IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 5, 2019, accepted August 29, 2019, date of publication September 4, 2019, date of current version September 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939368

A Blockchain-Based Privacy-Awareness Authentication Scheme With Efficient Revocation for Multi-Server Architectures

LING XIONG[1,2], FAGEN LI[2], SHENGKE ZENG[1,2], TU PENG[3], AND ZHICAI LIU[1]

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[3]School of Software, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Shengke Zeng (zengshengke@gmail.com)

This work was supported in part by the National Science Foundation of China under Grant 61502034 and Grant 61571375, and in part by the Civil Aviation Administration of China under Grant PSDSA201802.

ABSTRACT Multi-server authentication technology has become more and more popular with the extensive applications of networks. Although it has brought great convenience to people's life, security becomes a critical issue and attracts lots of attentions in both academia and industry. Over the past two decades, a series of multi-server authentication schemes without communication with the online registration center in each authentication phase using the self-certified public key cryptography have been proposed to enhance security. However, it may cause the single-point failure problem due to the centralized architecture. Besides, user revocation facility is not well resolved in these schemes. To the best of our knowledge, blockchain technology has lots of advantages, bringing a promising solution to the problems of single-point failure and user revocation compared with the traditional cryptography technologies. In this work, we apply the idea of blockchain technology to construct a privacy-awareness authentication scheme for the multi-server environment, which can achieve distributed registry and efficient revocation. Moreover, the proposed scheme not only provides multiple security requirements like mutual authentication, user anonymity and perfect forward secrecy, but also resists various kinds of malicious attacks. The security of the proposed scheme is proved by rigorous formal proof using the random oracle model. Compared with recently related schemes, the proposed scheme has better communication performance, which make it be very suitable for real-life applications.

INDEX TERMS Blockchain, multi-server, authentication, revocation.

I. INTRODUCTION

With the rapid development of network and information techniques, many applications and services that are based on the Internet platform are emerging one after another. As a result, two-thirds of users tend to reuse the same identities and passwords for multiple applications or services to make memorizing them easier. Although this kind of handing brings much convenience to users, it also comes with a potential security risk. The multi-server authentication mechanism is an effective solution to address this barrier, which only needs users to register once at the registration center. Thus, users can access all registered servers using the smae identity and password [1]. As shown in Fig. 1, a multi-server authentication system includes three roles, namely, a registration center (RC), servers and users [2], [3]. Generally, the RC is the system manager who is responsible to provide a trusted credential, which is also called certificate, to both servers and users. During the process of authentication, the server and the user can authenticate each other through this credential or certificate.

Due to the openness Internet network, the adversary can easily eavesdrop, insert, block, and alter the transmitted messages in the multi-server environment. Hence, it is indispensable to design privacy-awareness authentication schemes for multi-server environment [2]. Over the past two decades, a series of remarkable multi-server privacy-awareness authentication schemes (e.g. [2]–[6]) have been proposed. According to whether requires communicating with the online RC in the authentication process,

The associate editor coordinating the review of this article and approving it for publication was Kaiping Xue.

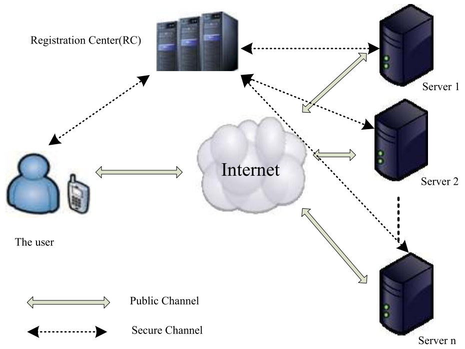


FIGURE 1. The multi-server authentication architecture.

these schemes are divided into two types, namely, online RC schemes and no online RC schemes [2]. Obviously, online RC schemes increase the communication costs and complexity. Therefore, in recent years, no online RC schemes have gradually become the research focus. In this article, we concern with no online RC schemes for multi-server architectures. Generally, the self-certified public key cryptography (SCPKC) [2] has been used in multi-server authentication schemes to achieve no online RC. Although these schemes have many advantages, such as low communication overhead. However, there still exist some security and design issues needed to be resolved as follows.

- **Single-point failure:** In the multi-server environment, when a new user wants to access a server, he/she has to register first. Generally, there is only a single RC in the traditional multi-server architecture. Thus, the only RC has full knowledge of the registered users' information (such as identity, secret key, etc.) and can trace the actions of users [7]. Additionally, if the single RC is a failure under attacks or natural disasters, the whole stored data will be in danger [8].
- **User revocation:** Several circumstances in the multi-server system require the user revocation mechanism to revoke misbehaving/compromised users from the system within the stipulated expiration dates [9], [10]. To the best of our knowledge, the existing SCPKC multi-server authentication protocols (e.g. [2], [4], [10], [11]) adopts two measures to revoke users for access authorization. The first is the black/white (or revocation/permission) list mechanism [10]. Once the user is revoked, the RC will notify each server to add the revoked user to the black/white list. Thus, the RC and servers may require to manage a backend channel for the black/white list. The second is the expiration time method [11]. The user's credential is bound by a time period. Before the expiration time, users remain legitimate unless the time has expired. Unfortunately, if the credential is obtained by an adversary within the expiration time, the adversary can access servers in the multi-server system using the old credential.

Certainly, traditional cryptography techniques generally may not be applicable to the above two issues. Blockchain technology, that has several additional technological advantages like decentralized, unforgeability, etc, offers a promising alternative solution. Motivated by this idea, in this work, we design a blockchain-based privacy-awareness authentication scheme for the multi-server system. The proposed scheme avoids the single RC problem and provides an effective user revocation method. Furthermore, our scheme can achieve mutual authentication, user anonymity, perfect forward secrecy, untraceability, and resistance to various attacks.

A. RELATED WORK

In this section, we first introduce the recent related work of multi-server authentication schemes. Then a rough overview of the blockchain and its application in authentication technique will be described.

1) MULTI-SERVER AUTHENTICATION

In 2001, Li *et al.* [12] proposed the first password-based multi-server authentication scheme based on neural networks. While the neural networks are so complex that the scheme cannot be practical. To enhance efficiency, Juang [13] proposed a multi-server authentication scheme using symmetric cryptography. However, Juang's scheme is vulnerable to insider attack and off-line dictionary attacks. In order to increase security, a series of improved schemes (e.g. [14]–[16]) had been proposed. However, these schemes still suffer from some security problems, such as perfect forward security, impersonation attack, user anonymity, etc. In 2009, Liao and Wang [17] designed a dynamic ID-based authentication scheme for the multi-server environment. Unfortunately, this scheme cannot resist impersonation attack, server spoofing attack. Although several schemes [18]–[20] had been proposed to improve after Liao er.al.'s scheme, there were still security weaknesses. In 2015, He and Wang [4] presented a robust biometrics-based authentication scheme for the multi-server environment. They claimed that their scheme could support various security requirements and resist a variety of attacks. Later, Odelu *et al.* [3] pointed out that He and Wang's scheme was vulnerable to known session-specific temporary information attack, impersonation attack, wrong password login attack. To address these issues, they put forward a secure biometrics-based multi-server authentication protocol using smart cards, which can provide the problem of user revocation and resist various attacks. Recently, more and more multi-server authentication schemes had been applied in various environments, such as cloud computing (e.g. [21]) and wireless sensor networks (e.g. [22]). Unfortunately, most of them like Odelu et al.'s scheme [3] requires a trust third-party to participate in each authentication phase, which may make the trusted third party being a bottleneck of communication. To solve this issue, several multi-server authentication schemes without online third-party participation had been proposed [2], [23]–[26]. Obviously, these schemes using the SCPKC have lower communication cost. So, they have become very popular among researchers and have been applied to mobile cloud computing environment (e.g. [5], [11], [27]). However, to the best of our knowledge, these multi-server authentication schemes using the SCPKC cryptography adopt the black/white list mechanism or expiration time method to revoke users, which may cause communication costs or security problem. Additionally, all of these multi-server authentication schemes share a common problem: users have to register on a single trust third party. Therefore, how to design a multi-server authentication scheme with a distributed registry and efficient revocation is an urgent problem to be solved.

2) BLOCKCHAIN AND ITS APPLICATION IN AUTHENTICATION

In 2008, the blockchain was originally published in the cryptography mailing group by a scholar named Nakamoto [28]. In recent years, with the increasing popularity of Bitcoin, blockchain technology research has been motivated to grow quickly. The blockchain is a distributed peer-to-peer network where transactions are posted and verified by non-trusting network members via a cryptographically verifiable manner [29], [30]. One of a key challenge in maintaining the blockchain data structure is the consensus algorithm, such as proof-of-work (PoW), proof-of-stake (PoS), delegated proof-of-stake (DPoS), etc. PoW utilizes a physical resource (either storage or computational power) to achieve the leader election process, in which miners have to complete some difficult but easily verifiable task. Bitcoin [28], Namecoin [31] and Litecoin [32] are typical PoW-based cryptocurrencies. The disadvantage of this kind of consensus algorithm is that it expends a lot of energy and causes a serious waste. PoS is an alternative consensus algorithm to resolve the waste of energy. Rather than miners investing computational resources, PoS randomly selects one of the miners proportionally to be the leader [33], [34]. Most recently, Kiayias *et al.* [35], [36] presented the first blockchain protocol named Ouroboros based on PoS with rigorous security guarantees, which offers qualitative efficiency advantages over blockchains based on PoW. DPoS [37] is a variant of PoS, in which the leader is performed by voting. Due to the better performance in computation and energy efficiency, many cryptocurrencies adopt PoS or DPoS as their consensus algorithm after Bitcoin.

