U_i inputs his/her identity ID_{U_i} , old password PW_{U_i} and new password $PW^*_{U_i}$.

2) The smart card checks whether the equation $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$ holds. If not, the smart card stops the request; otherwise, it randomly select a number $b^*_{U_i}$, computes $\psi^*_{U_i} = \psi_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i}) \oplus h_0(ID_{U_i}, PW^*_{U_i}, b^*_{U_i})$,$\upsilon^*_{U_i} = h_0(h_0(ID_{U_i}, PW^*_{U_i}, b^*_{U_i}), \psi^*_{U_i})$. Finally, the smart card replaces $\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}, b_{U_i}\}$ and with $\{g_{U_i}, \psi^*_{U_i}, \upsilon^*_{U_i}, b^*_{U_i}\}$.

5 Security Analysis

We analyze the security of the proposed AMUA protocol for multi-server architectures in this section. First, we demonstrate that the proposed AMUA protocol is provably secure. Second, we prove that the proposed AMUA protocol can satisfy the requirements presented in Section II. Then, we present comparison of the security between the proposed AMUA protocol and two of the latest AMUA protocols.

5.1 Security model

Based on Choi *et al.'s* work [34, 35], we propose a security model for the AMUA protocol. The security of a AMUA protocol is defined by a game played between an adversary $\mathscr{A}$ and a challenger $\mathscr{C}$. Let Π^l_Λ denote the lth instance of the participant of $\Lambda \in \{U_i, S_j\}$. In the game, $\mathscr{A}$ can send queries to $\mathscr{C}$ and $\mathscr{C}$ answers them as follows.

- $h_i(m_i)$: When $\mathscr{A}$ executes the query with the message m_i, $\mathscr{C}$ generates a random number $r_i \in Z^*_q$, stores (m_i, r_i) in the list L_{h_i} and returns r_i to $\mathscr{A}$, where $i = 0, 1, \cdots 6$.
- $ExtractUser(ID_{U_i})$: When $\mathscr{A}$ executes the query with the user U_i's identity ID_{U_i}, $\mathscr{C}$ generates $U_i i$'s private key and stores it in the list L_{UK}.
- $ExtractServer(ID_{S_j})$: When $\mathscr{A}$ executes the query with the server S_j's identity ID_{S_j}, $\mathscr{C}$ generates S_j's private key and stores it in the list L_{SK}.
- $Send(\Pi^l_\Lambda)$: When $\mathscr{A}$ executes the query with the message m, $\mathscr{C}$ executes the AMUA protocol according to its specification and returns the result to $\mathscr{A}$.
- $Reveal(\Pi^l_\Lambda)$: When $\mathscr{A}$ executes the query, $\mathscr{C}$ returns the session key involved in Π^l_Λ to $\mathscr{A}$.
- $CorruptUser(ID_{U_i})$: When $\mathscr{A}$ executes the query with the user U_i's identity ID_{U_i}, $\mathscr{C}$ returns U_i's private key to $\mathscr{A}$.
- $CorruptServer(ID_{S_j})$: When $\mathscr{A}$ executes the query with the server S_j's identity ID_{S_j}, $\mathscr{C}$ returns S_j's private key to $\mathscr{A}$.
- $Test(\Pi^l_\Lambda)$: When $\mathscr{A}$ executes the query, $\mathscr{C}$ selects a random coin $b \in \{0, 1\}$. If $b = 1$, $\mathscr{C}$ sends the session key involved in Π^l_Λ to $\mathscr{A}$; otherwise ($b = 0$), $\mathscr{C}$ randomly selects a number with the same length of the session key and returns it to $\mathscr{A}$.

After executing the above queries, $\mathscr{A}$ outputs his/her guess b' about b generated in $Test$-query. We say $\mathscr{A}$ violates the authenticated key agreement (AKA) of the AMUA protocol Σ if he/she can guess b correctly. Let E_{A-W} denote the event that $\mathscr{A}$ can guess b successfully. The advantage that $\mathscr{A}$ attacking the AKA of the AMUA protocol Σ is defined as $Adv^{AKA}_\Sigma(\mathscr{A}) = |2Pr[E_{A-W}] - 1|$.

Definition 1(AKA-secure). We say a AMUA protocol Σ for multi-server architectures is AKA-secure if $Adv^{AKA}_\Sigma(\mathscr{A})$ is negligible for any polynomial adversary $\mathscr{A}$.

We say $\mathscr{A}$ violates the mutual authentication of the AMUA protocol Σ if he/she is able to generate a legal login message or a response message. Let E_{U-S} and E_{S-U} denote the events that $\mathscr{A}$ generates a legal login message or a response message. The advantage that $\mathscr{A}$ attacking the MA of the AMUA protocol Σ is defined as $Adv^{MA}_\Sigma(\mathscr{A}) = Pr[E_{U-S}] + Pr[E_{S-U}]$.

Definition 2(MA-secure). We say a AMUA protocol Σ for multi-server architectures is mutual authentication (MA)-secure if $Adv^{MA}_\Sigma(\mathscr{A})$ is negligible for any polynomial adversary $\mathscr{A}$.

5.2 Provable security

In this subsection, we show that the proposed AMUA protocol for multi-server architectures is AKA-secure and MA-secure in the security model described in the above subsection.

Lemma 1. No polynomial adversary against the proposed AMUA protocol for multi-server architectures can forge a legal login message with a non-negligible probability.

Proof. Suppose the adversary $\mathscr{A}$ forges a legal login message with a non-negligible probability ϵ. We show that there is a challenger $\mathscr{C}$ that can solve the DL problem in G_2 with a non-negligible probability.

