Design of Secure Authenticated Key Management Protocol for Cloud Computing Environments

Wei Li, Xuelian Li, Juntao Gao, and Haiyu Wang

Abstract—With the maturity of cloud computing technology in terms of reliability and efficiency, a large number of services have migrated to the cloud platform. To convenient access to the services and protect the privacy of communication in the public network, three-factor Mutual Authentication and Key Agreement (MAKA) protocols for multi-server architectures gain wide attention. However, most of the existing three-factor MAKA protocols don't provide a formal security proof resulting in various attacks on the related protocols, or they have high computation and communication costs. And most of the three-factor MAKA protocols haven't a dynamic revocation mechanism, which leads to malicious users can not be promptly revoked. To address these drawbacks, we propose a provable dynamic revocable three-factor MAKA protocol that achieves the user dynamic management using Schnorr signatures and provides a formal security proof in the random oracle. Security analysis shows that our protocol can meet various demands in the multi-server environments. Performance analysis demonstrates that the proposed scheme is well suited for computing resource constrained smart devices. The full version of the simulation implementation proves the feasibility of the protocol.

Index Terms—Protocols, cloud computing, smart cards, authentication, bioceramics

1 Introduction

In the recent decade, cloud computing technology has been completely commercialized. It can not only improve service efficiency but also reduce costs. More and more companies are putting their services on the cloud platform for development, management and maintenance. This not only reduces the local maintenance burden for these enterprises, but also provides unified security and operation management for all services on the third-party cloud platform, as shown in Fig. 1. Although third-party cloud platforms have more powerful technologies and more standard technical specifications to ensure that the servers run in a relatively secure environment, users and servers communicate in the public network. Therefore, authentication and key agreement are critical for the communication security. The use of mutual authentication and key agreement (MAKA) protocols not only prevent attackers from abusing server resources, but also prevent malicious attackers posing as the server to obtain the user's information. Therefore, the MAKA protocols have been extensively studied since Lamport proposed a password-based authentication protocol [1].

Earlier MAKA protocols [2], [3], [4] are designed for single-server architecture. As Internet users grow exponentially, the number of cloud servers rendering different services has also grown significantly. For the single-server architecture, it is difficult for users to maintain a variety of passwords for each server. To improve user experience, many scholars propose more flexible MAKA protocols [5], [6], [7], [8], [9], [10] for multi-server environments. Combined with the unified management features of the cloud platform, such protocols can be conveniently applied. The protocols for multi-server architectures model as shown in Fig. 2, users and cloud servers only need to register in the registration center (RC) to mutual authentication and key agreement.

In the multi-server environments, the MAKA protocols can be further divided into two categories, two-factor MAKA protocols, namely identity, password, and three-factor MAKA protocols, namely identity, password, biometrics. The works in [11], [12] have shown that the password-based MAKA protocols suffer from several attacks such as guessing password attack. The cost of the password guessing attack on password-based protocol becomes lower and lower as the rapid development of computers. On the other hand, users usually utilize simple letters or numbers as their passwords, and even a large number of users directly use the default password if the smart devices don't require the user to modify the password mandatory. In order to solve this problem, several biometrics-based MAKA protocols [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] have been proposed. Due to the uniqueness, availability and non-transferability of biometrics keys (palm print, iris, finger print etc.), the three-factor MAKA protocols for

- *W. Li, X. Li, and H. Wang were with the School of Mathematics and Statistics, Xidian University, Xi'an, Shaanxi Province, China. E-mail: liwei3013@126.com, {xuelian202, slimshady225}@163.com.*
- *J. Gao was with the School of Telecommunication and Engineering, Xidian University, Xi'an, Shaanxi Province, China. E-mail: jtgao@mail.xidian.edu.cn.*

Manuscript received 17 Oct. 2018; revised 22 Jan. 2019; accepted 2 Apr. 2019. Date of publication 9 Apr. 2019; date of current version 13 May 2021. (Corresponding author: Xuelian Li.) Recommended for acceptance by M. L. Das. Digital Object Identifier no. 10.1109/TDSC.2019.2909890

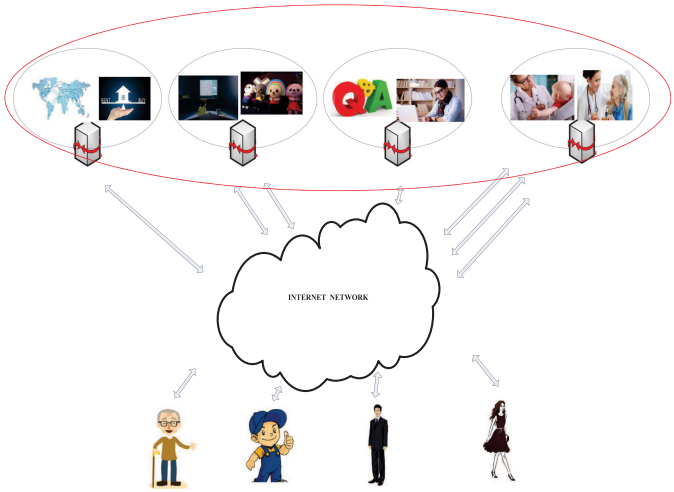


Fig. 1. Cloud service environment.

multi-server environments provide more security than the two-factor protocols. In view of the openness of wireless networks, an adversary can intercept, modify, delete and replay any communication messages. Anonymity and untraceability are also indispensable part of the MAKA protocols [24], [25], [26], [27], [28] to resist the above-mentioned attacks. However, the current three-factor MAKA protocols still have the following defects.

1) *Security vulnerabilities*. Most of the existing MAKA protocols based on the three factors haven't a formal proof, but some informal security analysis. And some protocols were embed insecure factors such as key authentication factors easily extracted. We will analyze such weaknesses in the security comparisons and cryptanalysis subsection.
2) *Incomplete basic functions*. Some important basic functions, such as dynamic user management, authentication phase without RC, are not considered in most MAKA protocols.
3) *High cost*. Some three-factor MAKA protocols didn't take full account of their actual application environment, which results these protocols are not suited for the limited resource of the devices.

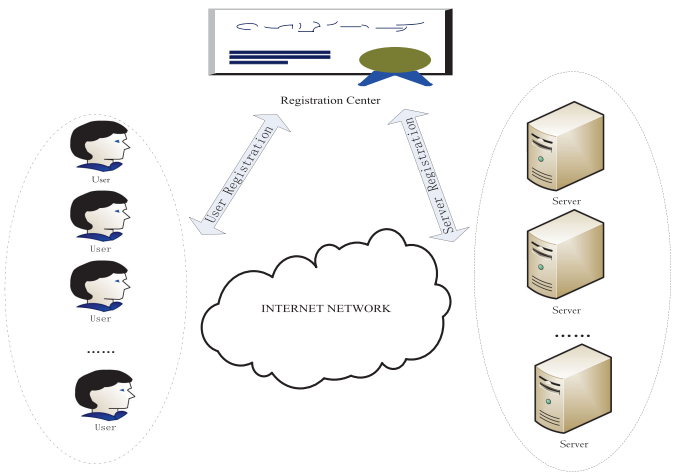


Fig. 2. Multi-service architectures of the protocol model.

Therefore, it still a challenge to design an effective three-factor MAKA protocol for achieving secure communication between user and server.

1.1 Related Works

In 2001, Li et al. [5] introduced the concept of authentication protocol for multi-server environments and proposed the first password-based MAKA protocol using the neural network. Thanks to the complicated neural network, Li et al.'s protocol isn't suitable for smart devices with limited computing power. To improve efficiency, Juang [6] proposed a MAKA protocol for multi-server architectures by using hash functions and symmetric key cryptosystems. In the same year, Chang et al. [7] pointed out that Juang's protocol is flawed in terms of efficiency. They proposed a more efficient MAKA scheme for multi-server environments. However, in their protocol RC shares system private key with all servers. This will undoubtedly result in many security vulnerabilities. To improve security, some new MAKA protocols [8], [9] using hash functions and symmetric-key cryptosystems had also been proposed. In 2013, Liao et al. [10] proposed a multi-server remote user authentication protocol using self-certified public keys for mobile clients. However, their scheme doesn't establish a shared session key and the communication cost is unacceptable.

Given the fact that wireless networks are open environment, the privacy protection is also considered in such protocols. To provide user anonymity, Das et al. [24] proposed the first dynamic two-factor authentication scheme which uses dynamic pseudo-identities instead of a user's true identity. Unfortunately, in 2009 Wang et al. [25] pointed out that Das et al.'s protocol fails to provide mutual authentication, user anonymity and they proposed an improved version. However, Yeh et al. [26] and Wen et al. [27] found that Wang et al.'improved version is vulnerable to impersonation attack and is incapable of providing user anonymity, respectively. In 2016, based on self-certified public key cryptography, He et al. [28] proposed a provable security anonymous MAKA protocol for multi-server architectures. The protocols discussed above are password-based, but such protocols are insecure under off-line guessing password attack. We will analyze the protocol [28] that has such security weakness in the security comparisons and cryptanalysis subsection.

To address this weakness, based on fingerprint, Lee et al. [13] proposed a authentication protocol using smart card. However, Lin et al. [14] and Chang et al. [15] found that Lee et al.'s protocol suffers from the masquerade attack and the conspiring attack, respectively. To enhance security, Kim et al. [16] proposed a new biometrics-based authentication protocol using smart card. Unfortunately, Scott [17] pointed out that Kim et al.'s protocol can be completely compromised by a passive adversary. Later, Khan et al. [18] found that Lin et al.'s scheme also suffers from the server spoofing attack and proposed an improved version. For multi-server architectures, Yoon et al. [19] proposed a biometrics-based authentication protocol using elliptical curve cryptosystem (ECC) and smart card. Unfortunately, Kim et al. [20] and He [29] pointed out that Yoon et al.'scheme is insecure under the offline password-guessing attack, the privileged insider attack and the impersonation attack. Later, He et al. [21]

proposed a robust biometrics-based authentication scheme for multi-server architectures. However, Odelu et al. [22] point out that He et al.'s protocol suffers from the known session-specific temporary information attack, the impersonation attack and so on. However, each authentication and key agreement in Odelu et al.'protocol requires the involvement of RC. In 2017, Reedy et al. [23] also proposed a biometrics-based MAKA protocol for multi-server environments. Unfortunately, after our analysis in the security comparisons and cryptanalysis subsection of this paper, their protocol is vulnerable the server impersonation attack and the man-in-the-middle attack. On the other hand, the MAKA protocol is also widely used in other environments, such as Passive Internet of Things [30], Vehicles in Smart City [31], and Mobile Devices [32], [33].

1.2 Our Contributions

In this paper, we propose a dynamic revocable three-factor mutual authentication and key agreement (3DRMAKA) protocol which has more comprehensive functions, reliable security and relatively higher execution efficiency. Our contribution can be summarized as follows:

1) We design a three-factor MAKA protocol which implements three-factor security. And we show that the proposed protocol can meet the demands of multi-server architectures such as anonymity, non-traceability, resistance password guessing attack and smart card extraction attack, and so on.
2) Our scheme achieves the user's dynamic management. In our protocol, users can be dynamically revoked to promptly prevent attacks from malicious users. Without a dynamic revocation mechanism, RC can't punish malicious users in a timely manner. This may result in such malicious users still active in the network to communicate with other servers.
3) In the random oracle, we provide a formal proof of the proposed protocol based on BDH, CDH and Schnorr signatures unforgeability assumptions. We show that the proposed protocol is mutual authentication secure and authenticated key agreement secure.
4) Our protocol has a good execution efficiency. Especially on the client side, the computation cost of our scheme is the lowest in the related existing protocols. This shows that our protocol is more suitable for device mobiles with limited computing resource. And, to prove that the protocol is technically sound, we programmatically simulate the proposed protocol.