The authentication technology based on blockchain has come to the foreground in recent years and receives more and more attentions [8], [38]. In 2014, Conner *et al.* [39], [40] proposed the first blockchain public key infrastructure (PKI) system called Certcoin, which provides a solution to some security problems, such as DigiNotar incident [41], in the traditional PKI. However, all network numbers can find the link between the identity and its corresponding public key by viewing the blockchain. Then, they can trace the actions of identities. Thus, privacy cannot be provided by Certcoin. To address this issue, Axon [42] and Axon and Goldsmith [43] designed a privacy-awareness blockchain PKI, which achieve user anonymity through short term online public keys. Obviously, Axon et al.'s scheme is sacrificing storage and efficiency in exchange for privacy. Different from the above blockchain PKI, Matsumoto and Reischuk [44] presented Instant Karma PKI (IKP), which offers automatic responses to certificate authority (CA) misbehavior using smart contract [45], [46] and incentives for those who help detect misbehavior. Although these existing schemes explore the potential of applying blockchain technology for authentication, there still exist many challenges. The current research on the blockchain for the multi-server system has not been reported. In this paper, we are to address these challenges.

TABLE 1. **Notations.**

Notation	Descriptions
U_i	The remote user
S_j	The server
ID_i	Unique identity of U_i
ID_{sj}	Unique identity of S_j
PW_i	Password of U_i
REV_{ui}	Revocation status, if $REV_{ui} = 1$, it specifies that U_i has been revoked, otherwise not.
G	An additive group with order q
$Tx_1, Tx_2, ..., Tx_n$	The number of transaction in blockchain
T	Current time stamp values
$h_i(.), i = 0, 1, 2, 3, 4$	One-way hash function
$MAC(k, M)$	Authenticate a message M using the entity's secret key k
$X\|\|Y$	Concatenate operation
$\oplus$	XOR operation

B. CONTRIBUTIONS

In this paper, we present a blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures. The contributions of the paper are summarized mainly as follows.

(1) The proposed scheme focuses on the combination of the blockchain and multi-server authentication. The permission servers as blockchain network miners utilize Ouroboros algorithm to ensure the consistency. Thus the false issuing credential can be avoided.

(2) The proposed scheme can solve the problem of a single RC.

(3) The proposed scheme increases user revocation mechanism to prevent the misuse of the smart card when it is lost/stolen.

(4) The proposed scheme has higher efficiency in communication, which makes it more suitable for real-life applications.

C. ORGANIZATION OF THE PAPER

The rest of this paper is organized as follows. Section II reviews the background for our system. Section III shows the system building blocks in our system. Section IV presents the detailed procedure of the proposed scheme. Section V gives security analysis of our scheme. The computation, communication costs and the qualitative property analysis of the proposed scheme are discussed in Section VI. Finally, Section VII concludes this paper. All the notations mentioned in our proposed scheme are defined in Table 1.

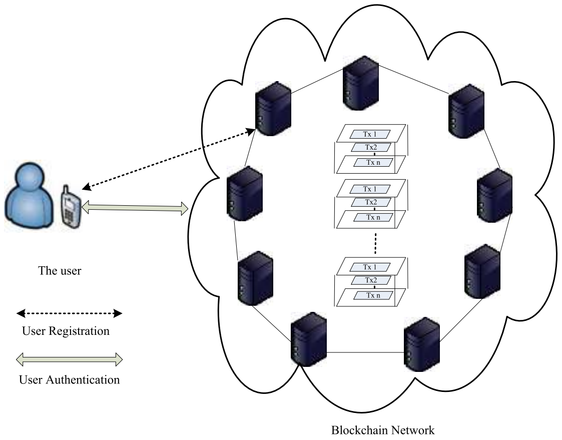


FIGURE 2. The multi-server authentication architecture based on blockchain.

II. BACKGROUND

This section will introduce the system model and the security requirements of our scheme.

A. SYSTEM MODEL

1) PROTOCOL PARTICIPANT

The proposed scheme involves two participants: the user U_i and the server S_j. S_j in the blockchain network is the service provider who is assessed by the remote user.

- **Users:** The remote users with smart card or mobile device, are able to access multiple servers. As shown in Fig. 2, when these users wish to ask for an access request to the multi-server system, they need to register in the nearest server first, who will post a corresponding transaction to the blockchain network (see Section IV-B).
- **Servers:** In our multi-server system, the permission servers as the role of miners or consensus nodes constitute the blockchain network. We assume that servers in our system are semi-trusted parties, which means that servers may misbehave on their own but will not conspire with either of the other servers [47]. So, the proposed system adopts private or consortium blockchain, which adopts an efficient consensus mechanism like PoS. As shown in Fig. 2, when a server S_j receives registering request from a user U_i, he/she need to check the validity of user's public key and personal information, such as passport, identification card, mobile number or any authorized identities. After successful verification, S_j signs user's identity and public key using her/his own private key and posts the signature to the blockchain network.

2) PROTOCOL EXECUTION

The proposed scheme has five phases: the initialization phase, the user registration phase, the mutual authentication phase, the password update phase, and user revocation and re-registration phase. The initialization phase, the user registration phase and user revocation and re-registration phase are assumed to be executed securely.

3) ADVERSARY MODEL

The adversary A has two goals. One is that A can successfully impersonate U_i authenticating to S_j, and the other is A can successfully impersonate S_j authenticating to U_i. Assume that A is a probabilistic polynomial time attacker, and the feasible attacks are summarized as follows:

- A can control the channel between the user and the server. It means that A can eavesdrop, insert, block, and alter the transmitted messages through the communication channel.
- A can obtain one of the two authentication factors: the smart card or the password. If A has obtained the smart card, he/she can extract the secret information in the smart card. Then he/she has the capability of enumerating the password space $|D_{PW}|$.
- A may be another legitimate but malicious user in the multi-server system.

B. SECURITY REQUIREMENTS

According to the recent literatures for multi-server authentication (e.g. the literatures [2]–[5], [48]), the blockchain-based multi-server authentication scheme should satisfy the following security properties.

- **Mutual authentication:** It ensures that servers and users can successfully authenticate each other.
- **User anonymity:** It ensures that the adversary cannot obtain users' identities through the transmitted messages in the public channel.
- **Un-traceability:** It ensures that the adversary cannot trace users' behaviors from the transmitted messages in the public channel.
- **Efficient and user-friendly password update:** It ensures that users can freely update passwords and should be allowed updating passwords without servers' assistance.
- **Multi-factor security:** Multi-factor (assuming there are n factors, generally, $n = 2$ or $n = 3$) security implies the protocol is still secure when $n - 1$ of n factors are lost [49], [50]. In our proposed scheme, we adopt $n = 2$, the password and the smart card are two used factors. So, it ensures that the blockchain-based two-factor authentication scheme for multi-server architecture should be able to satisfy the following requirements. 1) If an adversary has obtained the smart card and gets its secret value, he/she should not be able to perform the off-line password guessing attack; 2) The adversary who knows the password should not be able to perform impersonation attack without secret value in the smart card [48], [51].
- **Resistance to wrong password login/update attack:** To avoid the waste of computation and communication resources for invalid login, it is necessary to check the correctness of the password in the user login procedure. Besides, once a mistake occurs in the password update phase, a valid user can no longer log in the server using the same smart card. Therefore, the blockchain

based multi-server authentication scheme should consider quick detection mechanism to avoid wasting the server's resources [3], [48].

- **User revocation and re-registration:** It ensures that the blockchain based multi-server authentication scheme should support user revocation and re-registration. If the user's smart card is lost or stolen, there must be some measures to prevent the adversary to impersonate the user. In other words, if an adversary has obtained the identity of the user, he/she cannot impersonate the user in the registration phase [3], [48].
- **Secure session key agreement:** It ensures that two participants should be able to agree with a secure session key, which will protect transmitted messages in future communications.
- **Perfect forward secrecy:** It ensures that the adversary is unable to obtain the session key generated in previous sessions even if the long-term private keys of the two participants are leaked.
- **Resistance to various attacks:** It ensures that various attacks should be prevented in the multi-server environment, such as impersonation attack, man-in-the-middle attack, replay attack and stole-verifier attack.

III. SYSTEM BUILDING BLOCKS

In this section, we will introduce the cryptographic primitives, transaction and consensus mechanism of blockchain network used in the proposed blockchain-based scheme.

A. CRYPTOGRAPHIC PRIMITIVES

The proposed scheme leverages the elliptic curve digital signature algorithm ECDSA [52]. The digital signature consists of three algorithms, which will be reviewed as follows:

- $keygen(1^k) \rightarrow (SK, PK)$: the function generates a private key SK and a corresponding public key PK with the security parameter k.
- $Sig(SK, m) \rightarrow S_i$: the function computes a digital signature value S_i of message m using the private key SK.
- $Ver(PK, S_i, m) \rightarrow b \in \{0, 1\}$: the function verifies whether the value S_i is corrcct signature value of message m using the public key PK.

The signature algorithm should be unforgeable [39], [53], which means that no probabilistic polynomial-time adversary can forge a legal signature value S_i without the private key SK.