Given an instance $(g, \theta = g^\tau)$ of the DL problem, the task of $\mathscr{C}$ is to compute $\tau \in Z^*_q$. $\mathscr{C}$ generates a random number $\hat{\tau} \in Z^*_q$, sets $g_{pub} \leftarrow \theta$, computes $P_{pub} = \hat{\tau} \cdot P$ and sends the system parameters $params = \{G_1, G_2, e, q, P, g, g_{pub}, P_{pub}, h_0, h_1, h_2, h_3, h_4, h_5, h_6\}$ to $\mathscr{A}$. $\mathscr{C}$ randomly picks a user's identity ID_{U_I} as the challenge identity and answers $\mathscr{A}$'s queries as follows:

- $h_i(m_i)$: $\mathscr{C}$ maintains a list L_{h_i} initialized empty. $\mathscr{C}$ checks if a tuple (m_i, r_i) exists in L_{h_i}. If it exists, $\mathscr{C}$ returns r_i to $\mathscr{A}$; otherwise, $\mathscr{C}$ randomly select a number r_i, inserts (m_i, r_i) into L_{h_i} and returns r_i to $\mathscr{A}$, where $i = 0, 1, \cdots, 6$.
- $ExtractUser(ID_{U_i})$: $\mathscr{C}$ maintains a list L_{UK} initialized empty. $\mathscr{C}$ checks if a tuple (ID_{U_i}, w_{U_i}, g_{U_i},τ_{U_i}) exists in L_{UK}. If it exists, $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$; otherwise, $\mathscr{C}$ executes the following operations:

- If $ID_{U_i} = ID_{U_I}$, $\mathscr{C}$ chooses two random numbers $w_{U_i}, \xi_{U_i} \in Z_q^*$, computes $g_{U_i} = g^{w_{U_i}}$, sets $h_1(ID_{U_i}, g_{U_i}) \leftarrow \xi_{U_i}, \tau_{U_i} \leftarrow \perp$, and inserts ($ID_{U_i}$, w_{U_i}, g_{U_i},τ_{U_i}) and (ID_{U_i}, g_{U_i}, ξ_{U_i}) into L_{UK} and L_{h_1} respectively. $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$.
- Otherwise($ID_{U_i} \neq ID_{U_I}$), $\mathscr{C}$ chooses two random numbers $w_{U_i}, \xi_{U_i} \in Z_q^*$, computes $g_{U_i} = g^{w_{U_i}} \cdot g_{pub}^{-\xi_{U_i}}$, sets $h_1(ID_{U_i}, g_{U_i}) \leftarrow \xi_{U_i}, \tau_{U_i} \leftarrow w_{U_i}$, and inserts ($ID_{U_i}$, w_{U_i}, g_{U_i},τ_{U_i}) and (ID_{U_i}, g_{U_i}, ξ_{U_i}) into L_{UK} and L_{h_1} respectively. $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$

- $ExtractServer(ID_{S_j})$: $\mathscr{C}$ maintains a list L_{SK} initialized empty. $\mathscr{C}$ checks if a tuple (ID_{S_j}, D_{S_j}) exists in L_{SK}. If it exists, $\mathscr{C}$ returns ID_{S_j} to $\mathscr{A}$; otherwise, $\mathscr{C}$ randomly selects a number $w_{S_j} \in Z_q^*$, computes $D_{S_j} = \frac{1}{\hat{\tau}+w_{S_j}} \cdot P$ and inserts (ID_{S_j}, D_{S_j}) and (ID_{S_j}, w_{S_j}) into L_{SK} and L_{h_2} respectively. At last, $\mathscr{C}$ returns ID_{S_j} to $\mathscr{A}$.
- $Send(\Pi_\Lambda^k)$: $\mathscr{C}$ checks if Λ and U_I are equal. If they are not equal, $\mathscr{C}$operates according to the specification of the proposed AMUA protocol because it knows Λ's private key; otherwise ($\Lambda = U_I$), $\mathscr{C}$ aborts the game.
- $Reveal(\Pi_\Lambda^l)$: $\mathscr{C}$ returns the session key involved in Π_Λ^l to $\mathscr{A}$.
- $CorruptUser(ID_{U_i})$: $\mathscr{C}$ looks up L_{UK} for the tuple (ID_{U_i}, w_{U_i}, g_{U_i},τ_{U_i}) and returns (g_{U_i},τ_{U_i}) to $\mathscr{A}$.
- $CorruptServer(ID_{S_j})$: $\mathscr{C}$ looks up L_{SK} for the tuple (ID_{S_j}, D_{S_j}) and returns D_{S_j} to $\mathscr{A}$.
- $Test(\Pi_\Lambda^l)$: $\mathscr{C}$ randomly selects a number with the same length of the session key and returns it to $\mathscr{A}$.

At last, $\mathscr{A}$ outputs a legal login message (R_{U_i}, C_{U_i}) corresponding to the user's identity ID_{U_i}, where $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha_{U_i})$. If $ID_{U_i} \neq ID_{U_I}$, $\mathscr{C}$ aborts the game. Based on the forking lemma [41], $\mathscr{A}$ can output another legal login message (R_{U_i}, C'_{U_i}) corresponding to ID_{U_i} if we repeat the simulation with a different choice of h_1, where $C'_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha'_{U_i})$. Due to the legality login messages, we get the following two equations.

$$g^{\alpha_{U_i}} = g_{U_i} \cdot g_{pub}^{\xi_{U_i}} \cdot x^{\theta_{U_i}} \tag{3}$$

and

$$g^{\alpha'_{U_i}} = g_{U_i} \cdot g_{pub}^{\xi'_{U_i}} \cdot x^{\theta_{U_i}} \tag{4}$$

Based on the above two equations, we derive the following:

$$\begin{aligned} g^{\alpha_{U_i}-\alpha'_{U_i}} &= \frac{g^{\alpha_{U_i}}}{g^{\alpha'_{U_i}}} \\ &= \frac{g_{U_i} \cdot g_{pub}^{\xi_{U_i}} \cdot x^{\theta_{U_i}}}{g_{U_i} \cdot g_{pub}^{\xi'_{U_i}} \cdot x^{\theta_{U_i}}} = \frac{g_{pub}^{\xi_{U_i}}}{g_{pub}^{\xi'_{U_i}}} \\ &= g_{pub}^{\xi_{U_i}-\xi'_{U_i}} = g^{\tau \cdot (\xi_{U_i}-\xi'_{U_i})} \end{aligned} \tag{5}$$

$\mathscr{C}$ outputs $(\alpha_{U_i} - \alpha'_{U_i}) \cdot (\xi_{U_i} - \xi'_{U_i})^{-1}$ as the answer to the DL problem. The probability that $\mathscr{C}$ solves the DL problem is analyzed below. For convenience, some events are defined as follows.

- E_1: $\mathscr{C}$ does abort in any $Send$-query.
- E_2: $\mathscr{C}$ outputs a valid login message.
- E_3: ID_{U_i} and ID_{U_I} are equal.