1.3 Organization of the Paper

The rest of the paper is organized as follows. In Section 2, the basic knowledge will be introduced. We present a new 3DRMAKA protocol in Section 3. In Section 4, we prove the proposed protocol and perform the analysis of related protocols. We compare the performance of our protocol with related protocols in Section 5. Finally, we conclude the paper in Section 6.

2 Preliminaries

To the integrity of the paper, some relevant definitions will be presented below. Let G_1, G_2 be two cyclic groups of prime order q, in which G_1 is a subgroup of the group of points on elliptic curve over a finite field F_p and G_2 is multiplicative group. Suppose G_1 is generated by generator P. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ if it has the below properties:

- Bilinearity: we have $e(a \cdot S, b \cdot T) = e(S,T)^{ab}$, for all $a, b \in Z_q^*$ and $S, T \in G_1$.
- Non-degeneracy: the $e(P,P) \neq 1_{G_2}$ holds.
- Computability: There must be an efficient algorithm to compute $e(S,T)$ for all $S, T \in G_1$.

A fuzzy extractor could extract a random string ∂ if given biometric input B. One key property of the fuzzy extractor is that it can output the same random string even though the input changes, but it remains close. To recover ∂ from a new biometric B^*, a consistent random auxiliary string θ is generated and used for the following operations. The following definition is the formal description of the fuzzy extractor.

Definition 1 (Fuzzy Extractor). *[34] A fuzzy extractor includes two algorithms (Gen, Rep).*

- *When receiving biometric input B, the Gen algorithm will output a random string ∂ and a random auxiliary string θ.*
- *When receiving a close biometric input B^* and the corresponding random auxiliary string θ, the Rep algorithm will output ∂.*

Since the proposed scheme is based on the unforgeability of the Schnorr signatures [35], the following is a brief introduction of this signatures.

- *Initialization*. Let p, q be two big prime such that $q|p-1$, choose a generator g of G_2, a one-way hash function h and a private key $x \in Z_q^*$, and then computes the public key $g_{pub} = g^x$.
- *Signature*. The signer chooses a random number $k \in Z_q^*$, and then computes $r = g^k$, $e = h(m,r)$, $s = (xe + k) mod q$ for message m. Finally, the signer outputs the signatures (s, e, m).
- *Verification*. Upon receiving of the signature messages (m, s, e), the verifier computes $r = g^s \cdot g_{pub}^{-e}$, and then checks $e? = h(m,r)$. if the equation holds, this signatures are valid and vice versa.

At the Eurocrypt 1996, Pointcheval and Stern [36] have shown that Schnorr signatures are provably security based on the hardness of the discrete logarithm (DL) problem in the random oracle model. However, in 2014, Nils et al. [37] pointed out that the reduction of the paper [36] from DL to breaking Schnorr signatures is not tight. In their paper, they introduced a new meta-reduction technique to prove the security of Schnorr signatures.

Definition 2. *Based on [37], there is no probabilistic polynomial adversary can forge Schnorr signatures with a non-negligible probability.*

Definition 3 (Computational Diffie-Hellman (CDH) Problem). *Given (g, g^a, g^b), where $g \in G_2$ and $a, b \in Z_q^*$.*

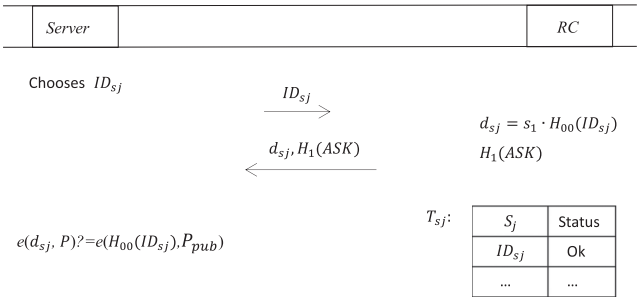


Fig. 3. The server registration phase.

There is no valid algorithm to compute g^{ab} within polynomial time.

Definition 4 (Bilinear Diffie-Hellman (BDH) Problem). *Given $(P, a \cdot P, b \cdot P, c \cdot P)$, where $P \in G_1$ and $a, b, c \in Z_q^*$, it is difficult to compute $e(P, P)^{abc}$ within polynomial time.*

3 The Proposed 3DRMAKA Protocol

In this section, we describe the proposed 3DRMAKA protocol. The proposed protocol can be divided into eight phases: Initialization Phase, Server Registration Phase, Users Registration Phase, Time Key Update Phase, Login and Mutual Authentication Phase, Password and Biometrics Change Phase, New Server Update Phase and Dynamic Revocation Phase.

3.1 Initialization Phase

The RC initializes the system private key and the system parameters in this phase.

1) According the definition of bilinear pairing, RC selects two groups G_1, G_2 of the same prime order q and a bilinear paring e:$G_1 \times G_1 \rightarrow G_2$. Then, RC also chooses two random numbers $s_1, s_2 \in Z_q^*$ as the system private keys, a generator P of G_1 and a share key ASK.
2) RC calculates $P_{pub} = s_1 \cdot P$, $g = e(P, P)$, $g_{pub} = g^{s_2}$ and chooses nine secure hash functions: H_{00}:$\{0,1\}^* \rightarrow G_1$, H_{01}:$Z_q^* \rightarrow G_1$, H_1:$\{0,1\}^* \rightarrow Z_q^*$, H_2:$\{0,1\}^* \times Z_q^* \rightarrow Z_q^*$, H_3:$\{0,1\}^* \times Z_q^* \times Z_q^* \rightarrow Z_q^*$, H_4:$\{0,1\}^{2q} \rightarrow Z_q^*$, H_5:$G_2 \times Z_q^* \times \{0,1\}^* \rightarrow Z_q^*$, H_6:$G_2 \times \{0,1\}^* \times Z_q^* \rightarrow Z_q^*$, H_7:$G_2 \times G_2 \times G_2 \times G_2 \times Z_q^* \times \{0,1\}^* \rightarrow Z_q^*$.
3) RC publics the system parameters $\{q, G_1, G_2, P_{pub}, g, g_{pub}, e, P, H_{00}, H_{01}, H_1, H_2, H_3, H_4, H_5, H_6, H_7\}$.

3.2 Server Registration Phase

In this phase, the server S_j registers with the RC in order to be an authorized server of the network. As shown in Fig. 3, the processes of communication between S_j and RC are performed as follows.

1) S_j transmits registration request with his/her identity ID_{sj} to RC securely.
2) RC computers $d_{sj} = s_1 \cdot H_{00}(ID_{sj})$, $H_1(ASK)$ and delivers them back to S_j under a secure channel.
3) Upon receiving $(d_{sj}, H_1(ASK))$, the S_j can validate the private key by checking whether the equation $e(d_{sj}, P)? = e(H_{00}(ID_{sj}), P_{pub})$ holds. If the equation holds, the private key is valid and vice versa.
4) RC maintains a table database T_{sj}, which stores the status of the corresponding registration servers.

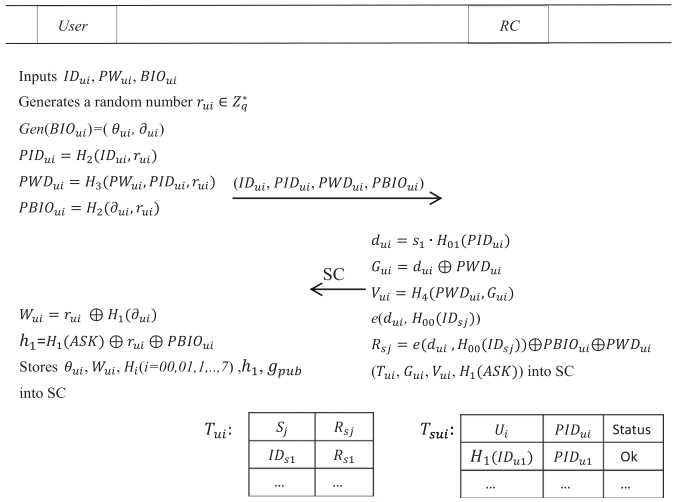


Fig. 4. The user registration phase.

3.3 Users Registration Phase

Under multi-server environments, a new user U_i who wants to access the services provided by S_j must register with the RC. As shown in Fig. 4, the steps of communication between U_i and RC are carried out as follows.

1) U_i inputs his/her identity ID_{ui}, password PW_{ui} and biometrics BIO_{ui}. Then U_i generates a random number $r_{ui} \in Z_q^*$ and computers $Gen(BIO_{ui}) = (\theta_{ui}, \partial_{ui})$, $PID_{ui} = H_2(ID_{ui}, r_{ui})$, $PWD_{ui} = H_3(PW_{ui}, PID_{ui}, r_{ui})$, and $PBIO_{ui} = H_2(\partial_{ui}, r_{ui})$. Finally U_i transmits $(ID_{ui}, PID_{ui}, PWD_{ui}, PBIO_{ui})$ to RC through an out of band (secure) channel.
2) RC calculates $d_{ui} = s_1 \cdot H_{01}(PID_{ui})$, $G_{ui} = d_{ui} \oplus PWD_{ui}$, $V_{ui} = H_4(PWD_{ui}, G_{ui})$ for corresponding user. To reduce the user's computational burden, RC computes $e(d_{ui}, H_{00}(ID_{sj}))$ for the user based on valid server entries in the T_{sj} table database. And then generates a T_{ui} table whose data is $R_{sj} = e(d_{ui}, H_{00}(ID_{sj})) \oplus PWD_{ui} \oplus PBIO_{ui}$. RC inserts $(T_{ui}, G_{ui}, V_{ui}, H_1(ASK))$ into smartcard(SC) for corresponding user and sends SC to the user under a secure channel. Finally, RC creates a form database T_{sui} with hash of user identity information in which also stores the status of the corresponding registration users to dynamically manage users.
3) Upon receiving SC, U_i computes $W_{ui} = r_{ui} \oplus H_1(\partial_{ui})$, $h_1 = H_1(ASK) \oplus r_{ui} \oplus PBIO_{ui}$ and then replaces $H_1(ASK)$ with $\theta_{ui}, W_{ui}, H_i(i = 00, 01, 1, \ldots, 7), h_1, g_{pub}$.

3.4 Time Key Update Phase

In order to manage users dynamically, RC periodically updates the time key to legitimate users in the T_{sui}. As shown in Fig. 5, the steps below are run between RC and U_i.

1) Before updating the time key for a user, RC checks the status of the corresponding anonymous user

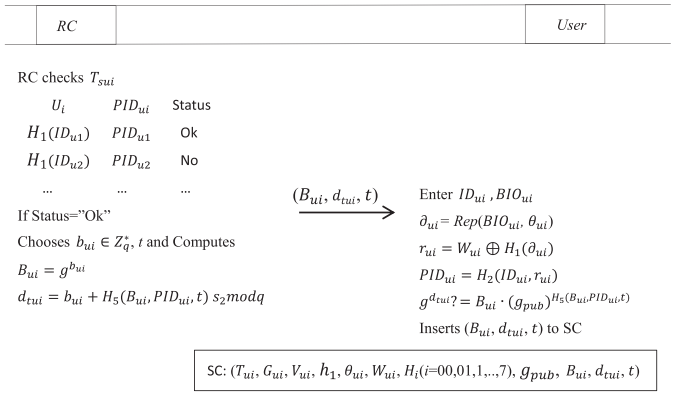


Fig. 5. Time key update phase.

PID_{ui} in T_{sui}. Then, RC chooses a random number $b_{ui} \in Z_q^*$, valid time period t and computes $B_{ui} = g^{b_{ui}}$, $d_{tui} = b_{ui} + H_5(B_{ui}, PID_{ui}, t) \cdot s_2 mod q$ for the legitimate user. RC transmits (B_{ui}, d_{tui}, t) to the corresponding user through a public channel.