B. TRANSACTION

As shown in Table 2, instead of posting transactional information in the transaction of the bitcoin system, the transactions (Tx) in our system include identity, public key, revocation status and signature. The detail of the transaction structure is described below.

- **from:** represents the identity of the user.
- **UPK:** represents the public key linked with the user.
- **to:** represents the identity of the server who handle registration information from the user.
- **SPK:** represents the public key of the server who handle registration information from the user.

TABLE 2. The transaction structure of our system.

from	UPK	to	SPK	REV	T	USIG	SSIG

- **REV:** represents the revocation status of the user. If the value of REV is one, it represents the user is revoked. Otherwise not.
- **T:** represents the current timestamp.
- **USIG:** represents the signature value of the user's information (the identity of the user, the revocation status, the current timestamp) with the user's private key.
- **SSIG:** represents the signature value of the user's information (the identity of user, the public key of the user, the identity of the server, the revocation status, the current timestamp) with the server's private key.

C. BLOCKCHAIN

The blockchain is made of a chronologically ordered chain of blocks. Every block includes a certain amount of transactions and each block links to its predecessor by a hash value [54]. As shown in Fig. 3, the structure of our blockchain system is similar to the Bitcoin [39], which includes the number of block, the hash value of the previous block, the timestamp and Merkle tree root. In our blockchain-based multi-server system, the permission servers are miners in blockchain network, who will participate in issuing the next block. Generally, miners have to compete to complete some PoW to create a new block (e.g. Bitcoin [39], Namecoin [31]). However, these systems have always relied on large computing power to verify transactions and write them into a new block, which costs a lot of money and energy. Another alternative to PoW is the concept of PoS [35], [36] or DPoS [37], which randomly selects one of the miners to complete a new block. Obviously, the two later consensus mechanisms PoS and DPoS are more effective. In our design, a provable secure PoS protocol name Ouroboros [35], [36] is selected as our consensus mechanism to write a new block. Ouroboros can process hundreds or a couple of thousand transactions within seconds and eliminate the needs for an energy-hungry PoW. When a user initiates a registration requirement to the server the blockchain node (the server), the blockchain node will verify his/her information. After successful checking, the blockchain node will post the transaction into the blockchain network and the whole nodes will generate a new block through Ouroboros algorithm.

IV. THE PROPOSED SCHEME

This section will describe the details of the proposed privacy-awareness authentication scheme. Our proposed scheme consists of five phases: initialization phase, user registration phase, mutual authentication phase, password update phase, and revocation and re-registration phase. Each phase in detail will be introduced as follows.

A. INITIALIZATION PHASE

Assuming that there are n permission servers. In the initialization phase, all servers S_j agrees upon an additive group

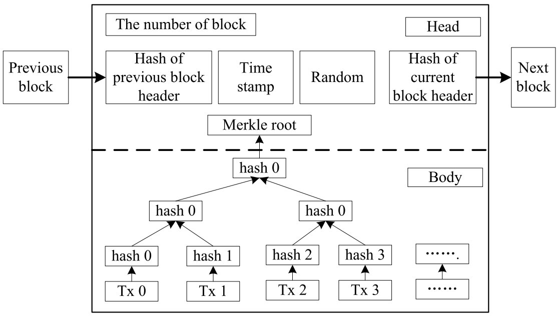


FIGURE 3. The bitcoin blockchain structure.

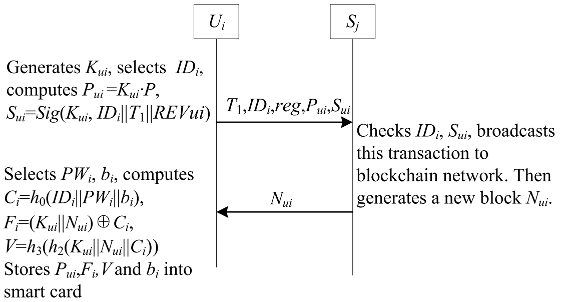

U_i
S_j
Generates K_{ui}, selects ID_i, computes $P_{ui}=K_{ui}\cdot P$, $S_{ui}=Sig(K_{ui}, ID_i||T_1||REVui)$
$T_1,ID_i,reg,P_{ui},S_{ui}$
Checks ID_i, S_{ui}, broadcasts this transaction to blockchain network. Then generates a new block N_{ui}.
N_{ui}
Selects PW_i, b_i, computes $C_i=h_0(ID_i||PW_i||b_i)$, $F_i=(K_{ui}||N_{ui})\oplus C_i$, $V=h_3(h_2(K_{ui}||N_{ui}||C_i))$
Stores P_{ui},F_i,V and b_i into smart card

FIGURE 4. The user registration phase.

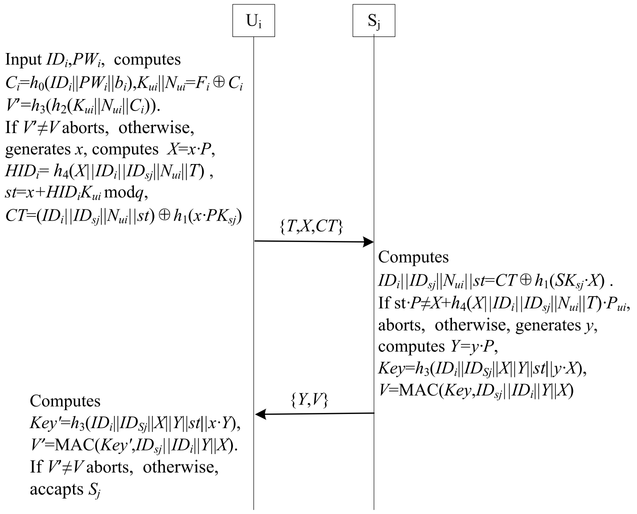

$\mathrm{U_i}$
$\mathrm{S_j}$
Input ID_i,PW_i, computes
$C_i=h_0(ID_i||PW_i||b_i)$,$K_{ui}||N_{ui}=F_i\oplus C_i$
$V'=h_3(h_2(K_{ui}||N_{ui}||C_i))$.
If $V'\neq V$ aborts, otherwise,
generates x, computes $X=x\cdot P$,
$HID_i= h_4(X||ID_i||ID_{sj}||N_{ui}||T)$,
$st=x+HID_iK_{ui}$ mod$q$,
$CT=(ID_i||ID_{sj}||N_{ui}||st)\oplus h_1(x\cdot PK_{sj})$
$\{T,X,CT\}$
Computes
$ID_i||ID_{sj}||N_{ui}||st=CT\oplus h_1(SK_{sj}\cdot X)$.
If st$\cdot P\neq X+h_4(X||ID_i||ID_{sj}||N_{ui}||T)\cdot P_{ui}$,
aborts, otherwise, generates y,
computes $Y=y\cdot P$,
$Key=h_3(ID_i||ID_{Sj}||X||Y||st||y\cdot X)$,
V=MAC(Key,$ID_{sj}||ID_i||Y||X$)
$\{Y,V\}$
Computes
$Key'=h_3(ID_i||ID_{Sj}||X||Y||st||x\cdot Y)$,
V'=MAC(Key',$ID_{sj}||ID_i||Y||X$).
If $V'\neq V$ aborts, otherwise,
accapts S_j

FIGURE 5. Mutual authentication phase.

of point G with order q, P is a generator of G, and five hash functions $h_i : \{0, 1\}^* \rightarrow \{0, 1\}^l$, $h_2 : \{0, 1\}^* \rightarrow \{0, 1, 2, \ldots, 1023\}$, $h_4 : \{0, 1\}^* \rightarrow Z_q^*$, where l is the bit length of output and $i = 0, 1, 3$. Every server S_j generates its private key $SK_{sj} \in Z_q^*$ and calculates the public key $PK_{sj} = SK_{sj} \cdot P$. We assume without loss of generality that each public key PK_{sj} is knowing by all servers and users. Then S_j stores SK_{sj} into its memory as secret and publishes the parameters $\{G, P, PK_{sj}, h_0, h_1, h_2, h_3, h_4\}$.

B. USER REGISTRATION PHASE

When a user U_i wants to access a multi-server system, he/she must register with any one of the servers in multi-server architectures. As shown in Fig. 4, the procedure of user registration is described as follows.

(1) A user U_i chooses a nearest server S_j to him/her in the blockchain network, selects identity ID_i and a random number $K_{ui} \in Z_q^*$, sets $REV_{ui} = 0$, and calculates the public key $P_{ui} = K_{ui} \cdot P$, $S_{ui} = Sig(K_{ui}, ID_i||T_1||REV_{ui})$, where T_1 is the timestamp. U_i submits the messages $\{T_1, ID_i, reg, P_{ui}, S_{ui}\}$ and his personal information (e.g.passport, identification card and mobile number) to S_j through a secure channel (The signature S_{ui} demonstrates that the user is able to sign with K_{ui}), where *reg* is the registration requirement.