Let q_{send} and q_{h_1} denote the number of $Send$- queries and h_1- queries executed in the game. We can get $Pr[E_1] \geq (1-\frac{1}{q_{send}+1})^{q_{send}}$, $Pr[E_2|E_1] \geq \epsilon$ and $Pr[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{h_1}}$. Therefore, the non-negligible probability that $\mathscr{C}$ can solve the DL problem is

$$\begin{aligned} &Pr[E_1 \wedge E_2 \wedge E_3] \\ &= Pr[E_3|E_1 \wedge E_2] \cdot Pr[E_2|E_1] \cdot Pr[E_1] \\ &\geq \frac{(1-\frac{1}{q_{send}+1})^{q_{send}}}{q_{h_1}} \cdot \epsilon \end{aligned} \tag{6}$$

This contradicts with the hardness of the DL problem in G_2. Therefore, we conclude that no polynomial adversary against the proposed AMUA protocol for multi-server architectures can forge a login message with a non-negligible probability.

Lemma 2. No polynomial adversary against the proposed AMUA protocol for multi-server architectures can forge a response message with a non-negligible probability.

Proof. Suppose the adversary $\mathscr{A}$ can forge a response message with a non-negligible probability ϵ. We will show that there is a challenger $\mathscr{C}$ can solve the k-mBIDH problem with a non-negligible probability.

Given an instance ($\Theta = \hat{\tau} \cdot P, \Phi = \eta \cdot P, \frac{1}{\hat{\tau}+\alpha_1} \cdot P, \frac{1}{\hat{\tau}+\alpha_2} \cdot P, \cdots, \frac{1}{\hat{\tau}+\alpha_k} \cdot P \in G_1$) of the k-mBIDH problem, the task of $\mathscr{C}$ is to compute $e(P,P)^{\frac{\eta}{\hat{\tau}+\alpha}}$. $\mathscr{C}$ generates a random number $\tau \in Z_q^*$, computes $g_{pub} = g^\tau$, sets $P_{pub} \leftarrow \Theta$ and sends the system parameters $params = \{G_1, G_2, e, q, P, g, g_{pub}, P_{pub}, h_0, h_1, h_2, h_3, h_4, h_5, h_6\}$ to $\mathscr{A}$. $\mathscr{C}$ randomly picks a server's identity ID_{S_J} as the challenge identity. $\mathscr{C}$ answers the h_i-query ($i = 0, 1, \cdots, 6$), $Test$-query, $Reveal$-query, $CorruptUser$-query and $CorruptServer$-query as he/she does in the above lemma. He/she also answers other queries as follows.

- $ExtractUser(ID_{U_i})$: $\mathscr{C}$ maintains a list L_{UK} initialized empty. $\mathscr{C}$ checks if a tuple (ID_{U_i}, w_{U_i}, g_{U_i},τ_{U_i}) exists in L_{UK}. If it exists, $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$; otherwise, $\mathscr{C}$ generates two random numbers $w_{U_i} \in Z_q^*$, computes $g_{U_i} = g^{w_{U_i}}$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$,and inserts ($ID_{U_i}$, w_{U_i}, g_{U_i},τ_{U_i}) and (ID_{U_i}, g_{U_i}, ξ_{U_i}) into L_{UK} and L_{h_1} respectively. $\mathscr{C}$ returns ID_{U_i} to $\mathscr{A}$.
- $ExtractServer(ID_{S_j})$: $\mathscr{C}$ maintains a list L_{SK} initialized empty. $\mathscr{C}$ checks if a tuple (ID_{S_j}, D_{S_j}) exists in L_{SK}. If it exists, $\mathscr{C}$ returns ID_{S_j} to $\mathscr{A}$; otherwise, $\mathscr{C}$ checks if ID_{S_j} and ID_{S_J} are equal as follows.
  - If $ID_{S_j} = ID_{S_J}$, $\mathscr{A}$ sets $h_2(ID_{S_J}) \leftarrow \alpha, D_{S_j} \leftarrow \perp$ and inserts (ID_{S_J}, α) and (ID_{S_j}, D_{S_j}) into L_{h_2} and L_{SK} respectively.
  - Otherwise if $ID_{S_j} \neq ID_{S_J}$, $\mathscr{A}$ randomly chooses $\alpha_j \in \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$, sets $h_2(ID_{S_J}) \leftarrow \alpha_j, D_{S_j} \leftarrow \frac{1}{\hat{\tau}+\alpha_2} \cdot P$ and inserts (ID_{S_J}, α) and (ID_{S_J}, D_{S_j}) into L_{h_2} and L_{SK} respectively.
- $Send(\Pi_\Lambda^l)$: $\mathscr{C}$ checks if Λ and S_J are equal, if they are equal, $\mathscr{C}$ aborts the game; otherwise, $\mathscr{C}$ does the following operations.
  - If $m =$ "$start$" and Λ's partner is S_J, $\mathscr{C}$ returns Φ to $\mathscr{A}$.

– Otherwise, $\mathscr{C}$ behaves according to the specification of the proposed AMUA protocol.

At last, $\mathscr{A}$ outputs a legal response message (R_{U_i}, C_{U_i}) corresponding to the server's identity ID_{S_j}. $\mathscr{A}$ randomly chooses a tuple (R, x, y) from the list L_{h_3} and outputs x as the solution of the k-mBIDH problem. The probability that $\mathscr{C}$ solves the DL problem is analyzed below. For convenience, some events are defined as follows.

- E_1: $\mathscr{C}$ does abort in any $Send$-query.
- E_2: $\mathscr{C}$ outputs a legal response message.
- E_3: ID_{S_j} and ID_{S_J} are equal.
- E_4: $\mathscr{C}$ chooses a correct tuple from L_{h_3}.

Let q_{send}, q_{h_2} and q_{h_3} denote the number of $Send$-query, h_2-query and h_3-query executed in the game. We can get $Pr[E_1] \geq (1 - \frac{1}{q_{send}+1})^{q_{send}}$, $Pr[E_2|E_1] \geq \epsilon$, $Pr[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{h_2}}$ and $Pr[E_4|E_1 \wedge E_2 \wedge E_3] \geq \frac{1}{q_{h_3}}$. Therefore, the non-negligible probability that $\mathscr{C}$ can solve the DL problem is computed as follows.

$$\begin{aligned} &Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \\ &= Pr[E_4|E_1 \wedge E_2 \wedge E_3] \cdot Pr[E_3|E_1 \wedge E_2] \cdot \\ &Pr[E_2|E_1] \cdot Pr[E_1] \\ &\geq \frac{(1-\frac{1}{q_{send}+1})^{q_{send}}}{q_{h_2} \cdot q_{h_3}} \cdot \epsilon \end{aligned} \tag{7}$$

This contradicts with the hardness of the k-mBIDH problem. Therefore, we can conclude that no polynomial adversary against the proposed AMUA protocol for multi-server architectures can forge a response message with a non-negligible probability.