2) Upon receiving (B_{ui}, d_{tui}, t), the user first enters ID_{ui}, BIO_{ui} and then computes $\partial_{ui} = Rep(BIO_{ui}, \theta_{ui})$, $r_{ui} = W_{ui} \oplus H_1(\partial_{ui})$, $PID_{ui} = H_2(ID_{ui}, r_{ui})$. U_i checks $g^{d_{tui}}? = B_{ui} \cdot g_{pub}^{H_5(B_{ui}, PID_{ui}, t)}$ to verify the validity of the time key. If the equation holds, the user inserts (B_{ui}, d_{tui}, t) to SC. The elements contained in the SC are $\{T_{ui}, G_{ui}, V_{ui}, h_1, \theta_{ui}, W_{ui}, H_i(i = 00, 01, 1, \ldots, 7), g_{pub}, B_{ui}, d_{tui}, t\}$.

3.5 Login and Mutual Authentication Phase

As shown in Fig. 6, the user must first complete the smart card login. Then, U_i and S_j can authenticate each other and negotiate a session key. This phase is described as follows.

1) U_i inputs his/her ID_{ui}, PW_{ui} and BIO_{ui} into SC. SC uses its stored information to calculate $\partial_{ui} = Rep(BIO_{ui}, \theta_{ui})$, $r_{ui} = Wui \oplus H_1(\partial_{ui})$, $PID_{ui} = H_2(ID_{ui}, r_{ui})$, $PWD_{ui} = H_3(PW_{ui}, PID_{ui}, r_{ui})$, $PBIO_{ui} = H_2(\partial_{ui}, r_{ui})$. SC validates whether the equation $V_{ui}? = H_4(PWD_{ui}, G_{ui})$ holds. If the equation holds, the user logins successfully, otherwise, rejects the request.
2) Using the information in table T_{ui}, the user chooses a target server ID_{sj}. SC generates a random number $N_1 \in Z_q^*$ and calculates $H_1(ASK) = h_1 \oplus r_{ui} \oplus PBIO_{ui}$, $k_{ui} = g^{N_1}$, $F_{ui} = PID_{ui} \oplus H_6(k_{ui}, ID_{sj}, H_1(ASK))$. Then, SC sends $(F_{ui}, k_{ui}, B_{ui}, d_{tui}, t)$ to the corresponding server S_j.
3) Upon receiving the messages, S_j first verifies the time key correctness by checking $g^{d_{tui}}? = B_{ui} \cdot g_{pub}^{H_5(B_{ui}, PID_{ui}, t)}$, where $PID_{ui} = F_{ui} \oplus H_6(k_{ui}, ID_{sj}, H_1(ASK))$. If the equation holds, S_j continues next steps; otherwise, rejects the request.
4) S_j chooses a random number $N_2 \in Z_q^*$ and computes $k_{1t} = e(d_{sj}, H_{01}(PID_{ui}))$, $k_{sj} = g^{N_2}$ and $D_{sj} = H_7(k_{1t}, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. Then, S_j transmits (D_{sj}, k_{sj}) to the corresponding U_i.
5) Upon receiving (D_{sj}, k_{sj}), U_i computes $k_1 = R_{sj} \oplus PWD_{ui} \oplus PBIO_{ui}$ and then checks whether the $D_{sj}? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$ holds. If the equation holds, U_i computes $k_u = k_{sj}^{N_1}$, $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, the session key $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Then, U_i sends D_{ui} to S_j.
6) When S_j receives the message D_{ui}, he/she checks whether the $D_{ui}? = H_7(g_{pub}, k_{ui}, k_{sj}, k_{1t}, PID_{ui}, ID_{sj})$ holds. If the equation holds, S_j computes $k_s = k_{ui}^{N_2}$ and the session key $sk = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$.

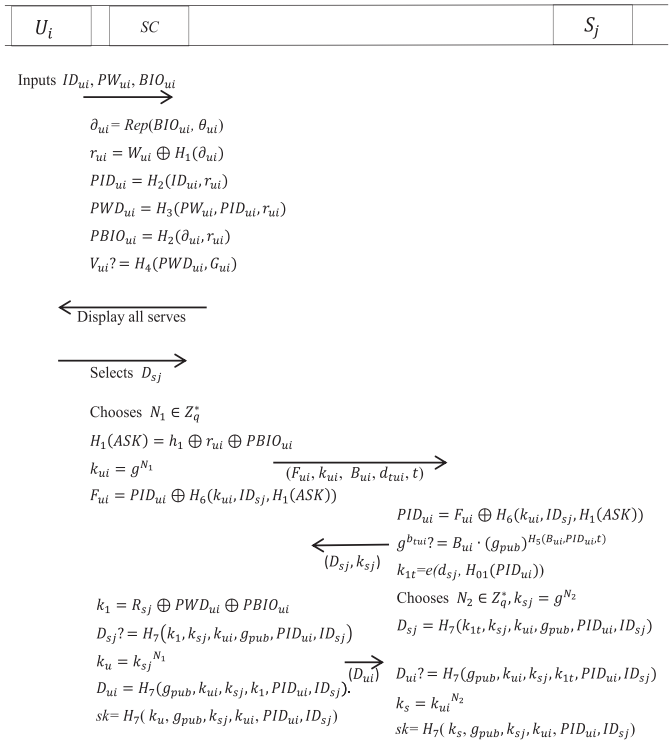


Fig. 6. Login and Mutual Authentication phase.

3.6 Password and Biometrics Change Phase

In order to reduce the burden of RC, the user can change the password and biometrics without RC in our scheme. The detailed steps are as follows.

1) U_i enters his/her ID_{ui}, PW_{ui}, BIO_{ui} and new PW_{nui}, BIO_{nui} into SC. The SC computes $\partial_{ui} = Rep(BIO_{ui}, \theta_{ui})$, $r_{ui} = W_{ui} \oplus H_1(\partial_{ui})$, $PID_{ui} = H_2(ID_{ui}, r_{ui})$, $PWD_{ui} = H_3(PW_{ui}, PID_{ui}, r_{ui})$, $PBIO_{ui} = H_2(\partial_{ui}, r_{ui})$. The SC validates whether the equation $V_{ui}? = H_4(PWD_{ui}, G_{ui})$ holds. If not, SC terminates the follow-up operations.
2) SC computes $PWD_{nui} = H_3(PW_{nui}, PID_{ui}, r_{ui})$, $Gen(BIO_{nui}) = (\theta_{nui}, \partial_{nui})$, $PBIO_{nui} = H_2(\partial_{nui}, r_{ui})$ and then uses them to update other relevant informations in SC, namely $R_{nsj} = R_{sj} \oplus PWD_{ui} \oplus PBIO_{ui} \oplus PBIO_{nui} \oplus PWD_{nui} \rightarrow T_{nui}$, $G_{nui} = G_{ui} \oplus PWD_{ui} \oplus PWD_{nui}$, $V_{nui} = H_4(PWD_{nui}, G_{nui})$, $h_{n1} = h_1 \oplus PBIO_{ui} \oplus PBIO_{nui}$, $W_{nui} = W_{ui} \oplus H_1(\partial_{ui}) \oplus H_1(\partial_{nui})$. Finally, SC replaces $(T_{ui}, G_{ui}, V_{ui}, h_1, W_{ui}, \theta_{ui})$ with $(T_{nui}, G_{nui}, V_{nui}, h_{n1}, W_{nui}, \theta_{nui})$.

3.7 New Server Update Phase

When a new server wants to join the network, the proposed protocol can insert the new server information without RC . The detailed processes are as follows.

1) When a new server S_{nj} has finished registration, the RC updates the T_{sj} and broadcasts ID_{nsj} all valid users in the T_{sui}.
2) Upon receiving ID_{nsj}, users input ID_{ui}, PW_{ui}, BIO_{ui} and ID_{nsj} into SC. The SC computes $\partial_{ui} = Rep(BIO_{ui}, \theta_{ui})$, $r_{ui} = W_{ui} \oplus H_1(\partial_{ui})$, $PID_{ui} = H_2(ID_{ui}, r_{ui})$, $PWD_{ui} = H_3(PW_{ui}, PID_{ui}, r_{ui})$, $PBIO_{ui} = H_2(\partial_{ui}, r_{ui})$, $d_{ui} = G_{ui} \oplus PWD_{ui}$, $R_{nsj} = e(d_{ui}, H_{00}(ID_{nsj})) \oplus PWD_{ui} \oplus PBIO_{ui}$. Then, the SC inserts R_{nsj} into T_{ui}.

3.8 Dynamic Revocation Phase

In practice, the importance of an efficient revocation mechanism is self-evident. It has positive meaning both in preventing malicious users and improving the efficiency of RC management. In this phase, we introduce two kinds of dynamic revocation supported by this scheme, namely, the revocation of malicious users and the user initiative to apply for revocation.

3.8.1 Malicious User Revocation

During the session, if servers find that a user is visiting illegally, the server reports the RC. RC will verify the authenticity. If the situation is true, he/she immediately stop updating the user time key. Otherwise, RC will punish the service to a certain degree of downgrade.

3.8.2 User Apply For Revocation

The user sends the revocation request with identity ID_{ui} through the secure channel. After receiving the message, RC checks the identity ID_{ui} and then stops updating the corresponding user time key.

4 Security Analysis

In this section, we analyze the security of the proposed protocol in the random oracle model. In this model, each party involved in a session is treated as an oracle, and an adversary can access the oracles by issuing some specified queries.

4.1 Security Model

In this subsection, we define the capabilities of an adversary based on the literatures [28], [38], [39]. Assume A is a probabilistic polynomial time adversary. We allow adversary A to gain the capabilities through a series of random oracles controlled simulator S. Specific query can refer to the following. Let $\alpha \in \{PID_{ui}, ID_{sj}\}$ and $\top_\alpha^m$ be the mth instance of α.

- $H_i(m_i)$. When an adversary A makes H_i query with a message m_i, the oracle returns r if the query has been asked before. Otherwise, H_i oracle random generates a r, records (m_i, r) into list $L_i(i = 0, \ldots, 7)$, and returns r to A, where L_i is initially empty.
- *Extract*(α). With this query, the corresponding participant's private key can be obtained.
- *Tupdate*(PID_{ui}, t). Through this query, the adversary A can get the update time key of corresponding user.
- *Send*($\top_\alpha^m$, M). In this query, the adversary A can send a message M to the oracle $\top_\alpha^m$. Upon receiving M, according the proposed protocol the oracle $\top_\alpha^m$ performs some computations and then replies to the adversary A.
- *Reveal*($\top_\alpha^m$). This query is executed by A to obtain a session key sk from the oracle $\top_\alpha^m$. If the oracle has accepted the session, $\top_\alpha^m$ returns the corresponding session key sk; otherwise, S outputs a null value.
- *Corrupt*(α). When A executes the query with the identity ID_α, the oracle returns the private key to A.
- *Test*($\top_\alpha^m$). During the session, the adversary A is allowed to send a Test query to the oracle $\top_\alpha^m$. On receiving a Test query, $\top_\alpha^m$ flips an unbiased coin b, and then returns the session key if $b = 1$, otherwise outputs a random string $sk \in Z_q^*$ as the session key.

Before the formal proof, some basic security definitions of the mutual authentication and key agreement protocol are given.