(2) Upon receipt of the message, S_j at first checks the correctness of personal information and timestamp. Then, S_j sets $REV_{ui} = 0$ and verifies whether the equation $Ver(P_{ui}, S_{ui}, ID_i||T_1||REV_{ui}) = 1$ holds. If it does not hold, S_j rejects the registration request. Otherwise, S_j checks whether ID_i has been previous registered through lookup the blockchain. If it has been registered and $REV_{ui} = 0$, S_j rejects the registration request. Otherwise, S_j computes $Sj_{ui} = Sig(SK_{sj}, ID_i||ID_{sj}||P_{ui}||REV_{ui} = 0||T_2)$ and broadcasts the transaction $\{ID_i, P_{ui}, ID_{sj}, PK_{sj}, REV_{ui}, T_2, S_{ui}, Sj_{ui}\}$ to blockchain network, where T_2 is the timestamp. After that, the block miners generat a new candidate block N_{ui} by Ouroboros [35], [36] algorithm, where N_{ui} is the number of block in the blockchain. Finally, S_j transmits $\{N_{ui}\}$ to U_i through a secure channel. (The signature Sj_{ui} demonstrates that the server S_j has verified that P_{ui} is the corresponding public key of the identity owner. S_j has to take responsibility for this claim.).

(3) After received the message, U_i selects password PW_i and a random number b_i. Then, U_i computes $C_i = h_0(ID_i||PW_i||b_i)$, $F_i = (K_{ui}||N_{ui}) \oplus C_i$, $V = h_3(h_2(K_{ui}||N_{ui}||C_i))$. Finally, U_i stores P_{ui},F_i,V and b_i into the secret memory of smart card.

C. MUTUAL AUTHENTICATION PHASE

When the user U_i wants to log in a server S_j, U_i needs to achieve mutual authenticate with S_j. As shown in Fig. 5, the process of mutual authentication is as follows.

(1) U_i inputs ID_i and PW_i into the smart card. The smart card computes $C_i = h_0(ID_i||PW_i||b_i)$, $K_{ui}||N_{ui} = F_i \oplus C_i$, $V' = h_3(h_2(K_{ui}||N_{ui}||C_i))$ and checks whether V' and V are equal. If not, the smart card terminates this session. Otherwise, it generates a random number $x \in Z_q^*$, and computes $X = x \cdot P$, $HID_i = h_4(X||ID_i||ID_{sj}||N_{ui}||T)$, $st = x + HID_iK_{ui}\text{mod}q$, $CT = (ID_i||ID_{sj}||N_{ui}||st) \oplus h_1(x \cdot PK_{sj})$, where T is the current timestamp. Then U_i sends the messages $\{T, X, CT\}$ to S_j by a public channel.

(2) After received the messages $\{T, X, CT\}$, S_j first checks the timestamp T, and computes $ID_i||ID_{sj}||N_{ui}||st = CT \oplus h_1(SK_{sj} \cdot X)$. Then, S_j checks whether the following three conditions are satisfied.

- whether ID_i exists in the block N_{ui}.

- whether ID_i's revocation status $REV_{ui} = 0$ holds in the block N_{ui}.
- the blocks $N_x > N_{ui}$ do not include the tuple $\{ID_i, P_{ui}, ID_{sj}, PK_{sj}, REV_{ui} = 1, S_{ui}, Sj_{ui}\}$.

If one of the above conditions does not hold, S_j terminates the session. Otherwise, S_j gets the public key P_{ui} of U_i, and verifies whether the equation $st \cdot P = X + h_4(X||ID_i||ID_{sj}||N_{ui}||T) \cdot P_{ui}$ holds. If it does not hold, S_j terminates the session. Otherwise, S_j generates a random number $y \in Z_q^*$, and computes $Y = y \cdot P$, $Key = h_3(ID_i||ID_{sj}||X||Y||st||y \cdot X)$, $V = MAC(Key, ID_{sj}||ID_i||Y||X)$. After that, S_j sends $\{Y, V\}$ to U_i via a public channel.

(3) Upon receiving the messages $\{Y, V\}$ from S_j, U_i computes $Key' = h_3(ID_i||ID_{sj}||X||Y||st||x \cdot Y)$, $V' = MAC(Key', ID_{sj}||ID_i||Y||X)$, and checks whether V' matches with the received V. If it holds, U_i completes the authentication. Otherwise, U_i fails to authenticate the S_j.

D. PASSWORD UPDATE PHASE

When a user U_i wants to update the password, he/she needs to run the following steps.

(1) U_i inputs ID_i, PW_i into the smart card SC. SC computes $C_i = h_0(ID_i||PW_i||b_i)$, $K_{ui}||N_{ui} = F_i \oplus C_i$, $V' = h_3(h_2(K_{ui}||N_{ui}||C_i))$, and checks whether V' and V are equal. If not, SC fails to authenticate U_i, and rejects the request of the password update. Otherwise U_i inputs a new password PW_i^*.

(2) SC computes $C_i^* = h_0(ID_i||PW_i^*||b_i)$, $F_i^* = F_i \oplus C_i \oplus C_i^*$, $V^* = h_3(h_2(K_{ui}||N_{ui}||C_i^*))$.

(3) Finally, F_i^* and V^* are stored in SC to replace F_i and V respectively.

E. USER REVOCATION AND RE-REGISTRATION PHASE

Once the smart card is lost or stolen, the user must revoke his/her account and re-register with any one of the servers in multi-server architectures using the same identity. The process of user revocation and re-registration is described as follows.

(1) U_i chooses the nearest server S_j in the multi-server system, selects a new random number $K'_{ui} \in Z_q^*$, set $REV_{ui} = 0$, and calculates the public key $P'_{ui} = K'_{ui} \cdot P$, $S'_{ui} = Sig(K'_{ui}, ID_i||T_3||REV_{ui})$, where T_3 is the timestamp. U_i submits the messages $\{T_3, ID_i, rev, P'_{ui}, S'_{ui}\}$ and some personal information to S_j through a secure channel, where rev is the revocation requirement.

(2) Upon receipt of the messages, S_j at first checks the correctness of personal information and timestamp. Then S_j set $REV_{ui} = 0$, and verifies whether the equation $Ver(P'_{ui}, S'_{ui}, ID_i||T_3||REV_{ui}) = 1$ holds. If they do not hold, S_j rejects the revocation and re-registration request. Otherwise, S_j gets the corresponding old public key of ID_i by looking up blockchain, computes $Sj1_{ui} = Sig(SK_{sj}, ID_i||ID_{sj}||P_{ui}||REV_{ui} = 1||T_4)$, $Sj2_{ui} = Sig(SK_{sj}, ID_i||ID_{sj}||P'_{ui}||REV_{ui} = 0||T_5)$, broadcasts two transactions $\{ID_i, P_{ui}, ID_{sj}, PK_{sj}, REV_{ui} = 1, T_4, S_{ui}, Sj1_{ui}\}$, $\{ID_i, P'_{ui}, ID_{sj}, PK_{sj}, REV_{ui} = 0, T_5, S'_{ui}, Sj2_{ui}\}$ to blockchain network, where T_4 and T_5 are current timestamp. The block miners generate a new candidate block N'_{ui} by Ouroboros algorithm, where N'_{ui} is the number of block in the blockchain. Then, S_j transmits $\{N'_{ui}\}$ to U_i through a secure channel.

(3) After received the message, U_i selects password PW'_i and a random number b'_i. Then, U_i computes $C'_i = h_0(ID_i||PW'_i||b'_i)$, $F'_i = (K'_{ui}||N'_{ui}) \oplus C'_i$, $V' = h_3(h_2(K'_{ui}||N'_{ui}||C'_i))$. Finally, U_i stores P'_{ui},F'_i,V' and b'_i into the secret memory of smart card.

V. SECURITY ANALYSIS OF THE PROPOSED SCHEME

In this section, we will show our proposed scheme meets all the security requirements in Section II. Because the initialization phase, user registration phase, and user revocation and re-registration phase are executed in the secure channel. The proposed scheme may suffer security and privacy threats in the authentication phase. Therefore, in this section, we demonstrate the authentication phase is secure.

Security Model. Based on literature [50], [53], [55]–[57], we proposed a security model for our scheme. The security model of our scheme is defined by a game played by a probabilistic polynomial time (PPT) adversary A and a PPT Turing machine ζ. Let instance $\prod_U^s$ be the user oracle in session s, $\prod_S^s$ be the server oracle in session s. A can make oracle queries as follows.

(1) $ExtractUi-Oracle$: This query simulates A registration as a legitimate user U_i. A issues this inquiry with U_i's identity ID_i. ζ generates the number of block N_{ui}, U_i's private key and public key, stores them in the list L_U and returns N_{ui} and ID_i to A.

(2) $ExtractSj-Oracle$: This query simulates A registration as a legitimate server S_j. A issues this inquiry with S_j's identity ID_{sj}. ζ generates S_j's private key and public key, stores them in the list L_S.

(3) $Send - Oracle(\iota, s, \iota', M)$: This query simulates the participate ι sends message M to the oracle $\prod_\iota^s$. A issues inquiry and receives a response which is specified by the protocol.

(4) $Reveal - Oracle(U_i, S_j, s)$: This query simulates the leakage of session key attack and will output the session key Key.

(5) There are three corruption queries:

   a) $Corrupt(ID_i, PW_i)$: This query simulates password leakage attack, and will output the user password PW_i.

   b) $Corrupt(ID_i, SC_i)$: This query simulates the smart card loss attack, and will output the secret information stored in the smart card SC_i.

   c) $Corrupt(S_j)$: This query simulates the server compromise attack.

Definition 1: Matching sessions: The session in instance $\prod_U^s$ and the session in instance $\prod_S^{s'}$ are said to be matching if $s = s'$, $pid_U = S$, $pid_S = U$ and both have accepted, where pid_U and pid_S denote as a peer identity.