Theorem 1. The proposed AMUA protocol for multi-server architectures is MA-secure if the DL-problem and the k-mBIDH problem are hard.

Proof. Based on Lemma 1 and Lemma 2, we get that no polynomial adversary can forge a legal login message or a legal response message if the DL-problem and the k-mBIDH problem are hard. Therefore, we conclude that the proposed AMUA protocol for multi-server architectures is MA-secure.

Theorem 2. The proposed AMUA protocol for multi-server architectures is AKA-secure if the CDH problem is hard.

Proof. Suppose the adversary $\mathscr{A}$ correctly guesses b used in $Test$-query with a non-negligible probability ϵ. We will show there is a challenger $\mathscr{C}$ that can solve the CDH problem with a non-negligible probability.

Let E_{sk} denote the event that $\mathscr{A}$ gets the correct session key. Since the probability that $\mathscr{A}$ correctly guesses the value b is at least $\frac{1}{2}$, we can get $Pr[E_{sk}] \geq \frac{\epsilon}{2}$.

Let E_{TU} and E_{TS} denote the events $\mathscr{A}$ uses in the $Test$-query to a user's instance and a server's instance respectively. Let E_{U-S} denote the event that $\mathscr{A}$ can violate the user-to-server authentication. We get the following two equations.

$$\begin{aligned} &\tfrac{\epsilon}{2} \leq Pr[E_{sk}] \\ &= Pr[E_{sk} \wedge E_{TU}] + Pr[E_{sk} \wedge E_{TS} \wedge E_{U-S}] + \\ &Pr[E_{sk} \wedge E_{TS} \wedge \neg E_{U-S}] \\ &\leq Pr[E_{sk} \wedge E_{TU}] + Pr[E_{U-S}] + \\ &Pr[E_{sk} \wedge E_{TS} \wedge \neg E_{U-S}] \end{aligned} \tag{8}$$

and

$$\begin{aligned} &Pr[E_{sk} \wedge E_{TU}] + Pr[E_{sk} \wedge E_{TS} \wedge \neg E_{U-S}] \\ &\geq \tfrac{\epsilon}{2} - Pr[E_{U-S}] \end{aligned} \tag{9}$$

Since $E_{TS} \wedge \neg E_{U-S}$ and E_{TU} are equal, then we get

$$Pr[E_{sk} \wedge E_{TU}] \geq \frac{\epsilon}{4} - \frac{Pr[E_{U-S}]}{2} \tag{10}$$

Therefore, we the probability is computed as follows.

$$\begin{aligned} &Pr[sk = h_5(ID_{U_i}, ID_{S_j}, x, y, k)|x, y, k \leftarrow G_2] \\ &\geq \tfrac{\epsilon}{4} - \tfrac{Pr[E_{U-S}]}{2} \end{aligned} \tag{11}$$

According to the proof of Lemma 1, we know that $Pr[E_{U-S}]$ is negligible. Therefore, $\frac{\epsilon}{4} - \frac{Pr[E_{U-S}]}{2}$ is non-negligible. Suppose that $x = g^a$ and $y = g^b$ for some unknown $a, b \in Z_q^*$. Given an instance (x,y) of the CDH problem, $\mathscr{A}$ computes $k = g^{a \cdot b}$ with a non-negligible probability $\frac{\epsilon}{4} - \frac{Pr[E_{U-S}]}{2}$, i.e., $\mathscr{C}$ can use $\mathscr{A}$ to solve the CDH problem with a non-negligible probability. This contradicts with the hardness of the CDH problem. Therefore, we can conclude that the proposed AMUA protocol for multi-server architectures is AKA-secure if the CDH problem is hard.

5.3 Analysis of security requirements

In this subsection, we show that the proposed AMUA protocol for multi-server architectures satisfies all the security requirements described in Section II.

Single registration: According to the specification of the proposed AMUA protocol, the user only needs to register with the registration center and can freely login into any server in the system after that. Therefore, the proposed AMUA protocol is able to provide the single registration.

Mutual authentication: According to the proofs of Lemma 1 and Lemma 2, there is no polynomial adversary that can forge a login message or a response message. Thus, the user and the server authenticate the other participant by verifying the validity of received message. Therefore, the proposed AMUA protocol is able to provide mutual authentication.

User anonymity: Based on the description of the proposed AMUA protocol, the user's identity ID_{U_i} is only included in $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha_{U_i})$. To extract ID_{U_i} from C_{U_i}, the adversary has to compute $x = g^{r_{U_i}}$ from $R_{U_i} = r_{U_i} \cdot (P_{pub} + h_2(ID_{S_j}) \cdot P)$, i.e., the adversary has to solve the k-mBIDH problem. Then, we know that the proposed AMUA protocol is able to provide user anonymity since the k-mBIDH problem is hard.

Un-traceability: According to the specification of the proposed AMUA protocol, the user generates a new random number $r_{U_i} \in Z_q^*$ to compute $R_{U_i} = r_{U_i} \cdot (P_{pub} + h_2(ID_{S_j}) \cdot P)$, $x = g^{r_{U_i}}$ and $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha_{U_i})$. Due the randomness of r_{U_i}, the adversary cannot find any relation between two messages sent by U_i and cannot trace his/her action. Therefore, the proposed AMUA protocol is able to provide un-traceability.

Session key agreement: According to the specification of the proposed AMUA protocol, both two participants calculate the session key $sk = h_5(ID_{U_i}, ID_{S_j}, x, y, x^{r_{S_j}}) =$

$h_5(ID_{U_i}, ID_{S_j}, x, y, y^{r_{U_i}})$, which can be used in the future communications. Therefore, the proposed AMUA protocol is able to provide the session key agreement.

Perfect forward secrecy: Suppose the adversary steals both private keys of the user and the server. We also assume that the adversary intercepts messages ($R_{U_i}, y = g^{r_{S_j}}, \alpha_{S_j}, C_{U_i}$) transmitted between the user and the server. Using the server's private key, the adversary computes $x = e(R_{U_i}, D_{S_j}) = g^{r_{U_i}}$. To get the session key $sk = h_5(ID_{U_i}, ID_{S_j}, x, y, x^{r_{S_j}}) = h_5(ID_{U_i}, ID_{S_j}, x, y, y^{r_{U_i}})$, the adversary has to compute $x^{r_{S_j}} = y^{r_{U_i}} = g^{r_{U_i} \cdot r_{S_j}}$ from $x = g^{r_{U_i}}$ and $y = g^{r_{S_j}}$, i.e., he/she has to solve the CDH problem. Then, the proposed AMUA protocol is able to provide the perfect forward secrecy since the CDH problem is hard.