Definition 5 (AKA-Secure). *The Authenticated Key Agreement (AKA) advantage is defined as $Adv_p^{aka}(A) = |2Pr[Asucceeds] - 1|$, where $Pr[Asucceeds]$ is the probability that A wins the game of $Test(\top_\alpha^m)$. We say the protocol P is AKA-secure if $Adv_p^{aka}(A)$ is negligible.*

Definition 6 (MA-Secure). *The MAKA protocol P is Mutual Authentication-secure (MA-Secure) if both $Pr[C2S]$ and $Pr[S2C]$ are negligible, where $C2S$ and $S2C$ denote A breaking client-to-server authentication and server-to-client authentication events, respectively.*

Definition 7 (Partnership). *When two oracles $\top_{PID_{ui}}^m$ and $\top_{ID_{sj}}^m$ mutually authenticate each other and compute a mutual session key between them, we call them Partnership.*

4.2 Provable Security

In the following, we show that the proposed 3DRMAKA protocol can achieve AKA-Secure and MA-Secure under the above security model. Based on the BDH assumption, we show that the outside adversary cannot impersonate a client to access servers (*Lemma1*) and the adversary cannot impersonate a server to fool users information (*Lemma3*). Then, based on the definition 2 assumption, we prove the proposed scheme can resist a revoked user impersonate attack (*Lemma2*). Finally, we show that our protocol can achieve AKA-secure (*Theorem2*) under the CDH assumption.

Lemma 1. *In the random oracle model, no client outside polynomial adversary against the proposed 3DRMAKA protocol can forge legal authentication messages with a non-negligible probability.*

Proof. Suppose that the outside adversary A can forge legal authentication message with a non-negligible advantage ε. Based on [40] *Lemma1*, for a given ID, the A can own a non-negligible advantage within polynomial time to violate the $C2S$ of the proposed scheme. Given an instance $(P, x \cdot P, y \cdot P, z \cdot P)$ of BDH problem, where

$P \in G_1$ and unknown $x, y, z \in Z_q^*$. The goal of the simulation S is to solve $e(P,P)^{xyz}$. S generates the public parameters $\{q, G_1, G_2, P_{pub}, g, g_{pub}, e, P, H_{0*}, H_1, H_2, H_3, H_4, H_5, H_6, H_7\}$, where $P_{pub} = x \cdot P$, $g_{pub} = g^w$ and w is unknown number in Z_q^*. Then, S randomly chooses a user's anonymous identity PID_{uI} as challenge identity and let ID_{sJ} be his/her Partnership. S interacts with A as follows:

- *H_{0*} queries*. S maintains a list L_0 initialized empty. If $PID_{ui} \neq PID_{uI} \wedge ID_{sj} \neq ID_{sJ}$, S checks if the query has been asked in L_0. If it exists, S responses the recorded value of $a \cdot P$, which $a \in Z_q^*$; if there is no record, S randomly chooses $a \in Z_q^*$ and computes $h_0 = a \cdot P$, and then inserts (PID_{ui}, a, h_0) or (ID_{sj}, a, h_0) into L_0 and returns h_0 to A. Otherwise, when $PID_{ui} = PID_{uI}$, S returns $y \cdot P$ to A. When $ID_{sj} = ID_{sJ}$, S responses $z \cdot P$ to A.
- *H_i queries*. S maintains a series of lists L_i initialized empty. When A issues such a query along with m_i, S checks if the query has been asked in L_i. If it exists, S returns the recorded value of r_i, which $r_i \in Z_q^*$; otherwise, S randomly generates $r_i \in Z_q^*$, records (m_i, r_i) into L_i and then responses r_i to A.
- *Extract(α)*. S maintains a list L_{PUSK} initialized empty. If $\alpha \neq PID_{uI} \wedge \alpha \neq ID_{sJ}$, S checks if the query has been asked in L_{PUSK}. If it exists, S responses the recorded value. If there is no record, S chooses a number $a \in Z_q^*$, sets $H_{0*}(\alpha) \leftarrow a \cdot P$, and then computes $d_{ui} = a \cdot P_{pub}$ or $d_{sj} = a \cdot P_{pub}$. Then, S inserts (α, d_{**}) and $(\alpha, a, H_{0*}(\alpha))$ into L_{PUSK} and L_0, respectively. S returns d_{ui} or d_{sj} to A. When $\alpha = PID_{uI} \vee \alpha = ID_{sJ}$, S terminates the game.
- *Tupdate(PID_{ui}, t)*. Upon receiving this query along with (PID_{ui}, t), S responses the recorded value in L_{PUT}. If not exits, S randomly chooses two number $b_{ui}, v_{ui} \in Z_q^*$, computes $B_{ui} = g^{b_{ui}} \cdot g_{pub}^{-v_{ui}}$, sets $H_5(B_{ui}, PID_{ui}, t) \leftarrow v_{ui}$, $d_{tui} \leftarrow b_{ui}$. Then, S inserts $(PID_{ui}, B_{ui}, t, d_{tui})$ and (B_{ui}, PID_{ui}, t) into L_{PUT} and L_5 respectively, and then returns (d_{tui}, B_{ui}, t) to A.
- *Send queries*.
  1. When A issues a Send($\top^m_{PID_{ui}}$,"Start with ID_{sj}") query, S generates a random number $N_1 \in Z_q^*$, and then computes $H_1(ASK)$, $k_{ui} = g^{N_1}$, $F_{ui} = PID_{ui} \oplus H_6(k_{ui}, ID_{sj}, H_1\ (ASK))$. S checks whether L_{PUT} has the update time key corresponding to the user, and if not, S calculates (d_{tui}, B_{ui}, t) similar to Tupdate (PID_{ui}, t). Then, S returns $(F_{ui}, k_{ui}, B_{ui}, d_{tui}, t)$ to A.
  2. When A makes a Send($\top^m_{ID_{sj}}$, $(F_{ui}, k_{ui}, B_{ui}, d_{tui}, t)$) query, to achieve perfect simulation, S first verifies the time key correctness by checking $g^{d_{tui}} ? = B_{ui} \cdot g_{pub}^{H_5(B_{ui}, PID_{ui}, t)}$, where $PID_{ui} = F_{ui} \oplus H_6(k_{ui}, ID_{sj}, H_1(ASK))$. If the equation holds, S continues next steps; otherwise, rejects the request. If $ID_{sj} \neq ID_{sJ}$, S generates a number $N_2 \in Z_q^*$ and computes $k_{1t} = e(d_{sj}, H_{01}(PID_{ui}))$, $k_{sj} = g^{N_2}$, $D_{sj} = H_7(k_{1t}, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. Then, S transmits (D_{sj}, k_{sj}) to the adversary A. If $ID_{sj} = ID_{sJ} \wedge PID_{ui} \neq PID_{uI}$, S chooses a random number $N_2 \in Z_q^*$ and computes $k_{1t} = k_1 = e(d_{ui}, H_{00}(ID_{sj}))$, $k_{sj} = g^{N_2}, D_{sj} = H_7(k_{1t}, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. S sends (D_{sj}, k_{sj}) to the adversary A. If $ID_{sj} = ID_{sJ} \wedge PID_{ui} = PID_{uI}$, S makes $k_{1t} = k_{r1}$ and a random number $N_2 \in Z_q^*$, where k_{r1} is a random element in G_2, and then computes $k_{sj} = g^{N_2}$,$D_{sj} = H_7(k_{1t}, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. S sends (D_{sj}, k_{sj}) to the corresponding A.
  3. When A issues a Send $(\top^m_{PID_{ui}}, (D_{sj}, k_{sj}))$ query, if $PID_{ui} \neq PID_{uI}$, S computes $k_1 = R_{sj} \oplus PWD_{ui} \oplus PBIO_{ui}$ and then checks whether the $D_{sj} ? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$ holds. If the equation holds, S computes $k_u = k_{sj}^{N_1}$, $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Then, S sends D_{ui} to A. If $ID_{sj} \neq ID_{sJ} \wedge PID_{ui} = PID_{uI}$, S makes $k_1 = k_{1t}$ and then checks whether the $D_{sj} ? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$ holds. If the equation holds, S computes $k_u = k_{sj}^{N_1}$, $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Then, S sends D_{ui} to A. If $ID_{sj} = ID_{sJ} \wedge PID_{ui} = PID_{uI}$, S makes $k_1 = k_{r1}$ then checks whether the $D_{sj} ? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$ holds. If the equation holds, S computes $k_u = k_{sj}^{N_1}$, $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Then, S sends D_{ui} to A.
  4. When A makes a Send $(\top^m_{ID_{sj}}, D_{ui})$ query, S checks whether the $D_{ui} ? = H_7(g_{pub}, k_{ui}, k_{sj}, k_{1t}, PID_{ui}, ID_{sj})$. If the equation holds, S computes $k_s = k_{ui}^{N_2}$ and the session key $sk = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Otherwise, S terminates the session.

Let q_m, q_n, q_s and q_i be the number of clients, servers, send queries and hash queries, where $i = 0, \ldots, 7$. In the case of $ID_{sj} = ID_{sJ} \wedge PID_{ui} = PID_{uI}$ and m_1th session of $PID_{uI} \wedge m_2th$ session of ID_{sJ}, if A outputs a legal message tuple $(F_{uI}, k_{uI}, D_{uI}, B_{uI}, d_{tui}, t)$, he/she must make H_7 query. According to the responses of queries above, we have made $H_{01}(PID_{uI}) \leftarrow y \cdot P$ and $H_{00}(ID_{sJ}) \leftarrow z \cdot P$. A must compute k_1 before he/she successfully outputs the legal message. Therefore, S can traverse H_7 to address the BDH problem because A has computed the valid k_1 by $k_1 = e(y \cdot P_{pub}, z \cdot P) = e(P,P)^{xyz}$ and made H_7 query. Let E_1, E_2, E_3 and E_4 be the events of No-Extract($PID_{uI} \wedge ID_{sJ}$), both sides of the session $ID_{sj} = ID_{sJ} \wedge PID_{ui} = PID_{uI}$, m_1th session of $PID_{uI} \wedge m_2th$ session of ID_{sJ}, finding k_1 in H_7. According to the assumption, A outputs a valid message tuple with non-negligible ε, ie. $Pr[C2S] = \varepsilon$. We can get $Pr[E_1] = (1 - \frac{2}{q_m + q_n})^{(q_m + q_n)}$, $Pr[E_2|E_1] = Pr[E_2] = \frac{1}{q_m \cdot q_n}$, $Pr[E_3|E_1 \wedge E_2] = Pr[E_3] = \frac{1}{q_s^2}$, $Pr[E_4|E_1 \wedge E_2 \wedge E_3] = \frac{1}{q_7}$ and $Pr[C2S$

$|E_1 \wedge E_2 \wedge E_3 \wedge E_4] = \varepsilon$. Therefore, the probability that S solves the BDH problem is

$$\begin{aligned} &Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge C2S] \\ &= Pr[E_1] \cdot Pr[E_2|E_1] \cdot Pr[E_3|E_1 \wedge E_2] \\ &\cdot Pr[E_4|E_1 \wedge E_2 \wedge E_3] \cdot Pr[C2S|E_1 \wedge E_2 \wedge E_3 \wedge E_4] \\ &= \frac{q_m + q_n - 2}{q_m^2 \cdot q_n^2 \cdot q_s^2 \cdot q_7} \cdot \varepsilon. \end{aligned}$$

Since S is a probability polynomial adversary, q_m, q_n, q_s and q_7 are at most polynomial times so that S can violate BDH with a non-negligible probability. However, this contradicts with the difficulty of the BDH problem. Therefore, we conclude that the proposed protocol can resist an outside adversary impersonation user attack. □

Lemma 2. *In the random oracle model, no client revoked polynomial adversary against the proposed 3DRMAKA can forge a legal authentication message with a non-negligible probability.*