Definition 2: Security authentication protocol: A authentication scheme is secure if the following properties hold:

- $\prod_U^s$ and $\prod_S^s$ are matching session, and they accept each other.
- $\prod_U^s$ and $\prod_S^s$ derive the same key.
- The probability of $\prod_U^s$ accepted A as $\prod_S^s$ is negligible.
- The probability of $\prod_S^s$ accepted A as $\prod_U^s$ is negligible.

A. FORMAL SECURITY ANALYSIS

To prove the security of our proposed scheme, we assume that our scheme is defined by a game played an adversary A and a Turing machine ζ. At first, we give two mathematical problems used for our security analysis as follows.

Definition 3: Discrete Logarithm (DL) Problem: Given $X = x \cdot P$, where $x \in Z_q^*$, $X \in G$, it is infeasible to compute x.

Our concrete protocol is as below.

(1) $U_i \rightarrow S_j$: $M_1 = \{T, X, CT\}$.

(2) $S_j \rightarrow U_i$: $M_2 = \{Y, V\}$.

Lemma 1: (Secure User Authentication): In the proposed scheme, if h_0, h_1, h_2, h_3, h_4 are ideal random functions, the DL problem is hard and $\prod_S^s$ has been accepted, then there is no polynomial adversary against our proposed scheme who can forge a legal user authentication message with a non-negligible probability.

Proof. We assume that the adversary A can forge a legitimate user authentication message with a non-negligible probability. Then there is a PPT Turing machine ζ who can win the DL problem with a non-negligible probability by employing A. We assume that the probability of the advantage of DL problem is $Pr_{win}[DL]$.

Given an instance $(P, P_{uc} = K_{uc} \cdot P)$ of DL problem, the task of ζ is to compute $(K_{ui} \in Z_q^*)$. To win the game, ζ must simulate an environment of our proposed scheme which is indistinguishable from the real proposed scheme to the adversary A. Hence, ζ should answer all oracle queries issued by A. To achieve this goal, ζ needs to generate all initialization parameters $\{G, P, PK_{sj}, h_0, h_1, h_2, h_3, h_4\}$ and public them. Besides, ζ needs to generate all users' private key $SK_i \in Z_q^*$ except for the challenger ID_c's private key K_{uc} and calculates their public key $P_{ui} = K_{ui} \cdot P$. $\prod_U^s$ denotes the user oracle. $\prod_S^s$ denotes the server oracle. Then, ζ answers A's queries as follows:

1) *$Hi(m_i)$*: The hash query $Hi(m_i)$, $i = 0, 1, 2, 3, 4$ maintains a list L_{hi} with initialized empty. ζ checks whether the message m_i exists in L_{hi}. If it exists, ζ returns its value h_i to A. Otherwise, ζ generates a random number h_i, stores the tuple (m_i, h_i) into L_{hi} and returns h_i to A.

2) *$ExtractUi - Oracle$*: In this query ζ maintains a list L_U with initialized empty. ζ checks if a tuple $(ID_i, P_{ui}, K_{ui}, N_{ui})$ exists in L_U. If it exists, ζ returns ID_i and N_{ui} to A. Otherwise, ζ operates as follows:
   - If $ID_i = ID_c$, ζ generates a random number as the number of block N_{ui}, sets $K_{ui} = \perp$, and asks the user oracle $\prod_U^s$ to get ID_i's public key P_{ui}. ζ stores the tuple $(ID_c, P_{uc}, K_{uc}, N_{uc})$ into L_U and returns ID_c and N_{uc} to A.
   - If $ID_i \neq ID_c$, ζ generates a random number as the number of block N_{ui}, selects a random number $K_{ui} \in Z_q^*$, and calculates the public key $P_{ui} = K_{ui} \cdot P$. ζ stores the tuple $(ID_i, P_{ui}, K_{ui}, N_{ui})$ into L_U and returns ID_i and N_{ui} to A.

3) *$ExtractSj - Oracle$*: In this query, ζ maintains a list L_S with initialized empty. ζ checks if a tuple $(ID_{sj}, PK_{sj}, SK_{sj})$ exists in L_S. If it exists, ζ returns ID_{sj} to A. Otherwise, ζ generates a random number SK_{sj}, calculates $PK_{sj} = SK_{sj} \cdot P$, stores the tuple $(ID_{sj}, PK_{sj}, SK_{sj})$ into L_S and returns ID_{sj} to A.

4) *$Send - Oracle(U_i, s, S_j, M)$*: In this query, A sends the first message M_1 to ζ. ζ decrypts CT and obtains ID_i and P_{ui}, ζ operates according to the specification of the proposed scheme and returns M_2 to A.

5) *$Send - Oracle(S_j, s, U_i, M)$*: After receiving this query, ζ checks whether the equation $ID_i = ID_C$ holds. If not, ζ operates according to the specification of the proposed scheme and returns the first message M_1 to A. Otherwise, ζ asks the user oracle $\prod_U^s$ to get M_1 and returns it to A.

6) *$Reveal - Oracle(U_i, S_j, s)$*: In this query, ζ reutrns the session key Key between U_i and S_j in session s.

7) *$Corrupt(ID_i, onefactor)$*:After receiving this query, ζ ask $\prod_U^s$ to send the corresponding password PW_i or the secret parameters in smart card SC_i. If $ID_i = ID_C$, ζ aborts the game.

8) *$Corrupt(S_j)$*: After receiving this query, ζ returns the private key of the server S_j.

According to above queries, if A can successful pass user authentication, it means that A has successful forged a authentication message $\{T, X, CT\}$ and sends it to ζ, where $CT = (ID_i||ID_{sj}||N_{ui}||st) \oplus h_1(x \cdot PK_{sj})$,$st = x + HID_iK_{ui}$. Based on the forking lemma [58], A has successful forged another authentication message $\{T, X, CT'\}$ via repeat the simulation with a difficult value of h_4. Thus, we gets the below two equations.

$$st = x + HID_iK_{ui} \tag{1}$$

$$st' = x + HID_i'K_{ui} \tag{2}$$

Based on equations (1) and (2), we get the following equations

$$st - st' = (HID_i - HID_i')K_{ui} \tag{3}$$

ζ computes $(st - st')(HID_i - HID_i')^{-1}$ as the answer of DL problem. The probability of it is analyzed below.

We assume that ϵ is the non-negligible probability of A forges a legal authentication message and ρ is the probability of ζ winning the DL problem when A has failed to forge the user authentication message. Thus, the probability of ζ

winning the DL problem may be reduced to the following value similar to that of reference [50].

$$Pr_{win}[DL] = \frac{1}{q_s} \cdot (\epsilon + (1-\epsilon) \cdot \rho) = \frac{\epsilon + (q_s - \epsilon) \cdot \rho}{q_s} \tag{4}$$

where q_s denote the number of Send query. Based on the above analysis, $Pr_{win}[DL]$ is non-negligible and ζ can win the DL problem with non-negligible. Obviously, it is a contradictory assumption. Therefore, there is no polynomial adversary can forge a legitimate user's authentication message with a non-negligible probability.

Lemma 2: (Secure Server Authentication): In our proposed scheme, if h_0, h_1, h_2, h_3, h_4 and the message authentication code (MAC) are ideal random functions, and $\prod_U^s$ has been accepted, then there is no polynomial adversary against the proposed scheme who can forge a legal server authentication message with a non-negligible probability.

Proof. We assume that the adversary A can forge a legal server authentication message with a non-negligible probability. Then there is a PPT Turing machine ζ who can win the underlying game of MAC (Game-MAC) without knowing the secret session key Key with a non-negligible probability by employing A.

The Game-MAC has two participants: a challenger and a MAC oracle $\prod_{MAC}$ which has the secret key Key. The challenger can ask $\prod_{MAC}$ for the MAC value of any message as many times as he/she wants. Let $Pr_{win}[MAC]$ is the probability that the challenger won the game. The game is described as the following three steps:

- The challenger sends two difficult messages m_0 and m_1 to the MAC oracle $\prod_{MAC}$.
- The oracle chooses a random bit $b \in \{0, 1\}$. If $b = 1$, the oracle returns $MAC(Key, m_0)$ to A, otherwise $MAC(Key, m_1)$ is returned.
- The challenger guesses the value of b'. If $b' = b$, it means that the challenger wins the game.