Two-factor security: Suppose the adversary steals the user's smart card. By using the side channel attack, the adversary can extract the data $(\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}\}, b_{U_i}$ stored in the smart card, where $g_{U_i} = g^{w_{U_i}}$, $\xi_{U_i} = h_1(ID_{U_i}, g_{U_i})$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$, $\psi_{U_i} = \tau_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$, and $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$. The adversary can guess a password PW'_{U_i}. However, he/she cannot verify its correctness because he/she does not know the user's identity. On the other hand, the adversary only knows the user's password and cannot get the user's private key τ_{U_i}. Therefore, both types of adversaries cannot impersonate the user to the serverand the proposed AMUA protocol provides two-factor security.

No verifier table: According to the specification of the proposed AMUA protocol, both two participants just need to store their own private keys and no verifier table is maintained by the registration center. Therefore, the proposed AMUA protocol provides no verifier table.

No on-line registration center: According to the specification of the proposed AMUA protocol, two participants can authenticate each other without the help of the registration centre. Therefore, the proposed AMUA protocol does not need an on-line registration center.

Resistance of various attacks: We will show that the proposed AMUA protocol can resist the insider attack, the off-line password guessing attack, the user impersonation attack, the server spoofing attack, the modification attack, the stolen card attack, the stolen verifier table attack, the replay attack and the man-in-the-middle attack. The details are presented as follows.

- **Insider attack**: Suppose an insider in the system gets the user's information $\{ID_{U_i}, h_0(ID_{U_i}, PW_{U_i}, b_{U_i})\}$. The adversary can guess a password PW'_{U_i}. However, he/she cannot verify its correctness because the user's password PW_{U_i} is protected by the secure hash function and the random number b_{U_i}. Therefore, the insider cannot get the user's password and the proposed AMUA protocol withstands the insider attack.
- **Off-line password guessing attack**: Suppose the adversary steals the user's smart card. Using the side channel attack, the adversary can extract the data $(\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}\}, b_{U_i}$ stored in the smart card, where $g_{U_i} = g^{w_{U_i}}$, $\xi_{U_i} = h_1(ID_{U_i}, g_{U_i})$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$, $\psi_{U_i} = \tau_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$, and $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$. The adversary can guess a password PW'_{U_i}. However, he/she cannot verify its correctness because he/she does not know the user's identity. Therefore, the adversary cannot get the user's password and the proposed AMUA protocol is able to withstand the off-line password guessing attack.
- **User impersonation attack**: According to the proof of Lemma 1, we conclude that no adversary is able to forge a legal login message without the user's private key. Therefore, the server can find out about the attack through verifying the validity of the received login message. Therefore, the proposed AMUA protocol withstands the user impersonation attack.
- **Server spoofing attack**: According to the proof of Lemma 2, we know that no adversary is able to generate a legal response message without the server's private key. Therefore, the user can find out about the attack through verifying the validity of the received response message. Therefore, the proposed AMUA protocol withstands the server spoofing attack.
- **Modification attack**: According to the proof of Lemma 1, we know that $(\{g_{U_i}, g_{U_i}, \alpha_{U_i})$ is a digital signature of the login message and no polynomial can forge a legal one. The server can find any modification by checking if the equation $g^{\alpha_{U_i}} = g_{U_i} \cdot g_{pub}^{\xi_{U_i}} \cdot x^{\theta_{U_i}}$ holds. Besides, $\alpha_{S_j} = h_3(R_{U_i}, x, y)$ is the message authentication code of the response message $\{y, \alpha_{S_j}\}$ under the key $x = e(R_{U_i}, D_{S_j})$. The user can find out about any modification of the response message because the hash function h_3 is secure. Therefore, the proposed AMUA protocol can resist the modification attack.
- **Stolen card attack**: Suppose the adversary steals the user's smart card. Using the side channel attack, the adversary extracts the data $(\{g_{U_i}, \psi_{U_i}, \upsilon_{U_i}\}, b_{U_i}$ stored in the smart card, where $g_{U_i} = g^{w_{U_i}}$, $\xi_{U_i} = h_1(ID_{U_i}, g_{U_i})$, $\tau_{U_i} = w_{U_i} + \tau \cdot \xi_{U_i}$, $\psi_{U_i} = \tau_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$, and $\upsilon_{U_i} = h_0(h_0(ID_{U_i}, PW_{U_i}, b_{U_i}), \psi_{U_i})$. The adversary can guess a password PW'_{U_i}. However, he/she cannot verify its correctness because he/she does not know the user's identity. Therefore, the proposed AMUA protocol is able to withstand the stolen card attack.
- **Stolen verifier table attack**: According to the specification of the proposed AMUA protocol, none of three roles in it needs to maintain a verifier table for mutual authentication. Therefore, the proposed AMUA protocol is able to withstand the stolen verifier table attack.
- **Replay attack**: According to the specification of the proposed AMUA protocol, both two participants generate new random numbers ($r_{U_i}, r_{S_j} \in Z_q^*$) and compute $g_{U_i} = g^{r_{U_i}}$ and $g_{S_j} = g^{r_{S_i}}$, which are involved in the login message and the response message respectively. Due to the freshness of g_{U_i} and g_{S_j}, the user and the server can find the replay of messages by checking the validity of the received message. Therefore, the proposed AMUA protocol withstands the replay attack.

- **Man-in-the-middle attack**: Based on the above analysis, we conclude that the proposed AMUA protocol provides the mutual authentication between two participants. Therefore, the proposed AMUA protocol resists the man-in-the-middle attack.

5.4 Security comparisons

In this subsection, we compare the security of the proposed AMUA protocol with two of the latest AMUA protocols [36, 37] recently proposed for multi-server architectures without using an on-line registration center. For convenience, we let $SR-1, SR-2, SR-3, SR-4, SR-5, SR-6, SR-7, SR-8, SR-9$ and $SR-10$ denote single registration, mutual authentication, user anonymity, un-traceability, session key agreement, perfect forward secrecy, two-factor security, not using the verifier table, no on-line registration center and resistance of various attacks respectively. The security comparisons of the three AMUA protocols are listed in Table 1.