Proof. Suppose that the revoked adversary A can forge legal authentication messages with a non-negligible advantage ε, ie. $Pr[C2S] = \varepsilon$. The goal of the simulation S is to output valid Schnorr signatures tuple $(H_5(B_{ui}, PID_{ui}, t), d_{tui}, t)$. S generates the public parameters $\{q, G_1, G_2, P_{pub}, g, g_{pub}, e, P, H_{0*}, H_1, H_2, H_3, H_4, H_5, H_6, H_7\}$, where $P_{pub} = x \cdot P$, $g_{pub} = g^y$ and x, y are unknown random numbers in Z_q^*. S randomly chooses a user's anonymous identity PID_{uI} as challenge identity. S interacts with A as follows:

- *H_{0*} queries*. S maintains a list L_0 initialized empty. When A issues such a query with α, S checks if the query has been recorded in L_0. If it exists, S responses the recorded value of $a \cdot P$, which $a \in Z_q^*$; if there is no record, S randomly chooses $a \in Z_q^*$ and computes $h_0 = a \cdot P$, and then inserts (PID_{ui}, a, h_0) into L_0 and returns h_0 to A.
- *H_i queries*. S maintains a series of lists L_i initialized empty. When A issues such a query along with m_i, S checks if the query has been asked in L_i. If it exists, S returns the recorded value of r_i, which $r_i \in Z_q^*$; otherwise, S randomly generates $r_i \in Z_q^*$, records (m_i, r_i) into L_i and then responses r_i to A.
- *Extract(α)*. S maintains a list L_{PUSK} initialized empty. When A makes such a query with α, S responses the recorded value in L_{PUSK}. If there is no record, S chooses a number $a \in Z_q^*$, sets $H_{0*}(\alpha) \leftarrow a \cdot P$, and then computes $d_{ui} = a \cdot P_{pub}$ or $d_{sj} = a \cdot P_{pub}$. Then S inserts (α, d_{**}) and $(\alpha, a, H_{0*}(\alpha))$ into L_{PUSK} and L_0. Finally, S returns d_{ui} or d_{sj} to A.
- *Tupdate (PID_{ui}, t)*. Upon receiving this query along with (PID_{ui}, t), if $PID_{ui} \neq PID_{uI}$, S responses the recorded value in L_{PUT}. If not exits, S randomly chooses two numbers $b_{ui}, v_{ui} \in Z_q^*$, computes $B_{ui} = g^{b_{ui}} \cdot g_{pub}^{-v_{ui}}$, sets $H_5(B_{ui}, PID_{ui}, t) \leftarrow v_{ui}$, $d_{tui} \leftarrow b_{ui}$. Then, S inserts $(PID_{ui}, B_{ui}, t, d_{tui})$ and (B_{ui}, PID_{ui}, t) into L_{PUT} and L_5 respectively, and returns (d_{tui}, B_{ui}, t) to A. When $PID_{ui} = PID_{uI}$, S stops the game.
- *Send queries*. When A issues Send($\top_{PID_{ui}}^m$,"start with ID_{sj}") query, if $PID_{ui} \neq PID_{uI}$, S responses according to the specification of the proposed protocol; otherwise, S terminates the session. For other queries, S response A according to the proposed protocol because S knows α's private key.

Let q_m and q_s be the number of clients and send queries. In the case of $PID_{ui} = PID_{uI}$ and mth session, if a revoked adversary A outputs a legal update time key tuple (B_{uI}, d_{tuI}, t), S can violate Schnorr signatures with a non-negligible probability. Because the $g^{b_{tuI}} = B_{uI} \cdot g_{pub}^{H_5(B_{uI}, PID_{uI}, t)} = g^{b_{uI} + H_5(B_{uI}, PID_{uI}, t) \cdot y}$ equation is satisfied, S outputs $(d_{tuI}, H_5(B_{uI}, PID_{uI}, t), t)$ as Schnorr signatures tuple, where $H_5(B_{uI}, PID_{uI}, t)$ can find in L_5 with $\frac{1}{q_5}$ probability. Let E_1, E_2, E_3, E_4 be the events of $PID_{ui} = PID_{uI}$, mth session, successful forged update time key and finding $H_5(B_{uI}, PID_{uI}, t)$ in L_5. According to the assumption, A outputs a valid time key message tuple with non-negligible ε, ie. $Pr[E_3] = Pr[C2S] = \varepsilon$. We can get $Pr[E_1] = \frac{1}{q_m}$, $Pr[E_2|E_1] = \frac{1}{q_s}$, and $Pr[E_3|E_1 \wedge E_2] = \varepsilon$, and $Pr[E_4|E_1 \wedge E_2 \wedge E_3] = \frac{1}{q_5}$. Therefore, the probability that S can forge the Schnorr signatures is

$$\begin{aligned} &Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \\ &= Pr[E_1] \cdot Pr[E_2|E_1] \\ &\cdot Pr[E_3|E_1 \wedge E_2] \cdot Pr[E_4|E_1 \wedge E_2 \wedge E_3] \\ &= \frac{\varepsilon}{q_s \cdot q_m \cdot q_5}. \end{aligned}$$

Since S is a probability polynomial adversary, q_m, q_s and q_5 are at most polynomial times so that S violates Schnorr signatures with a non-negligible probability. But this contradicts with the difficulty of the Schnorr signatures unforgeability assumption. Therefore, we conclude that the proposed protocol can resist a revoked adversary impersonation user attack. □

Lemma 3. *In the random oracle model, no server polynomial adversary against the proposed 3DRMAKA can forge a legal authentication message with a non-negligible probability.*

Proof. The simulation is the same for outside client attacker and server attacker in the proposed protocol. They also are all based on the BDH assumption. Therefore, according to the *Lemma1* we conclude that the proposed protocol can resist adversary impersonation server attack. □

Theorem 1. *Based on* Lemma1, Lemma2 *and* Lemma3, *the proposed 3DRMAKA protocol for multi-server environments is MA-secure.*

Theorem 2. *Based on CDH assumption, the proposed 3DRMAKA scheme for multi-server environments is AKA-secure, namely $Adv_p^{aka}(A)$ probability is negligible.*

Proof. Assume an adversary A correctly output the bit b which is chosen by S in the Test query with a non-negligible probability ε, then there must be a polynomial time adversary S that can solve the CDH problem with a non-negligible probability.Similar to the *Lemma1*, given an instance of the CDH problem $\{g, g_1 = g^x, g_2 = g^y\}$, where $g \in G_2$ and unknown $x, y \in Z_q^*$. The goal of the simulation S is to compute g^{xy}. S simulates the system initializing algorithm to

generate the system parameters $\{q, G_1, G_2, P_{pub}, g, g_{pub}, e, P, H_{0*}, H_1, H_2, H_3, H_4, H_5, H_6 \ , H_7\}$, where $P_{pub} = z \cdot P$, $g_{pub} = g^w$ and z, w are unknown numbers in Z_q^*. Let q_i, q_m, q_n and q_s be the number of hash queries, clients, servers and send queries, where $i = 0, \ldots, 7$. S chooses $I \in \{1, 2, \ldots, q_m\}$, $J \in \{1, 2, \ldots, q_n\}$ and $m_1, m_2 \in \{1, 2, \ldots, q_s\}$ as challenge identities, assuming that A can violate the key agreement against the oracle $\top_{PID_{uI}}^{m_1}$ with his/her partnership $\top_{ID_{sJ}}^{m_2}$. S interacts with A as follows.

- *H_{0*} queries*. S maintains a list L_0 initialized empty. When A makes such a query with PID_{ui} or ID_{sj}, S checks if the query has been asked in L_0. If it exists, S responses the recorded value of $a \cdot P$, which $a \in Z_q^*$; if there is no record, S randomly chooses $a \in Z_q^*$ and computes $h_0 = a \cdot P$, and then inserts (PID_{ui}, a, h_0) or (ID_{sj}, a, h_0) into L_0 and returns h_0 to A.
- *H_i queries*. S maintains a series of lists L_i initialized empty. When A issues such a query along with m_i, S checks if the query has been recorded in L_i. If it exists, S returns the recorded value of r_i, which $r_i \in Z_q^*$; otherwise, S randomly generates $r_i \in Z_q^*$, records (m_i, r_i) into L_i and then responses r_i to A.
- *Extract (α)*. S maintains a list L_{PUSK} initialized empty. When A makes such a query with α, S checks if the query has been asked in L_{PUSK}. If it exists, S responses the recorded value. If there is no record, S chooses a number $a \in Z_q^*$, sets $H_{0*}(\alpha) \leftarrow a \cdot P$, and then computes $d_{ui} = a \cdot P_{pub}$ or $d_{sj} = a \cdot P_{pub}$. Then, S inserts (α, d_{**}) and $(\alpha, a, H_{0*}(\alpha))$ into L_{PUSK} and L_0, respectively.
- *Tupdate(PID_{ui}, t)*. Upon receiving this query along with (PID_{ui}, t), S responses the recorded value in L_{PUT}. If not exits, S randomly chooses two numbers $b_{ui}, v_{ui} \in Z_q^*$, computes $B_{ui} = g^{b_{ui}} \cdot g_{pub}^{-v_{ui}}$, sets $H_5(B_{ui}, PID_{ui}, t) \leftarrow v_{ui}$ and $d_{tui} \leftarrow b_{ui}$. Then, S inserts $(PID_{ui}, B_{ui}, t, d_{tui})$ and (B_{ui}, PID_{ui}, t) into L_{PUT} and L_5 respectively, and then S returns (d_{tui}, B_{ui}, t) to A.
- *Send queries*.
  1. When A issues a Send($\top_{PID_{ui}}^m$, "start with ID_{sj}'') query, if $PID_{ui} \neq PID_{uI}$, S responses according to the proposed. If $PID_{ui} = PID_{uI}$, S makes $k_{ui} = g_1$ and other responses are based on the protocol.
  2. When A makes a Send($\top_{ID_{sj}}^m$, (F_{ui}, k_{ui}, B_{ui}, d_{tui}, t)) query, S first verifies the time key correctness by checking $g^{b_{tui}} ? = B_{ui} \cdot g_{pub}^{H_5(B_{ui}, PID_{ui}, t)}$, where $PID_{ui} = F_{ui} \oplus H_6(k_{ui}, ID_{sj}, H_1(ASK))$. If the equation holds, S continues next steps; otherwise, rejects the request. If $ID_{sj} = ID_{sJ}$, S sets $k_{sj} \leftarrow g_2$ and computes $k_{1t} = e(d_{sj}, H_{01}(PID_{ui}))$, $D_{sj} = H_7(k_{1t}, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. Then, S response (D_{sj}, k_{sj}) to A. Other cases respond according to the proposed scheme.
  3. When A issues a Send ($\top_{PID_{ui}}^m$, (D_{sj}, k_{sj})) query, if $PID_{ui} \neq PID_{uI}$, S responses A according the proposed protocol. If $PID_{ui} = PID_{uI} \wedge ID_{sj} \neq ID_{sJ}$, S computes $k_1 = R_{sj} \oplus PWD_{ui} \oplus PBIO_{ui}$ and then checks whether the $D_{sj} ? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$. If the equation holds, S computes $k_u = k_{ui}^{N_2}$, $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, the session key $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Then, S sends D_{ui} to A. If $PID_{ui} = PID_{uI} \wedge ID_{sj} = ID_{sJ}$, S computes $k_1 = R_{sj} \oplus PWD_{ui} \oplus PBIO_{ui}$ and checks whether the $D_{sj} ? = H_7(k_1, k_{sj}, k_{ui}, g_{pub}, PID_{ui}, ID_{sj})$ holds. If the equation holds, S chooses a random element k in G_2 and sets $k_u \leftarrow k$. Then, S computes $D_{ui} = H_7(g_{pub}, k_{ui}, k_{sj}, k_1, PID_{ui}, ID_{sj})$, the session key $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. S sends D_{ui} to A.
  4. When A makes a Send ($\top_{ID_{sj}}^m$, D_{ui}) query, S checks whether the $D_{ui} \ ?= H_7(g_{pub}, k_{ui}, k_{sj}, k_{1t}, PID_{ui}, ID_{sj})$. If the equation holds and $ID_{sj} \neq ID_{sJ}$, S computes $k_s = k_{ui}^{N_2}$ and the session key $sk = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. If $PID_{ui} \neq PID_{uI} \wedge ID_{sj} = ID_{sJ}$, S computes $k_s = k_{sj}^{N_1}$ and the session key $sk = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. If $PID_{ui} = PID_{uI} \wedge ID_{sj} = ID_{sJ}$, S sets $k_s \leftarrow k$ and computes the session key $sk = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Otherwise, S refuses the conversation.
- *Corrupt(α)*. when A issues such a query, S looks up L_{PUSK} for a tuple (α, d_{**}) and replies d_{**} to A.
- *Reveal($\top_\alpha^m$)*. If the response of the oracle$\top_\alpha^m$ has accepted the session, S returns the corresponding session key sk; otherwise, it outputs a null value.
- *Test-query*. If the query is not between $\top_{PID_{uI}}^{m_1}$ and $\top_{ID_{sj}}^{m_2}$, then S cancels the game; otherwise S flips an unbiased coin b and returns the session key if $b = 1$, otherwise outputs a random string $sk \in Z_q^*$ as the session key.