To win the Game-MAC, ζ must simulate an environment of our proposed scheme which is indistinguishable from the real proposed scheme to the adversary A. Hence, ζ should answer all oracle queries issued by A. Firstly, ζ setups all system parameters except challenger ID_{sc}'s private key SK_{sc}. ζ answers the Hi query, $Execute - Oracle$ query and $Reveal - Oracle$ query as he does in the proof of Lemma 1. Then, ζ answers A's queries as follows:

1) $ExtractUi - Oracle$: In this query ζ maintains a list L_U with initialized empty. ζ checks if a tuple $(ID_i, P_{ui}, K_{ui}, N_{ui})$ exists in L_U. If it exists, ζ returns ID_i and N_{ui} to A. Otherwise, ζ generates a random number as the revocation status value N_{ui}, selects a random number $K_{ui} \in Z_q^*$, and calculates the public key $P_{ui} = K_{ui} \cdot P$. ζ stores the tuple $(ID_i, P_{ui}, K_{ui}, N_{ui})$ into L_U and returns ID_i and N_{ui} to A.
2) $ExtractSj - Oracle$: In this query, ζ maintains a list L_S with initialized empty. ζ checks if a tuple $(ID_{sj}, PK_{sj}, SK_{sj})$ exists in L_S. If it exists, ζ returns ID_{sj} to A. Otherwise, ζ operates as follows:.
   - If $ID_{sj} = ID_{sc}$, ζ sets $SK_{sj} = \perp$, and asks the server oracle $\prod_S^s$ to get ID_{sj}'s public key PK_{sj}, stores the tuple $(ID_{sj}, PK_{sj}, SK_{sj})$ into L_S and returns ID_{sj} to A.
   - If $ID_{sj} \neq ID_{sc}$, ζ generates a random number SK_{sj}, calculates $PK_{sj} = SK_{sj} \cdot P$, stores the tuple $(ID_{sj}, PK_{sj}, SK_{sj})$ into L_S and returns ID_{sj} to A.
3) $Send - Oracle(U_i, s, S_j, M)$: In this query, A sends the first message M_1 to ζ, ζ operates according to the specification of the proposed scheme and returns M_2 to A. After receiving M_2 from A, ζ sends the result of user authentication messages according to M_1 and M_2 and asking $\prod_{MAC}$ in order to verify the MAC value.
4) $Send - Oracle(S_j, s, U_i, M)$: After receiving this query, ζ sends the first message M_1 as the protocol specified using the user's private key to A. If $ID_{sj} = ID_0$, ζ aborts the game.
5) $Corrupt(ID_i, onefactor)$:After receiving this query, ζ asks $\prod_U^s$ to send the corresponding password PW_i or the secret parameters in smart card SC_i.
6) $Corrupt(S_j)$: After receiving this query, ζ checks whether the equation $ID_{sj} = ID_0$ holds. If not, ζ returns the private key of the server S_j. Otherwise, ζ aborts the game.

According to above queries, if A can successful pass server authentication, it means that A has successful forged a authentication message $\{Y, V\}$ and sends it to ζ, where $V = MAC(Key, ID_{sj}||ID_i||Y||X)$. Upon receiving $\{Y, V\}$, ζ sends $m_0 = \{ID_{sj}||ID_i||Y||X\}$ and a random $m_1 = Rn$ to the $\prod_{MAC}$. $\prod_{MAC}$ returns $MAC(Key, m_b)$ to ζ. Then ζ can checks whether the value of b is 0 or 1 by using the $\{Y, V\}$ send from A. We assume that ϵ is the non-negligible probability of A forges a legal server authentication message. Thus, the probability of ζ winning the Game-MAC may be reduced to the following value similar to that of reference [50].

$$\begin{aligned} &Pr_{win}[MAC] \\ &\quad = \frac{1}{q_s} \cdot (\epsilon + (1-\epsilon) \cdot \frac{1}{2}) + \frac{q_s - 1}{q_s} \cdot \frac{1}{2}) - \frac{1}{2}) \\ &\quad = \frac{\epsilon}{2q_s} \end{aligned} \tag{5}$$

Based on the above analysis, $Pr_{win}[MAC]$ is non-negligible and ζ can win the Game-MAC with non-negligible. Obviously, it is a contradictory assumption. Therefore, there is no polynomial adversary can forge a legitimate server's authentication message with a non-negligible probability.

Theorem 1: Our proposed scheme is secure protocol, if: (A)$\prod_U^s$ and $\prod_S^s$ have been accepted; (B)h_0, h_1, h_2, h_3, h_4, MAC are ideal random functions; (C) the DL problem is hard.

Proof: Based on Lemma 1 and Lemma 2, we can know that there is no polynomial adversary can forge a legal user or server if MAC is ideal random function and the DL problem is hard. Besides, since $\prod_U^s$ has been accepted, it can ensure that there is a peer ($\prod_S^s$) session of the scheme that has derived precisely the same key. According to Definition 2, the proposed scheme is a secure protocol.

B. FURTHER SECURITY ANALYSIS OF THE PROPOSED SCHEME

1) MUTUAL AUTHENTICATION

According to Theorem 1, we can conclude that there is no polynomial adversary can forge a legal user or server if DL problem is hard and MAC is an ideal random function. Therefore, the user and the server can successfully authenticate each other.

2) USER ANONYMITY AND UN-TRACEABILITY

To protect user's real identity, our proposed scheme encrypt the identity ID_i using the $h_1(x \cdot PK_{sj})$. Besides, the value of $h_1(x \cdot PK_{sj})$ changes at every session due to the fresh of x. Anyone who does not know x or the server's private key SK_{sj} can not know the value of $h_1(x \cdot PK_{sj})$. Therefore, our proposed scheme can provide user anonymity and untraceability.

3) TWO-FACTOR SECURITY

Obviously, the adversary cannot forge a legitimate user when he only knows the user's password. On the other hand, when the smart card is lost or stolen by the adversary A. We assume that A can obtain the secret parameters in the smart card. A still cannot guess the correct password, because there exist $|D_{PW}/1024|$ candidates of the password, where $|D_{PW}|$ is the space of password. This method is called 'fuzzy verifier' [48], [51], which prevents the adversary from obtaining the exacting correct password. Therefore, our proposed scheme can provide two-factor security.

4) RESISTANCE TO WRONG PASSWORD LOGIN/UPDATE ATTACK

In the proposed scheme, the password verification information $V = h_3(h_2(K_{ui}||N_{ui}||C_i))$ is stored in the smart card, which is designed to check the correctness of password. If the user inputs wrong password PW_i', the verification data V and $V' = h_3(h_2(F_i \oplus h_0(ID_i||PW_i'||b_i)||h_0(ID_i||PW_i'||b_i)))$ will not be equal. Therefore, our proposed scheme can quickly detect unauthorized login and password update.

5) USER REVOCATION RE-REGISTRATION

In the proposed scheme, the identities and public keys of users are maintained in the blockchain. Once the smart card is lost or stolen, the user can revoke his/her account, update the revocation status and re-register with a new public key. Due to the revocation status value is recorded in the blockchain, a malicious adversary cannot access the multi-server system using the old public key. In addition, if an adversary wants re-register with the same identity of U_i, he/she must forge a signature $Sig(K_{ui}', ID_i||T_3||REV_{ui})$ or some personal information. However, we assume that the signature function is unforgeable against adaptive chosen message attack. Similarly, if a semi-trusted server wants to add a fake revocation transaction into blockchain, he/she also must forge a signature. But he/she cannot. Therefore, the revocation and invalid re-registration will be checked.

6) KNOWN SESSION KEY SECURITY

In our proposed scheme, the value $X = x \cdot P$ and $Y = y \cdot P$ are fresh and different at every session. If the adversary got the session keys in previous sessions, he/she could not compute the current session key without knowing the value of x or y. Therefore, our scheme can provide known session key security.

7) PERFECT FORWARD SECURITY

In our scheme, the value $X = x \cdot P$ and $Y = y \cdot P$ are fresh and different at every session. If the adversary has obtained the private keys of the user and the server, he/she still cannot compute the session key $Key = h_3(ID_i||ID_{sj}||X||Y||st||y \cdot X)$ or $Key = h_3(ID_i||ID_{sj}||X||Y||st||x \cdot Y)$ without the value x or y in previous sessions. Therefore, the proposed scheme can provide perfect forward security.

8) RESISTANCE TO USER IMPERSONATION ATTACK

In our scheme, in order to impersonate U_i, the adversary has to forge a valid message T, X, CT. However, Lemma 1 shows that it is infeasible due to the DL problem is hard. Therefore, our proposed scheme can resist against user impersonation attack.

9) RESISTANCE TO SERVER SPOOFING ATTACK

Theorem 1 shows that no polynomial adversary can forge a legitimate user's or a server's authentication message without the private key of them. In our scheme, the server only has his own private key and does not know other servers' or users' private key. Therefore, he cannot spoof any users to other servers.

10) RESISTANCE TO REPLY ATTACK

Our scheme uses the challenge-response mechanism and timestamp mechanism to prevent the replay attack. The random number x and y is fresh and different at every session and the timestamp is used in the first message. Therefore, when the user and the server accept each other, it must be the current session, not the previous session. So, our proposed scheme can avoid the replay attack.

11) RESISTANCE TO MAN-IN-THE-MIDDLE ATTACK

In our scheme, the message transmitted is protected by $h_1(x \cdot PK_{sj})$, anyone without x or SK_{sj} can not forge legal authentication message. Therefore, our scheme can resist the man-in-the-middle attack.

C. SECURITY COMPARISONS

In this section, we compare security features of our proposed scheme with the prior related schemes [2]–[5]. The results of the comparison are listed in Table 3. From Table 3, we can see that Odelu et al.'s scheme and our proposed scheme are only two schemes who can provide user revocation and re-registration. However, Odelu et al. scheme requires RC to participate in each user authentication phase, which may make RC being a bottleneck of security. Furthermore, our scheme is the only one which is able to resist against various

TABLE 3. Security comparisons between our proposed scheme and other related schemes.