TABLE 1
Security comparisons

	Liao and Hsiao's protocol	Hsieh and Leu's protocol	**The proposed protocol**
$SR-1$	Yes	Yes	**Yes**
$SR-2$	Yes	Yes	**Yes**
$SR-3$	No[37]	No[38]	**Yes**
$SR-4$	No[37]	No[38]	**Yes**
$SR-5$	Yes	Yes	**Yes**
$SR-6$	Yes	Yes	**Yes**
$SR-7$	Yes	Yes	**Yes**
$SR-8$	No	No	**Yes**
$SR-9$	Yes	Yes	**Yes**
$SR-10$	No[37]	No[38]	**Yes**

According to Table 1, both Liao et al.'s AMUA protocol [36] and Hsieh et al.'s AMUA protocol [37] cannot provide user anonymity, un-traceability, and resiliency to various attacks. In addition, both of these protocols need to use verifier table. In contrast, the proposed AMUA protocol can satisfy all nine security requirements. Therefore, the proposed AMUA protocol has better security than the two recently proposed AMUA protocols.

6 PERFORMANCE ANALYSIS

We analyze the computation and communication costs of the proposed AMUA protocol in this section. We also compare its performance with the other two bench marked AMUA protocols [36, 37].

To get a trusted security level (the security level of 1024-bits RSA algorithm), a Ate pairing $e : G_1 \times G_1 \rightarrow G_2$ is used in our experiments, where G_1 with order q is generated by a point on a super singular elliptic curve $E(F_p) : y^2 = x^3 + 1$ defined on the finite field F_p, q is a 160-bits prime number and p is a 512-bits prime number.

6.1 Analysis of computation cost

We present the running time of various operations performed in the proposed protocol and we compare the results with those obtained from Liao et al.'s and Hsieh et al.'s protocols in this section. We use the following notations for the following running times in this paper:

- T_{bp}: The running time of a bilinear paring operation.
- T_{sm}: The running time of a scalar multiplication operation in G_1.
- T_{mtp}: The running time of a map-to-point hash function in G_1.
- T_{pa}: The running time of a point addition operation in G_1.
- T_{exp}: The running time of an exponentiation operation in G_2.
- T_{mul}: The running time of a multiplication operation in G_2.
- T_h: The running time of a general hash operation.

We have implemented the above operations on a mobile device (Samsung Galaxy S5 with a Quad-core 2.45G processor, 2G bytes memory and the Google Android 4.4.2 operating system) and a personal computer (Dell with an I5-4460S 2.90GHz processor, 4G bytes memory and the Window 8 operating system) using the MIRACL library [42]. In our experiments, the mobile device and the personal computer are the user and the server respectively. The running time of those operations is listed in Table 2.

TABLE 2
The running time of related operations (millisecond)

	The user	The server
T_{mtp}	33.582	5.493
T_{bp}	32.713	5.427
T_{sm}	13.405	2.165
T_{pa}	0.081	0.013
T_{exp}	2.249	0.339
T_{mul}	0.008	0.001
T_h	0.056	0.007

The Liao and Hsiao's AMUA protocol needs to execute one hash-to-point operation, seven scalar multiplication operations in G_1, one point addition operation in G_1 and five general hash operations. Therefore, the user's running time is $T_{mtp} + 7 \times T_{sm} + T_{pa} + 5 \times T_h \approx 127.778$ milliseconds. The server in Liao and Hsiao's AMUA protocol needs to execute one hash-to-point operation, two bilinear paring operations, five scalar multiplication operations in G_1, one point addition operation in G_1 and four general hash operations. Therefore, the server's running time is $T_{mtp}+2\times T_{bp}+5\times T_{sm}+T_{pa}+4\times T_h \approx 27.213$ milliseconds.

In Hsieh and Leu's AMUA protocol, the user needs to execute one hash-to-point operation, seven scalar multiplication operations in G_1, one point addition operation in G_1 and eight general hash operations. Therefore, the user's running time is $T_{mtp} + 7 \times T_{sm} + T_{pa} + 8 \times T_h \approx 127.946$ milliseconds. The server in Hsieh and Leu's AMUA protocol needs to execute one hash-to-point operation, two bilinear paring operations, five scalar multiplication operations in G_1, one point addition operation in G_1 and four general hash operations. Therefore, the server's running time is $T_{mtp}+2\times T_{bp}+5\times T_{sm}+T_{pa}+3\times T_h \approx 27.206$ milliseconds.

The user in the proposed AMUA protocol needs to execute two scalar multiplication operations in G_1, one point addition operation in G_1, two exponentiation operations in G_2 and eight general hash operations. Therefore, the user's

running time is $2 \times T_{sm} + T_{pa} + 2 \times T_{exp} + 8 \times T_h \approx 31.837$ milliseconds. The server in the proposed AMUA protocol needs to execute one bilinear pairing operation, four exponentiation operations in G_2, two multiplication operations in G_2 and five general hash operations. Therefore, the server's running time is $T_{bp} + 4 \times T_{exp} + 2 \times T_{mul} + 5 \times T_h \approx 6.82$ milliseconds.

TABLE 3
Computation cost comparisons (millisecond)

	Liao and Hsiao's protocol	Hsieh and Leu's protocol	**The proposed protocol**
User	T_{mtp}+7·T_{sm}+ T_{pa}+5·T_h ≈ 127.778	T_{mtp}+7·T_{sm}+ T_{pa}+8·T_h ≈ 127.946	$\boldsymbol{2 \cdot T_{sm} + T_{pa}}$+ $\boldsymbol{2 \cdot T_{exp} + 8 \times T_h}$ $\boldsymbol{\approx 31.837}$
Server	T_{mtp}+2·T_{bp}+ 5·T_{sm}+T_{pa}+ 4·T_h≈ 27.213	T_{mtp}+2·T_{bp}+ 5·T_{sm}+T_{pa}+ 3·T_h≈ 27.206	$\boldsymbol{T_{bp} + 4 \cdot T_{exp}}$+ $\boldsymbol{2 \cdot T_{mul} + 5 \cdot T_h}$ $\boldsymbol{\approx 6.82}$