In the case of $ID_{sj} = ID_{sJ} \wedge PID_{ui} = PID_{uI}$ and m_1th session of $PID_{uI} \wedge$ m_2th session of ID_{sJ}, if A successfully guesses b with a non-negligible ε, we will show that S can solve CDH problem. According to the responses of queries above, we have set $k_{ui} = g_1$, $k_{sj} = g_2$. In the following we analyze the probability that A computes the correct k_u/k_s. Let E_{sk}, E_{TU} and E_{TS} be the events of obtaining the correct session key, successfully Test-query to the $\top_{PID_{uI}}^{m_1}$ and the $\top_{ID_{sj}}^{m_2}$. According to the assumption, we can get $Pr[E_{sk}] \geq \frac{\varepsilon}{2}$, ie.

$$Pr[E_{sk} \wedge E_{TS} \wedge C2S] + Pr[E_{sk} \wedge E_{TS} \wedge \neg C2S] + Pr[E_{sk} \wedge E_{TU}] \geq \frac{\varepsilon}{2}$$

Then, we have

$$Pr[E_{sk} \wedge E_{TS} \wedge \neg C2S] + Pr[E_{sk} \wedge E_{TU}] \geq \frac{\varepsilon}{2} - Pr[C2S].$$

In addition, the two events $(E_{TS} \wedge \neg C2S)$ and E_{TU} are equivalent thought observing the above simulation, so we can get $Pr[E_{sk} \wedge E_{TU}] \geq \frac{\varepsilon}{4} - \frac{Pr[C2S]}{2}$. Therefore, S can find k_u to solve the CDH problem with a probability

$$\frac{\varepsilon - 2Pr[C2S]}{4 \cdot q_s^2 \cdot q_m \cdot q_n \cdot q_7}.$$

According to the proof of *Lemma1*, we know that $Pr[C2S]$ is negligible. So, the probability is non-negligible. In view of the adversary A computes k_u/k_s before he/she successfully outputs b. Therefore, S traverses H_7 to solve the CDH problem. Thus S violates CDH with a non-negligible probability. However, this contradicts with the difficulty of the CDH problem. Therefore, we conclude that the proposed protocol can achieve AKA-secure, namely, $Adv_p^{aka}(A)$ probability is negligible. □

4.3 Analysis of Security Requirements

In this subsection, we analyze that the proposed 3DRMAKA protocol is able to resist various possible known attacks resulting from the multi-server environments.

Three-Token Security. In 3DRMAKA protocol, the three factors in the authentication process are indispensable. There is no way for the adversary to bypass any factor to complete the login or authentication process. Because only biological information can be used to compute r_{ui} and there is no way to complete the smart card login without the correct identity ID_{ui} and password PW_{ui}. Even if the adversary bypasses the smart card login through complex physical attacks, he/she can't compute $H_1(ASK)$ and k_1 to complete anonymous message delivery and authentication because he/she doesn't have PID_{ui}, PWD_{ui} and $PBIO_{ui}$. Thus, the proposed scheme can achieve three-token security.

Stolen Card Attack. Assume the adversary steals the user's smart card. If it is not a registered attacker, he fails to acquire any valid credentials using extracted information from smart card. Through we carefully design, any credentials in the smart card will not be obtained by an attacker using xor operation, because any xor operation will introduce a new unknown element, such as $W_{ui} \oplus h_1 = H_1(\partial_{ui}) \oplus H_1(ASK) \oplus PBIO_{ui}$. Even a malicious server with PID_{ui}, $H_1(ASK)$ and $e(d_{sj}, H_{01}(PID_{ui}))$, where $e(d_{sj}, H_{01}(PID_{ui})) = e(d_{ui}, H_{00}(ID_{sj}))$, would not be able to recover any useful information, such as $R_{sj} \oplus h_1 \oplus H_1(ASK) \oplus e(d_{sj}, H_{01}(PID_{ui})) = r_{ui} \oplus PWD_{ui}$, $r_{ui} \oplus PWD_{ui} \oplus G_{ui} = r_{ui} \oplus d_{ui}$. For malicious users only $H_1(ASK)$ can be used on stolen smart card attacks. Hence, the proposed protocol is secure against the stolen card attack.

Password Guessing Attack. As mentioned earlier, based on two-factor authentication schemes suffer from password guessing attack. Like He et al.'s protocol [28] is based on the user's identity for the attacker does not know the premise to protect their scheme from password guessing attack, but this assumption is vulnerable in the real world. In our 3DRMAKA scheme, the user's biometrics is used to resist the password guessing attack on smart card. Without the correct biometric information BIO_{ui} used to recover r_{ui}, the attack can't compute correctly PWD_{ui} unless a collision is found for the secure hash function. As a result, our scheme ia able to resist password guessing attack.

Perfect Forward Security. Suppose that the adversary obtains the private keys d_{ui} and d_{sj} of the user and the corresponding server through various uncontrollable means such as social engineering attack. In addition, previous interaction information $(F_{ui}, k_{ui}, D_{ui}, B_{ui}, d_{tui}, t, D_{sj}, k_{sj})$ of user U_i and server S_j is intercepted by the passive monitoring attack. The adversary can compute k_1, k_{1t} using d_{ui} and d_{sj}. However, the adversary must solve the CDH problem before calculating the correct session key $H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$. Because k_{ui}, k_{sj} and k_s are just a CDH problem tuple. Therefore, our scheme can achieve perfect forward security.

Anonymity and Un-Traceability. To achieve anonymity, the proposed protocol uses a double anonymity protection. First, users register directly with an anonymous identity in our scheme. Second, we use the hash of the shared key to add a second layer of protection to the user's anonymous identity against unregistered attackers. In order to prevent attackers from associating specific interaction messages to the corresponding users, we use random number N_1 to calculate F_{ui} for breaking message uniformity. Thus, our scheme can provide the anonymity and un-traceability.

Dynamic Revocation. Unlike previous schemes [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] in which there is not dynamic revocation, the proposed protocol gives RC greater flexibility in managing users. RC can set a reasonable time t and then let it be applied to a variety of different scenarios. For example, the time t can be set shorter in some situations where the security is sensitive; the time rewards can be appropriately given when the time key is next issued according to the user's honesty. Therefore, our scheme can provide flexible dynamic revocation.

Mutual Authentication and Session Key Agreement. Based on BDH problem and Schnorr signatures unforgeability assumption, we have proved in *Lemma 1, 2, 3* that both parties involved in the session can secure authenticate for each other. On the other hand, according to the proposed 3DRMAKA protocol for multi-server architectures, both two participants compute the session $sk = H_7(k_u, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj}) = H_7(k_s, g_{pub}, k_{sj}, k_{ui}, PID_{ui}, ID_{sj})$, which can be used in the future secure communications. Thus, the proposed protocol can provide mutual authentication and session key agreement.

Privileged Insider Attack. In the user registration phase of 3DRMAKA scheme, a legal user U_i transmits PID_{ui}, $PBIO_{ui}, ID_{ui}$ and the pseudo-password $PWD_{ui} = H_3(PW_{ui}, PID_{ui}, r_{ui})$ instead of sending the plain text password PW_{ui}, where r_{ui} is a random number in Z_q^*. The privileged insider of RC cannot get PW_{ui} from $H_3(PW_{ui}, PID_{ui}, r_{ui})$ since he/she doesn't know r_{ui} and reverse the one-way hash function. Therefore, our scheme has the capability to resist the privileged insider attack.

Replay and Man-in-the-Middle Attacks. Each time a new session is opened in our protocol, each participant chooses a random number N_1, N_2 respectively, and they will be used to send and respond to the request. Due to the freshness of g^{N_1} and g^{N_2}, the user and the server can find the replay of messages by validating the legitimacy of the received message. In man-in-the-middle attack, an attacker may try to impersonate a valid user U_i or server S_j by intercepting the messages. However, in our scheme, either the verification of the updated time key or the calculation of k_1, k_{1t} of the user or server is associated with the corresponding identity information. Based on BDH assumption, the adversary

TABLE 1
Security Comparisons

	Odelu et al.' protocol	Liao et al.' protocol	Reddy et al.' protocol	He et al.' protocol	Our protocol
SR-1	Yes	Yes	Yes	Yes	Yes
SR-2	Yes	Yes	No	Yes	Yes
SR-3	Yes	No	Yes	No robust	Yes
SR-4	Yes	–	Yes	Yes	Yes
SR-5	Yes	No	Yes	Yes	Yes
SR-6	No	No	No	Yes	Yes
SR-7	No	No	No	No	Yes
SR-8	Yes	No robust	No	No robust	Yes
SR-9	Yes	–	No	No robust	Yes
SR-10	Yes	Yes	Yes	Yes	Yes
SR-11	Yes	Yes	Yes	Yes	Yes
SR-12	Yes	Yes	No	Yes	Yes
SR-13	No	Yes	Yes	Yes	Yes
SR-14	Yes	No	Yes	Yes	Yes

cannot modify the identity information inside k_1, k_{1t}. As a result, our scheme can avoid replay and man-in-the-middle attacks.

4.4 Security Comparisons and Cryptanalysis

In this subsection, 3DRMAKA scheme is compared with the related protocols of He et al. [28], Liao et al. [10], Reddy et al. [23], Odelu et al. [22]. let $SR-1$, $SR-2$, $SR-3$, $SR-4$, $SR-5$, $SR-6$, $SR-7$, $SR-8$, $SR-9$, $SR-10$, $SR-11$, $SR-12$, $SR-13$ and $SR-14$ denote three/two-token security, resistance of stolen card attack, security of password guessing attack, service of perfect forward security, provided anonymity, provided formal proof, achieved dynamic revocation, achieved mutual authentication, provided session key agreement, resistance of privileged insider attack, no replay attack, no man-in-the-middle attack, no requiring RC in session and achieved un-traceability. As shown in Table 1, our scheme has better security. Note that "–" means there is no corresponding security requirement.

In the rest of this subsection, we will analyze the weaknesses of He et al. protocol and the security vulnerabilities of Reddy et al. protocol. To control the length of the paper, their papers [23], [28] will not be reintroduced in detail. However, to facilitate the corresponding, we will keep the original notation unchanged.