Security features	He [2]	Odelu [3]	He [4]	He [5]	Ours
Mutual authentication	√	√	√	√	√
User anonymity and untraceability	√	√	√	√	√
Multi-factor security	×	×	×	×	√
Resistance to wrong password login/update attack	×	√	×	×	√
User revocation re-registration	×	√	×	×	√
Known session key security	√	√	√	√	√
Perfect forward security	√	√	√	√	√
Resistance to user impersonation attack	√	√	√	√	√
Resistance to server spoofing attack	√	√	√	√	√
Resistance to reply attack	√	√	√	√	√
Resistance to man-in-the-middle attack	√	√	√	√	√

known attacks and fulfill the desirable security features. Therefore, our proposed scheme has better security than previously related schemes.

VI. COMPARISONS

This section first compares the computational costs and communication overheads of our proposed scheme with other related schemes such as He et al.'s scheme [2], [4], [5] and Odelu et al.'s scheme [3]. Because the initialization phase, registration phase, password update phase and user revocation and re-registration phase are not used frequently, we only compare the mutual authentication phase. Then, we will compare the qualitative property of our blockchain-based approach with the traditional registration center-based approach. In order to measure the effectiveness of our proposed scheme, we present the comparison results in different tables.

A. COMPUTATION ANALYSIS

For efficiency analysis, we compare the computation cost of our proposed scheme with the prior related schemes [2]–[5]. Almost all of the operations in our scheme and prior related schemes have appeared in He et al.'s scheme [5]. According to [59], one MAC operation is about as fast as two hash operations in software implementation. In addition, we assume that the running time in RC is as fast as one in the server. As shown in Table 4, we continue to follow the running time of all operations in their scheme. To facilitate analysis, we use the following notations and their running time to measure the computation cost.

(1) T_{tmp}: The execution time of map-to-point hash function;
(2) T_{bp}: The execution time of bilinear paring operation;
(3) T_{pm}: The execution time of point multiplication operation in G;
(4) T_{pa}: The execution time of point addition operation in G;
(5) T_{sig}: The execution time of signature operation in G;

TABLE 4. Running time of operations(millisecond).

	The user	The server
T_{tmp}	33.582	5.493
T_{bp}	32.713	5.427
T_{pm}	13.405	2.165
T_{pa}	0.081	0.013
T_{sig}	13.405	2.165
T_{ver}	26.81	4.33
T_{exp}	2.249	0.339
T_h	0.056	0.007
T_{MAC}	0.112	0.014

TABLE 5. Computation comparisons between our proposed scheme and other related schemes.

Scheme	User	Server	RC	Total cost
He [2]	$2T_{pm}+T_{pa}+2T_{exp}+8T_h \approx 31.837ms$	$T_{bp}+4T_{exp}+5T_h \approx 6.804ms$	\	$38.641ms$
Odelu [3]	$3T_{pm}+7T_h \approx 40.607ms$	$2T_{pm}+6T_h \approx 4.372ms$	$T_{pm}+11T_h \approx 2.242ms$	47.221
He [4]	$3T_{pm}+7T_h \approx 40.607ms$	$3T_{pm}+5T_h \approx 6.53ms$	$2T_{pm}+9T_h \approx 4.393ms$	$51.53ms$
He [5]	$T_{mtp}+3T_{pm}+2T_{exp}+4T_h \approx 78.519ms$	$2T_{bp}+2T_{pa}+2T_{exp}+5T_h \approx 11.766ms$	\	90.285
Ours	$3T_{pm}+T_{sig}+6T_h+T_{MAC} \approx 54.068ms$	$3T_{pm}+T_{ver}+3T_h+T_{MAC} \approx 10.86ms$	\	$64.928ms$

(6) T_{ver}: The execution time of verification operation in G;
(7) T_{exp}: The execution time of exponentiation operation;
(8) T_h: The execution time of general hash function.
(9) T_{MAC}: The execution time of MAC function.

The results of computation cost comparisons are summarized in Table 5. From Table 5, we can see that the computational efficiency of He et al.'s scheme [2] is the most efficient, while they use the heavy bilinear pairings operations and the security of this scheme is based on exponentiation operation. Although the computational efficiency of Odelu et al.'s scheme [3] and He et al.'s scheme [4] are more efficient than our scheme, they achieve at the price of frequent authentication interaction with an online trusted third party. The computational efficiency of our scheme is not the most efficient. But, our scheme provides more security functions.

B. COMMUNICATION ANALYSIS

In this section, we compare communication cost of our proposed scheme with the recent related schemes [2]–[5]. To achieve convincing comparisons, we assume that the bit length of the hash output, the number of block, the identity, the random number, the block size of symmetric encryption/decryption and the timestamp T are 160, 32, 32, 128, 128 and 32 bits, the bit length of the elliptic curve point and exponentiation are 160 and 1024 bits, respectively. Furthermore, we assume that the bit length of signature messages

TABLE 6. Communication comparisons between our proposed scheme and other related schemes.

Scheme	Rounds of message exchange	Number of bits required
He [2]	3	3072 bits
Odelu [3]	5	2944 bits
He [4]	5	3520 bits
He [5]	4	3296 bits
Ours	2	1088 bits

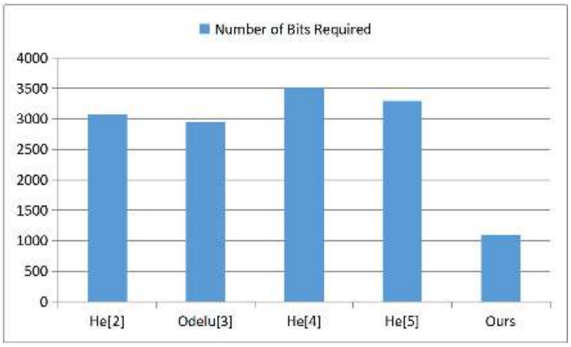


FIGURE 6. Communication comparisons.

is 320 bits [59]. The results of communication efficiency comparisons are summarized in Table 6.

In the proposed scheme, the first messages $\{T, X, CT\}$ require $(32 + 320 + (32 + 32 + 32 + 160)) = 608$ bits, and the second messages $\{Y, V\}$ require $320 + 160 = 480$ bits. Adding the two values, the total communication cost in the authentication phase of our scheme is 1568 bits. Similarly, the total communication cost of the other related schemes can be computed in Table 6.

From comparison in Table 6 and Fig. 6, we conclude that our proposed scheme requires the least rounds of message exchange. Furthermore, the proposed scheme is the most efficient in communication overhead.

C. QUALITATIVE COMPARISONS

The analysis of qualitative property includes single registration, using online RC, resistance to single-point failure, search times and storage. In Table 7, we compare the qualitative property of our blockchain-based approach with the traditional registration center-based (RC-based) approach. Here, we divide RC-based approach into two categories, namely no online RC-Based, and online RC-based, according to whether with the help of online RC in traditional RC-based approach.

The qualitative property of single registration represents whether users register only once. Obviously, the traditional RC-based approach enables users to register once. In our blockchain-based approach, if a user wants to access a server, he/she requires registration only once with any one of the servers in the multi-server system. After the user's information like the identity and public key have been recorded in the blockchain, the user can access the multi-server system. Besides, the traditional RC-based approach belongs to the centralized administration. All new users have to register with the only RC. Our blockchain-based approach can avoid it. A new user can select the closest server in the multi-server system to register, which may be more suitable for practical application.

The using online RC denotes whether the authentication phase between the user and the server needs the help of online RC. According to the above definition, no online RC-Based approach has not online RC in the authentication phase, while online RC-Based approach needs. For our blockchain-based approach, when the server authenticates a user, he/she only verifies the user's signature through searching for the public key in the blockchain. Generally, the blockchain is stored in the own side of the server, there's no need for a trust third party to take part in.

As already stated earlier in this document, the traditional RC-based approach has the problem of single-point failure. All users' data, including users' identities, public keys, possible secret parameters, blacklist, etc., are stored in the single RC. If the single RC attacked or suffered from natural disasters, the whole data will be in danger. To address this issue, we introduce blockchain technology into the multi-server authentication scheme. In our proposed blockchain-based scheme, users' data are recorded in the blockchain, which is decentralized stored in every server in the multi-server system. Once registered on the blockchain, users' data can not be unforgeability.

In practice, according to previous no online RC schemes, like [10], the server has to search for the blacklist to check whether the corresponding user is revoked in the authentication phase. Meanwhile, in previous online RC schemes, like [3], the server has to search for the user information table to check the revocation status. Similarly, in our proposed scheme, the server must search for the blockchain to check user's revocation status. It is obvious that all of multi-server authentication schemes which have considered user revocation have to search for the revocation status. The efficiency of search operations is determined by the length of blacklist, table or blockchain. In general, the blacklist includes all the revoked users, the user information table contains all registered users, and our blockchain involves all registered and revoked users. We let L_{rev}, L_{reg} and L_{bc} denote the length of blacklist, user information table and blockchain, respectively. Since the same user can revoke multiple times, $L_{bc} \geq L_{rev} + L_{reg}$. Obviously, the search efficiency of our blockchain-based is the lowest approach.

The qualitative property of storage means what the server-side stores to achieve user revocation. As analyzed above, the server stores blacklist to check whether the corresponding user is revoked for no online RC schemes. For online RC schemes, since the online RC participates in every authentication phase, the server can query revocation status from RC. So the server does not need to store any user information or revoke information. In our proposed scheme, the server must store the whole blockchain to check user's revocation status. Obviously, the storage cost of our blockchain-based approach is the highest.

From comparison in Table 7, it can be concluded that our blockchain-based scheme solves the problem of single-

TABLE 7. Qualitative comparisons between our blockchain-based approach and RC-based approach.