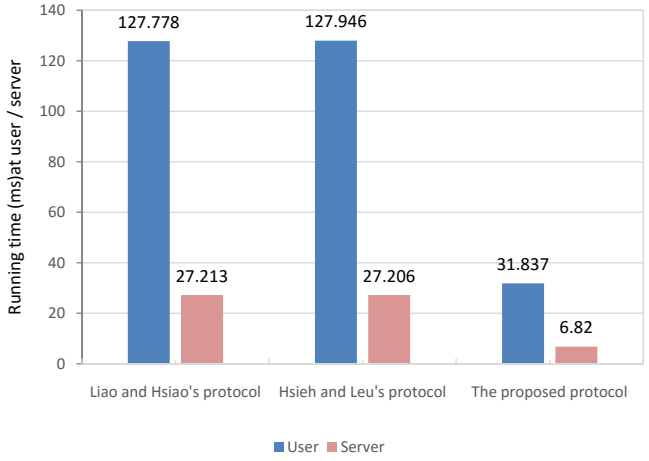

Fig. 6. Computation cost comparisons

The computation cost comparisons are demonstrated in Table 3 and Fig. 6. From the results in Table 3 and Fig. 6, the proposed AMUA protocol has lower computation cost than the other two AMUA protocols for both the user and server sides. At the user side, the percentage improvement for computation cost with our proposed approach is 75.08 and 75.12 lower as compared to Liao et al.'s and Hsieh et al.'s protocols respectively. At the server side, the percentage improvement for computation cost with our proposed approach is 74.94 and 74.93 lower as compared to Liao et al.'s and Hsieh et al.'s protocols respectively.

6.2 Analysis of communication cost

According to the above implementation, we know that the lengths of p and q are 512 bits and 160 bits respectively. Therefore, the size of an element in G_1 or G_2 and the length of hash function's output are 1024 bits and 160 bits respectively. Suppose the length of the user's identity is 32 bits. The communication cost is analyzed as follows.

The user and the server in Liao and Hsiao's AMUA protocols send messages $(ID_{U_i}, M_i, B_{ij}, R_i, Auth_{ij})$ and $(Auth_{ji}, K_{ji}, R_j)$ respectively to the other party, where ID_{U_i} is the user's identity, $M_i, B_{ij}, R_i, K_{ji}, R_j \in G_1$ and $Auth_{ij}, Auth_{ji} \in Z_q^*$. Therefore, the total communication cost of Liao and Hsiao's AMUA protocols is $32 + 1024 + 1024 + 1024 + 160 + 160 + 1024 + 1024 = 5472$ bits.

The user and the server in Hsieh and Leu's AMUA protocol send messages $(xAuth_i, C_m, M_i, B_{ij}, R_i, Auth_{ij})$ and $(Auth_{ji}, K_j, R_j)$ respectively to the other party, where ID_{U_i} is the user's identity, $xAuth_i, C_m, M_i, B_{ij}, R_i, W_j, K_j, R_j \in G_1$ and $Auth_{ij}, Auth_{ji} \in Z_q^*$. Therefore, the total communication cost of Liao and Hsiao's AMUA protocol is $1024+1024+1024+1024+1024+160+160+1024+1024 = 7488$ bits.

The user and the server in the proposed AMUA protocol send messages (R_{U_i}, C_{U_i}) and (y, α_{S_j}) to the other party, where $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, \alpha_{U_i})$, ID_{U_i} is the user's identity, $R_{U_i} \in G_1$, $y \in G_2$ and $\alpha_{U_i}, \alpha_{S_j} \in Z_q^*$. Therefore, the total communication cost with our proposed AMUA protocol is $1024 + 32 + 1024 + 160 + 1024 + 160 = 3424$ bits.

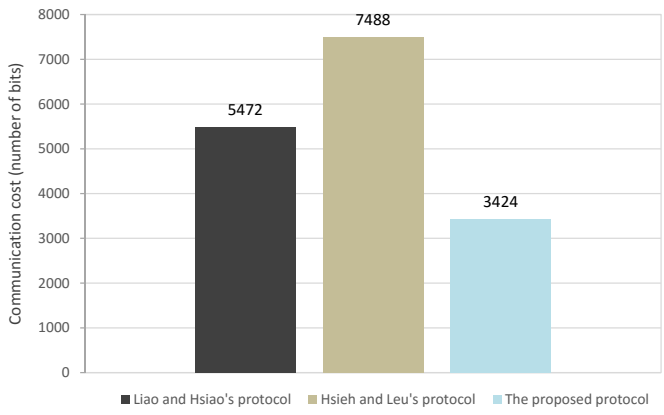

Fig. 7. Communication cost comparison

The communication cost comparisons are shown in Fig. 7. The percentage improvement for communication cost with our proposed approach is 37.43 and 54.27 lower as compared to Liao et al.'s and Hsieh et al.'sprotocols respectively.

7 Conclusion

To bring more convenience to mobile users in multi-server architectures, several mobile user authentication protocols without the on-line registration center using the SCPKC have been proposed in the last several years. However, most of them suffer from serious attacks and have unsatisfactory performance in terms of computation and communication costs. This paper proposes a new mobile user authentication protocol for multi-server architectures without the need of an on-line registration center. The security analysis shows that the proposed protocol is provably secure in the random oracle model and satisfies the security requirements in the mobile system with multi-server architectures. In addition, the performance analysis results show that the proposed protocol has lower communication and computation costs. The strong resilience of our proposed protocol against various types of attacks also make it suitable for use by a wide range of applications to maintain security at various levels.

8 ACKNOWLEDGMENTS

We thank Professor Vrizlynn Thing and the anonymous reviewers for the constructive comments which help improve the quality and presentation of this paper.