4.4.1 Analysis of He et al.'s Protocol

In their paper, He et al. used the user's identity ID_{U_i} to resist off-line password attack. Although their protocol is anonymous, attackers can exploit social-engineering attack to obtain ID_{U_i}. For some specific attacks, using social engineering to get relatively open identity information is still very easy. Given the commonality of user identity information, it obviously can't be an important security factor in security protocols. However, an attacker who successfully acquired the user identity corresponding to the smart card would launch an off-line guessing password attack in their scheme. In view of the rapid development of CPU/GPU and the availability of password dictionaries on the Internet, the cost of an attacker initiating the effective password guessing

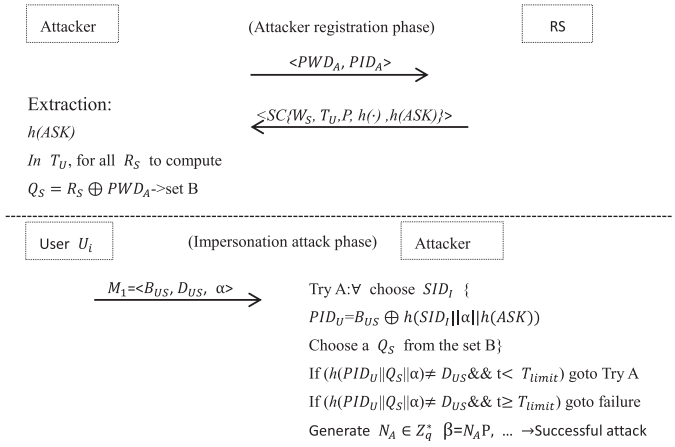


Fig. 7. Server impersonation attack.

attack is significantly reduced. In their scheme, once an attacker gains the user's PW_{U_i}, he can use PW_{U_i}, ID_{U_i} and b_{U_i} to compute the private key information $\tau_{U_i} = \psi_{U_i} \oplus h_0(ID_{U_i}, PW_{U_i}, b_{U_i})$. Now the attacker has completely broken the smart card, he can use PW_{U_i}, ID_{U_i} to complete the smart card login and then use τ_{U_i} to impersonate the user to complete and any server authentication and session key agreement. To a certain extent, this is also a common weakness of such authentication and key agreement protocols that only use password. The use of biometrics can solve the exhaustive attack of password.

4.4.2 Cryptanalysis of Reddy et al.'s Protocol

Vulnerable to Server Impersonation Attack. In Reddy et al.'s protocol, since the RC does not require any identity while users are enrolled, an attacker can get a smart card by submitting an anonymous identity requirement directly. The attacker extracted $h(ASK)$ and T_U from the smart card. Then, he recovers all Q_s from T_U, that is, he computes $Q_s = R_s \oplus PWD_A$, where PWD_A is a pseudo-password of the attacker. As shown in Fig. 7, We give the pseudo-code that the attacker can impersonate to be the server to communicate with users. Note T_{limit} is the time limit of the network response. The adversary can make an effective attack by correctly choosing Q_S from the set B and guessing SID_I. This attack is feasible because the adversary has recovered all Q_S and used them to generate the set B in the first phase. And the identities of the servers are public and SID_I that are often accessed by users are controllable. The adversary can also get servers identity information from T_U in smart card. The adversary can make multiple attack attempts within the effective time T_{limit}.

Vulnerable to Man-in-the-Middle Attack. First of all, the adversary registers to get smart card. He extracts T_U and $h(ASK)$ to generate set the B as shown in the impersonation server attack. Detailed middle-man attack algorithm shows in Fig. 8. The attacker first uses the information extracted from the smart card to complete the server's impersonation attack as shown in Fig. 7. At the same time he used his successful guess of the information SID_I, PID_U and Q_S impersonation the user to complete the authentication and key agreement with the real server S_j. After that, the attacker can negotiate session key with the server and the user respectively to see all communications between U_i and S_j.

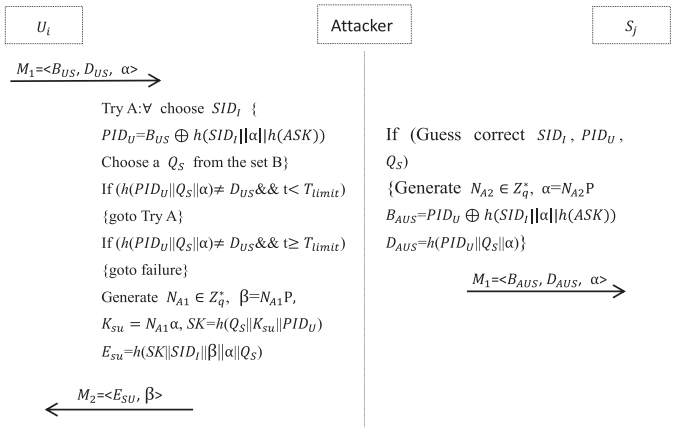


Fig. 8. Man-in-the-middle attack.

Through the above proof, comparison and analysis, the proposed 3DRMAKA protocol can have all security properties requirement from multi-server environments. Therefore, the proposed protocol has better security than the relevant comparison protocols.

5 Performance Analysis

In this section, we will analyze the performance of the proposed 3DRMAKA protocol and the related comparison schemes [10], [22], [23], [28] in terms of computation time, communication costs and the required number of round trip times (RTT). Depending on the network delay the RTT time can become the dominant cost for a protocol. A more extensive comparison can be obtained from paper [41]. To achieve a reliable security level of 1024-bits RSA algorithm, we choose a Tate pairing and super-singular curve $y^2 = x^3 - 3x\,modp$ over F_p which F_p is 512 bits finite field. And then we choose a subgroup of G_1 with order $q = 2^{159} + 2^{17} + 1$ that is generated from points on elliptic curve over a finite field F_p. First, we define the following notations.

- T_{map}: Time to execute a bilinear-pairing operation.
- T_{mtp}: Time to execute a map-to-point hash operation.
- T_{exp}: Time to execute a modular exponentiation operation.
- T_{pa}: Time to execute a point addition operation.
- T_h: Time to execute a general hash operation.
- T_{mul}: Time to execute a multiplication operation in G_2.
- T_{pmul}: Time to execute a scalar multiplication operation in G_1.

TABLE 2
The Running Time of Related Operations (Millisecond)

	T_{map}	T_{mtp}	T_{exp}	T_{pa}	T_h	T_{mul}	T_{pmul}	T_{sed}
User	32.55	30.40	3.05	0.10	0.225	0.05	11.85	0.03
Server	5.02	5.18	0.53	0.02	0.015	0.003	2.04	0.02

- T_{sed}: Time to execute a symmetric key encryption/decryption algorithm.

Based on the Miracl library, we tested the timing of the above operations on the computer side and mobile device side, respectively. The specific test environment is as follows.

- *PC*: The test code runs on the virtual machine Ubuntu 14.04.3 LTS. The physical machine is equipped with a Core 3G I7 processor (2.4 GHz), 8G DDR3 memory and Windows 7 Ultimate, 64-bit 6.1.7601, Service Pack 1 operating system.
- *Mobile*: The operating system is based on Android 6.0.1 MIUI 9 8.3.29 development version, the CPU is Qualcomm Snapdragon 801 (2.5GHz) processor, 2G running memory.
- *Connection And Compiler*: The connection tool is the Android Debug Bridge toolkit. The compiler environment for the PC and mobile devices is gcc version 4.8.4 and arm-none-linux-gnueabi cross-compilation tool, respectively.

The detailed experimental data is shown in Table 2.

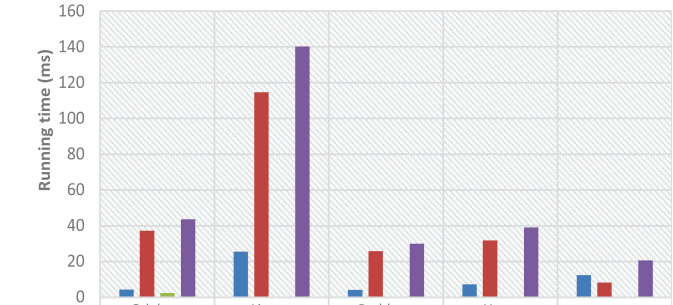
5.1 Computation Cost

For the sake of comparison, we summarize Table 3 and Fig. 9 in terms of the number of operations and the time of execution to complete a mutual authentication and key agreement in our scheme and He et al. [28], Liao et al. [10], Reddy et al. [23] and Odelu et al. [22] schemes, respectively. Assume that RC and server have the same computing power.

As Fig. 9 shows, our protocol has significant advantages in terms of client computing time and total cost time. This allows our protocol to be deployed on smart devices that have limited computing power. Improve the universality of the protocol. On the other hand, the computational cost of our protocol on the server side is slightly higher than that of the corresponding comparison schemes [23], [28], but our scheme achieves higher security and more comprehensive functionality, so it brings a certain server computing time ascension. In Odelu et al. protocol, RC needs to assist the server to complete each authentication and key agreement, which undoubtedly will make RC bear the stress of

TABLE 3
The Number of Operation

	Odelu et al.' protocol	Liao et al.' protocol	Reddy et al.' protocol	He et al.' protocol	The proposed protocol
User	$3T_{pmul} + 7T_h + T_{sed}$	$T_{mtp} + 7T_{pmul} + T_{pa} + 5T_h$	$9T_h + 2T_{pmul}$	$2T_{pmul} + T_{ap} + 2T_{exp} + 8T_h$	$9T_h + 2T_{exp}$
Server	$2T_{pmul} + 6T_h + 2T_{sed}$	$T_{mtp} + 5T_{pmul} + T_{pa} + 4T_h + 2T_{map}$	$6T_h + 2T_{pmul}$	$T_{map} + 4T_{exp} + 2T_{mul} + 5T_h$	$T_{map} + T_{mtp} + T_{mul} + 4T_{exp} + 5T_h$
RC	$T_{pmul} + 11T_h + 3T_{sed}$	Not Required	Not Required	Not Required	Not Required



	Odelu et al.'protocol	Liao et al.'protocol	Reddy et al.'protocol	He et al.'protocol	Our protocol
Server time	4.21	25.5	4.17	7.221	12.393
User time	37.115	114.575	25.725	31.7	8.125
RC time	2.265	0	0	0	0
Total time	43.59	140.075	29.895	38.921	20.518

Fig. 9. Computation cost comparisons.

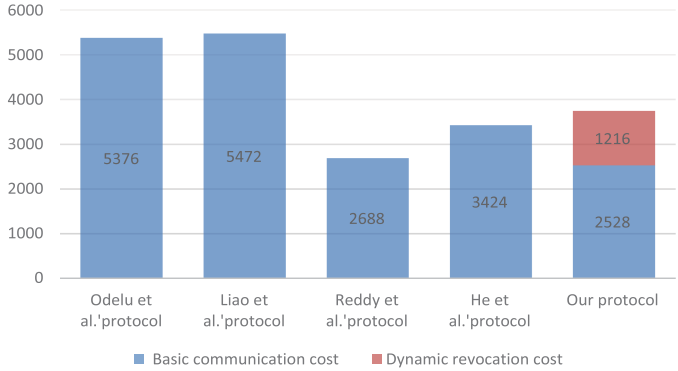


Fig. 10. Communication cost comparisons.

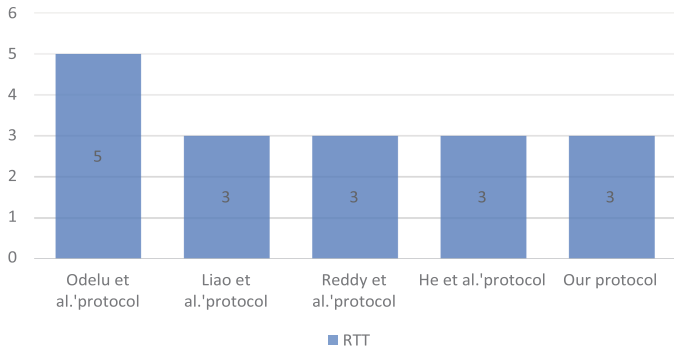


Fig. 11. The required number of round trip times.

the entire network from various server requests. Combined with the above analysis, the proposed protocol has advantages in terms of computation cost.