Qualitative property	No online RC-Based	Online RC-based	Blockchain-based
Single registration	Yes	Yes	Yes
Using online RC	No	Yes	No
Resistance to single-point failure	No	No	Yes
Search times	L_{rev}	L_{reg}	L_{bc}
Storage	The blacklist	Nothing	The blockchain

point failure at the price of storage and search efficiency. In practice, there may be some applications suitable for our blockchain-based approach. For example, in mobile cloud computing environment [5], [11], the service provider has the capability in powerful computing and massive data storage. Thus, it pay more attention to security and privacy. Our blockchain-based multi-server authentication scheme may be more suit for such environment.

VII. CONCLUSION

In this paper, we propose a blockchain-based privacy-awareness authentication scheme with efficient revocation for the multi-server system, which provides various security requirements like mutual authentication, user anonymity, perfect forward security. Besides, in comparison with recently related schemes, the proposed scheme solve the problem of a single registration center. The security analysis demonstrates that our scheme is secure the random oracle model. Performance analysis shows that the proposed scheme has higher communication efficient, which may be suitable to deploy in practice for the multi-server system.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the Associate Editor for providing constructive and generous feedback.

REFERENCES

[1] A. Irshad, M. Sher, S. Ashraf, M. S. Faisal, and M. U. Hassan, "Cryptanalysis for secure and efficient smart-card-based remote user authentication scheme for multi-server environment," *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 686, Jul. 2015.

[2] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2052–2064, Sep. 2016.

[3] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[4] 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.

[5] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Jun. 2018.

[6] L. Xiong, D. Peng, T. Peng, H. Liang, and Z. Liu, "A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks," *Sensors*, vol. 17, no. 11, p. 2681, 2017.

[7] D. Fett, R. Küsters, and G. Schmitz, "SPRESSO: A secure, privacy-respecting single sign-on system for the Web," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Denver, CO, USA, Oct. 2015, pp. 1358–1369.

[8] Q. Bo, H. Jikun, W. Qin, L. Xizhao, L. Bin, and S. Wenchang, "Cecoin: A decentralized PKI mitigating mitm attacks," *Future Gener. Comput. Syst.*, to be published.

[9] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, Oct. 2008, pp. 417–426.

[10] Y.-M. Tseng, S.-S. Huang, T.-T. Tsai, and J.-H. Ke, "List-free ID-based mutual authentication and key agreement protocol for multiserver architectures," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 102–112, Jan./Mar. 2016.

[11] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Gener. Comput. Syst.*, vol. 68, pp. 74–88, Mar. 2017.

[12] L.-H. Li, I.-C. Lin, and M.-S. 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.

[13] W.-S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 251–255, Feb. 2004.

[14] C.-C. Chang and J.-S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proc. Int. Conf. Cyberworlds*, Nov. 2004, pp. 417–422.

[15] W.-J. Tsaur, C.-C. Wu, and W.-B. Lee, "A smart card-based remote scheme for password authentication in multi-server Internet services," *Comput. Standards Interfaces*, vol. 27, no. 1, pp. 39–51, 2004.

[16] J.-L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Comput. Security*, vol. 27, nos. 3–4, pp. 115–121, 2008.

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

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

[19] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *J. Netw. Comput. Appl.*, vol. 34, no. 2, pp. 609–618, 2011.

[20] C.-C. Lee, T.-H. Lin, and R.-X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 13863–13870, 2011.

[21] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Gener. Comput. Syst.*, vol. 84, pp. 239–251, Jul. 2018.

[22] F. Wang, G. Xu, and G. Xu, "A provably secure anonymous biometrics-based authentication scheme for wireless sensor networks using chaotic map," *IEEE Access*, vol. 7, pp. 101596–101608, 2019.

[23] 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. Australas. Conf. Inf. Secur. Privacy*. Springer, 2005, pp. 494–505.

[24] Y.-H. Chuang and Y.-M. Tseng, "Towards generalized ID-based user authentication for mobile multi-server environment," *Int. J. Commun. Syst.*, vol. 25, no. 4, pp. 447–460, 2012.

[25] Y.-P. Liao and C.-M. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generat. Comput. Syst.*, vol. 29, no. 3, pp. 886–900, 2013.

[26] W.-B. Hsieh and J.-S. Leu, "An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," *J. Supercomput.*, vol. 70, no. 1, pp. 133–148, 2014.

[27] L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services," in *Proc. KSII Trans. Internet Inf. Syst. (TIIS)*, vol. 11, no. 12, pp. 6169–6187, 2017.

[28] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[29] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.

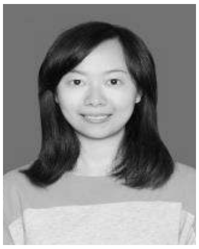
[30] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[31] H. A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of namecoin and lessons for decentralized namespace design," in *Proc. WEIS*, 2015, pp. 1–21.

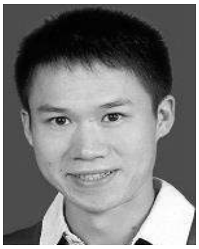
[32] *Litecoin.org*. Accessed: 2014. [Online]. Available: https://www.litecoin.org/
[33] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2016, pp. 142–157.
[34] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," 2015, *arXiv:1505.06895*. [Online]. Available: https://arxiv.org/abs/1505.06895
[35] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Proc. Annu. Int. Cryptol. Conf.* Nieuwpoort, Curaçao: Springer, 2017, pp. 357–388.
[36] A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov, "A provably secure proof-of-stake blockchain protocol," *IACR Cryptol. ePrint Arch.*, vol. 2016, p. 889, Sep. 2016.
[37] D. Larimer, "Delegated proof-of-stake (DPOS)," Bitshare, Blacksburg, VA, USA, White Paper, 2014.
[38] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Gener. Comput. Syst.*, vol. 96, pp. 185–195, Jul. 2019.
[39] C. Fromknecht, D. Velicanu, and S. Yakoubov, "A decentralized public key infrastructure with identity retention," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 803, Nov. 2014.
[40] C. Fromknecht, D. Velicanu, and S. Yakoubov, "CertCoin: A Name-Coin based decentralized authentication system," Class Project, Tech. Rep. 6.857, 2014.
[41] D. Fisher, "Final report on DigiNotar hack shows total compromise of CA servers," Threatpost, 2012. [Online]. Available: https://threatpost.com/final-report-diginotar-hack-shows-totalcompromise-ca-servers-103112/77170/
[42] L. Axon, "Privacy-awareness in blockchain-based PKI," CDT Tech. Paper Ser., 2015. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:f8377b69-599b-4cae-8df0-f0cded53e63b
[43] L. Axon and M. Goldsmith, "PB-PKI: A privacy-aware blockchain-based PKI," in *Proc. 14th Int. Joint Conf. e-Bus. Telecommun. (SCITEPRESS)*, 2017, pp. 1–8.
[44] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 410–426.
[45] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/548/469, doi: 10.5210/fm.v2i9.548.
[46] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," White Paper, 2014, vol. 3, p. 37.
[47] M. K. Franklin and M. K. Reiter, "Fair exchange with a semi-trusted third party (extended abstract)," in *Proc. 4th ACM Conf. Comput. Commun. Secur.*, Zürich, Switzerland, Apr. 1997, pp. 1–5.
[48] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 2039–2042, Jun. 2018.
[49] 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.
[50] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2302–2313, Dec. 2014.
[51] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.
[52] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[53] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.
[54] M. Swan, *Blockchain: Blueprint for a New Economy*. Newton, MA, USA: O'Reilly Media, 2015, pp. 445–470.
[55] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. 13th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, 1993, pp. 232–249.
[56] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EURO-CRYPT)*, Bruges, Belgium, 2000, pp. 139–155.
[57] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EU-ROCRYPT)*, Innsbruck, Austria, 2001, pp. 453–474.
[58] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.
[59] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Hoboken, NJ, USA: Wiley, 2007.

LING XIONG received the M.S. and Ph.D. degrees from the School of Information Science and Technology, Southwest Jiaotong University (SWJTU), Chengdu, China. She is currently an Associate Professor Fellow with the School of Computer and Software Engineering, Xihua University. She is also doing her Postdoctoral Research with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu. Her research interests include the security and privacy in the Internet of Things and blockchain.

FAGEN LI received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2007. From 2008 to 2009, he was a Postdoctoral Fellow with Future University Hakodate, Japan, which is supported by the Japan Society for the Promotion of Science (JSPS). He was a Research Fellow with the Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan, from 2010 to 2012. He is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. He has published more than 70 articles in the international journals and conferences. His current research interests include cryptography and network security.



SHENGKE ZENG received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. She is currently an Associate Professor Fellow with the School of Computer and Software Engineering, Xihua University. She is also doing her postdoctoral research with the School of Computer Science and Engineering, UESTC. Her research interests include cryptography and network security.



TU PENG is currently an Associate Professor Fellow with the School of Software, Beijing Institute of Technology. His current research interests include software reliability, fault localization, and cryptographic protocol.

ZHICAI LIU received the Ph.D. degree from the School of Information Science and Technology, Southwest Jiaotong University (SWJTU), Chengdu, China. He is currently an Associate Professor Fellow with the School of Computer and Software Engineering, Xihua University. His current research interests include the formal analysis of cryptographic protocol and intrusion detection.