REFERENCES

[1] G. Hunt, "Number of smartphone users in US skyrockets, but overreliance is a concern", Teck News, http://www.tecnews.tk, 2015.
[2] Q. Jiang, J. Ma, G. Li, and L. Yang, "An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks," Wireless Personal Communications, vol. 68, no. 4, pp. 1477-1491, 2013.
[3] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," Wireless Personal Communications, vol. 77, no. 2, pp. 1489-1506, 2014.
[4] L. Lamport, "Password authentication with insecure communication," Communications of the ACM, vol. 24, no. 11, pp. 770?LC772, 1981.
[5] X. Huang, Yang Xiang, Ashley Chonka, Jianying Zhou, and Robert H. Deng, "A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 8, pp. 1390-1397, 2011.
[6] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 6, pp. 568-581, 2014.
[7] P. Guo, J. Wang, B. Li, and S. Lee, "A variable threshold-value authentication architecture for wireless mesh networks," Journal of Internet Technology, vol. 15, no. 6, pp. 929-936, 2015.
[8] J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," Journal of Internet Technology, vol. 16, no. 1, pp. 171-178, 2015.
[9] D. He, S. Zeadally, N. Kumar, and J. Lee, "One-to-many authentication for access control in mobile pay-tv systems," Science China Information Sciences, DOI:10.1007/s11432-015-5469-5, 2016.
[10] D. He, S. Zeadally, N. Kumar, and J. Lee, "Anonymous authentication for wireless body area networks with provable security," IEEE Systems Journal, DOI: 10.1109/JSYST.2016.2544805, 2016.
[11] L. Li, L. Lin, and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1498-1504, 2001.
[12] W. Juang, "Efficient multi-server password authenticated key agreement using smart cards," IEEE Transactions on Consumer Electronics, vol. 50, no. 1, pp. 251-255, 2004.
[13] J. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," Computers & Security, vol. 27, no. 3, pp. 115-121, 2008.
[14] C. Chang, and J. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," In: International Conference on Cyberworlds, pp. 417-422, 2004.
[15] W. Tsaur, J. Li, and W. Lee, "An efficient and secure multi-server authentication scheme with key agreement," Journal of Systems and Software, vol. 85, no. 4, pp. 876-882, 2012.
[16] Y. Liao,and 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.
[17] H. Hsiang, and W. 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.
[18] C. Lee, T. Lin, and R. Chang, "A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards," Expert Systems with Applications, vol. 38, no. 11, pp. 13863-13870, 2011.
[19] S. Sood, A. 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.
[20] X. Li, Y. Xiong, J. Ma, and W 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.
[21] B. Wang, and M. Ma, "A smart card based efficient and secured multi-server authentication scheme," Wireless Personal Communications, vol. 68, no. 2, pp. 361-378, 2013.
[22] D. Mishra, A. Das, and S. Mukhopadhyay "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," Expert Systems with Applications, vol. 41, no. 18, pp. 8129-8143, 2014.
[23] J. Sahoo, A. Das, and A. Goswami, "An efficient approach for mining association rules from high utility itemsets," Expert Systems with Applications, vol. 42, no. 13, pp. 5754-5778, 2015.
[24] X. Li, Q. Wen, W. Li W, et al., "A biometric-based password authentication with key exchange scheme using mobile device for multi-server environment," Applied Mathematics & Information Sciences, vol. 9, no. 3, pp. 1123-1137, 2015.
[25] Y. Lu, L. Li, X. Yang, et al., "Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards," PLoS ONE, DOI:10.1371/journal.pone.0126323, 2015.
[26] I. Lin, M. Hwang, and L. Li, "A new remote user authentication scheme for multi-server architecture," Future Generation Computer Systems, vol. 19, no. 1, pp. 13-22, 2003.
[27] X. Cao, and S. Zhong, "Breaking a remote user authentication scheme for multiserver architecture," IEEE Communications Letters, vol. 10, no. 8, pp. 580-581, 2006.
[28] E. Yoon, and K. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem, Journal of Supercomputing, vol. 63, no. 1, pp. 235-255, 2013.
[29] H. Kim, W. Jeon, K. Lee, Y. Lee, and D. Won, "Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme," In: 12th International Conference on Computational Science and Its Applications, pp. 391-406, 2012.
[30] D. He, and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," IEEE Systems Journal, vol. 9, no. 3, pp. 816-823, 2015.
[31] V. Odelu, A. 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.
[32] J. Zhang, J. Ma, X. Li X, et al., "A secure and efficient remote user authentication scheme for multi-server environments uUsing ECC," KSII Transactions on Internet and Information Systems, vol. 8, no. 8, pp. 2930-2947, 2014.
[33] Y. Tseng, S. Huang, T. Tsai, et al., "List-free id-based mutual authentication and Key agreement protocol for multi-server architectures," IEEE Transactions on Emerging Topics in Computing, DOI: 10.1109/TETC.2015.2392380, 2015.
[34] K. Choi, J. Hwang, D. Lee, et al., "ID-based authenticated key agreement for low-power mobile devices," In: 10th Australasian Conference on Information Security and Privacy, pp. 494-505, 2005.
[35] Y. Chuang, and Y. Tseng, "Towards generalized ID-based user authentication for mobile multi-server environment," International Journal of Communication Systems, vol. 25, no. 4, pp. 447-460, 2012.
[36] Y. Liao and C. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," Future Generation Computer Systems, vol. 29, no. 3, pp. 886-900. 2013.
[37] W. Hsieh, and J. Leu, "An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," Journal of Supercomputing, vlo. 70, no. 1, pp. 133-148. 2014.
[38] R. Amin, and G. Biswas, "Design and analysis of bilinear pairing based mutual authentication and key agreement protocol ssable in multi-server environment," Wireless Personal Communications, vol. 84, no. 1, pp. 439-462, 2015.
[39] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 85, no. 2, pp. 481-484, 2002.
[40] K. Choi, J. Hwang, D. Lee, and I. Seo, "ID-based authenticated key agreement for low-power mobile devices," In: the 10th Australasian Conference on Information Security and Privacy(ACISP), pp. 494-505, 2005.

[41] D. Pointcheval, and J. Stern, "Security arguments for digital signatures and blind signatures," Journal of Cryptology, vol. 13, no. 3, pp. 361-396, 2000.
[42] Shamus Software Ltd., Miracl library, http://www.shamus.ie/index.php?page=home

Debiao He received his Ph.D. degree in applied mathematics from School of Mathematics and Statistics, Wuhan University in 2009. He is currently an Associate Professor of the State Key Lab of Software Engineering, Computer School, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Sherali Zeadally is an Associate Professor in the College of Communication and Information at the University of Kentucky. He received the Bachelor and Doctorate degrees in computer science from the University of Cambridge, England, and the University of Buckingham, England, respectively. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England.

Neeraj Kumar received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra, India. He is now an Associate Professor in the Department of Computer Science and Engineering, Thapar University, Patiala, Punjab (India). He is a member of IEEE. His research is focused on mobile computing, parallel/distributed computing, multi-agent systems, service oriented computing, routing and security issues in mobile ad hoc, sensor and mesh networks. He has more than 100 technical research papers in leading journals such as-IEEE TII, IEEE TIE, IEEE TDSC, IEEE ITS, IEEE TWPS, IEEE SJ,IEEE ComMag, IEEE WCMag, IEEE NetMag and conferences. His research is supported from DST, TCS and UGC. He has guided many students leading to M.E. and Ph.D.

Wei Wu received her Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2011. She is currently an Associate Professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. Her research focus is on public key cryptography and its applications. She has published more than 40 papers in refereed international journals and conferences.