5.2 Communication and RTT Cost

Based on the previous discussion of security level, we have assumed that the length of p and q are 512 bits and 160 bits respectively. Thus, the size of an element in G_1 or G_2 and the length of hash functions H_i $(i = 1, \ldots, 7)$ output are 1024 bits and 160 bits respectively. Assume the length of both user's identity and valid time t are 32 bits. In a mutual authentication and key agreement, the communication cost is computed as follows.

In Odelu et al.'protocol, all communications are $\{C_1, X, h_1, C_1, X, h_1, C_2, h_2, C_3, h_3, Y, h_4, h_5\}$. C_1 should be 512 bits because $C_1 = E_{K_{1x}}[\ldots]$ that K_{1x} represents the x-coordinate a point of the ECC. $\{X, Y\}$ are also points of the ECC and others are the output of the 160-bit hash function. Therefore, the total communication cost of Odelu et al.' protocol is $512 \times 2 + 1024 \times 2 + 1024 + 8 \times 160 = 5376$ bits.

In Reddy et al.' protocol, all communications are $\{B_{US}, D_{US}, \alpha, E_{SU}, \beta, F_{US}\}$, where $\{\alpha, \beta\}$ are two point of ECC and others are the output of the 160 bits. Therefore, the total communication cost of Reddy et al.'protocol is $1024 \times 2 + 4 \times 160 = 2688$ bits.

In Liao et al.'protocol, all communications are $\{ID_{U_i}, M_i, B_{ij}, R_i, Auth_{ij}, Auth_{ji}, K_{ji}, R_j\}$, where $\{M_i, B_{ij}, R_i, K_{ji}, R_j\}$ are points of ECC , ID_{U_i} is the identity and others are the output of the 160-bit hash function. Therefore, the total communication cost of Liao et al.' protocol is $1024 \times 5 + 2 \times 160 + 32 = 5472$ bits.

In He et al.'protocol, all communications are $\{R_{U_i}, C_{U_i}, y, \alpha_{S_j}\}$, where $C_{U_i} = h_6(x) \oplus (ID_{U_i}, g_{U_i}, a_{U_i})$, ID_{U_i} is the identity, $R_{U_i} \in G_1$, $\{y, g_{U_i}\} \in G_2$ and $\{\alpha_{S_j}, a_{U_i}\} \in Z_q^*$. Therefore, the total communication cost of He et al.' protocol is $1024 \times 3 + 2 \times 160 + 32 = 3424$ bits.

In our protocol, all communications are $\{F_{ui}, k_{ui}, D_{ui}, B_{ui}, d_{tui}, t, D_{sj}, k_{sj}\}$, where $\{k_{ui}, B_{ui}, k_{sj}\} \in G_2$, t is 32 bits valid time and others are the output of the 160 bits. Therefore, the total communication cost of the proposed protocol is $1024 \times 3 + 4 \times 160 + 32 = 3744$ bits. For convenience, we still summarize the communication cost of the relevant protocols in Fig. 10.

The proposed protocol over Odelu et al.'protocol and Liao et al.'protocol has obvious advantage in terms of communication cost. On the other hand, our protocol adds $\{B_{ui}, d_{tui}, t\}$ to achieve dynamic revocation, which brings 1216 bits of communication cost. Our scheme cost is minimal in basic authentication and key agreement communications.

The required number of round trip times in the protocol is also an important factor affecting the efficiency of the implementation. Fewer the number of communications tend to be more efficient, especially in complex network environments. So in Fig. 11, we summarize the number of communications required for each comparison protocol and our protocol.

In Odelu et al.' protocol, since each authentication requires RC participation, it needs two additional interactions than other protocols. In complex network environments, multiple interactions result in a significant increase in communication time. Through a comprehensive comparison of execution efficiencies, we conclude that our protocol has advantages over the relevant protocols in terms of computation time and communication cost.

To prove that our protocol is technically sound, our programming simulation implements a full-featured demo software. Combined with our theoretical proof, we can better explain the technical feasibility of our scheme. As shown in Figs. 12, 13, our demo not only authenticate each other but also negotiate the session key.

Note that our experiment is a pure software demonstration, so it is simplified in terms of the need to extract biometrics from hardware. We directly use the information

Fig. 12. Server result of authentication and agreement key.

Fig. 13. Client result of authentication and agreement key.

entered by the user as biometrics token. This work is mainly based on the MIRACL open source project. In order to implement our protocol, we have made minor modifications to some of the source code in this open source project. To control the length of the article, we have uploaded the detailed demonstration process, compilation environment, code, etc. as supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2019.2909890.

6 Conclusion

To resist the exhaustion of password attack on the two-factor MAKA protocols, a large number of three-factor MAKA protocols have been proposed. However, almost all three-factor MAKA protocols don't provide formal proofs and dynamic user management mechanism. In order to achieve more flexible user management and higher security, this paper proposes a new three-factor MAKA protocol that supports dynamic revocation and provides formal proof. The security shows that our protocol achieves the security properties of requirements from multi-server environments. On the other hand, through the comprehensive analysis of performance, our protocol doesn't sacrifice efficiency while improving the function. On the contrary, the proposed protocol has great advantages in terms of the total computation time.

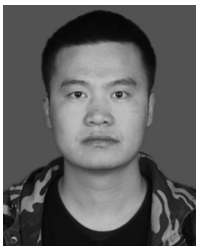
Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2016YFB0800601), the Natural Science Foundation of China (No. 61303217,61502372), the Natural Science Foundation of Shaanxi province (No. 2013JQ8002,2014JQ8313).

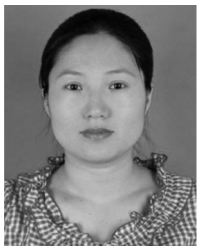
References

[1] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.
[2] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, Aug. 2011.
[3] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 6, pp. 568–581, Nov./Dec. 2014.
[4] D. He, S. Zeadally, N. Kumar, and J. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Syst. J.*, vol. 22, pp. 1–12, 2016.
[5] L. Li, L. Lin, and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1498–1504, Nov. 2001.
[6] W. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Trans. Consumer Electron.*, vol. 50, no. 1, pp. 251–255, Feb. 2004.
[7] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proc. Int. Conf. Cyberworlds*, 2004, pp. 417–422.
[8] J.-L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Comput. Secur.*, vol. 27, no. 3C4, pp. 115–121, 2008.
[9] W. Tsaur, J. Li, and W. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *J. Syst. Softw.*, vol. 85, no. 4, pp. 876–882, 2012.
[10] Y. Liao and C. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Comput. Syst.*, vol. 29, no. 3, pp. 886–900, 2013.
[11] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
[12] D. Wang and P. Wang, *Offline Dictionary Attack on Password Authentication Schemes Using Smart Cards*. New York, NY, USA: Springer International Publishing, 2015.
[13] J. K. Lee, S. R. Ryu, and K. Y. Yoo, "Fingerprint-based remote user authentication scheme using smart cards," *Electron. Lett.*, vol. 38, no. 12, pp. 554–555, 2002.
[14] C. Lin and Y. Lai, "A flexible biometrics remote user authentication scheme," *Comput. Standards Interfaces*, vol. 27, no. 1, pp. 19–23, 2004.
[15] C. Chang and I. Lin, "Remarks on fingerprint-based remote user authentication scheme using smart cards," *Operating Syst. Rev.*, vol. 38, no. 4, pp. 91–96, 2004.
[16] H. Kim, S. Lee, and K. Yoo, "Id-based password authentication scheme using smart cards and fingerprints," *Operating Syst. Rev.*, vol. 37, no. 4, pp. 32–41, 2003.
[17] M. Scott, "Cryptanalysis of an id-based password authentication scheme using smart cards and fingerprints," *Operating Syst. Rev.*, vol. 38, no. 2, pp. 73–75, 2004.
[18] M. K. Khan and J. Zhang, "Improving the security of 'a flexible biometrics remote user authentication scheme'," *Comput. Standards Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.
[19] E. Yoon and K. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *J. Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.
[20] 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 *Proc. Int. Conf. Comput. Sci. Appl.*, 2012, pp. 391–406.
[21] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.
[22] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.
[23] A. G. Reddy, E. J. Yoon, A. K. Das, V. Odelu, and K. Y. Yoo, "Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment," *IEEE Access*, vol. 5, pp. 3622–3639, Feb. 2017.
[24] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic id-based remote user authentication scheme," *IEEE Trans. Consumer Electron.*, vol. 50, no. 2, pp. 629–631, May 2004.
[25] Y. Wang, J. Liu, F. Xiao, and J. Dan, "A more efficient and secure dynamic id-based remote user authentication scheme," *Comput. Commun.*, vol. 32, no. 4, pp. 583–585, 2009.
[26] K. Yeh, C. Su, N. Lo, Y. Li, and Y. Hung, "Two robust remote user authentication protocols using smart cards," *J. Syst. Softw.*, vol. 83, no. 12, pp. 2556–2565, 2010.
[27] F. Wen and X. Li, "An improved dynamic id-based remote user authentication with key agreement scheme," *Comput. Electr. Eng.*, vol. 38, no. 2, pp. 381–387, 2012.
[28] 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 Trans. Inf. Forensics Secur.*, vol. 11, no. 9, pp. 2052–2064, Sep. 2016.
[29] D. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology Eprint Archive*, vol. 2011, p. 365, 2011.

[30] M. H. Afifi, L. Zhou, S. Chakrabartty, and R. Jian, "Dynamic authentication protocol using self-powered timers for passive internet of things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2927–2935, Aug. 2018.
[31] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.
[32] P. Xie, J. Feng, Z. Cao, and J. Wang, "Genewave: Fast authentication and key agreement on commodity mobile devices," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1688–1700, Aug. 2018.
[33] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 2, pp. 319–330, Feb. 2019.
[34] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.
[35] C. P. Schnorr, *Efficient Identification and Signatures for Smart Cards*. Berlin, Germany: Springer, 1989.
[36] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 1996, pp. 387–398.
[37] N. Fleischhacker, T. Jager, and D. Schröder, "On tight security proofs for schnorr signatures," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2014, pp. 512–531.
[38] Y. Tseng, S. Huang, T. Tsai, and J. Ke, "List-free id-based mutual authentication and key agreement protocol for multiserver architectures," *IEEE Trans. Emerging Top. Comput.*, vol. 4, no. 1, pp. 102–112, Jan. 2016.
[39] K. Y. Choi, J. Y. Hwang, D. H. Lee, and I. S. Seo, "Id-based authenticated key agreement for low-power mobile devices," in *Proc. 10th Australasian Conf. Inf. Secur. Privacy*, 2005, pp. 494–505.
[40] J. C. Cha and J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," in *Proc. 6th Int. Workshop Theory Practice Public Key Cryptography: Public Key Cryptography*, 2003, pp. 18–30.
[41] H. H. Kilinc and T. Yanik, "A survey of sip authentication and key agreement schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, Apr.-Jun. 2014.



Wei Li is currently working toward the MS degree from Xidian University, Xi'an, China. His research interests include authentication encryption and bitcion.



Xuelian Li received the PhD degree in cryptography in 2010. She is now an associate professor with the School of Mathematics and Staticstics, Xidian University. Her research interests include information security and block chain.

Juntao Gao received the PhD degree in cryptography in 2006. He is now an associate professor with the School of Telecommunication and Engineering, Xidian University. His research interests include pseudorandom sequences and block chain.

Haiyu Wang is currently working toward the MS degree from Xidian University, Xi'an, China. Her research interests include security and privacy of bitcoin wallets.