Hindawi
Security and Communication Networks
Volume 2020, Article ID 8820271, 9 pages
https://doi.org/10.1155/2020/8820271

Research Article

Identity-Based Identification Scheme without Trusted Party against Concurrent Attacks

Fei Tang,[1,2] **Jiali Bao**,[1] **Yonghong Huang**,[2] **Dong Huang**,[3] **and Fuqun Wang**[4,5]

[1] *College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2] *School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[3] *Chongqing Vocational and Technical University of Mechatronics, Chongqing 402760, China*
[4] *Department of Mathematics, Hangzhou Normal University, Hangzhou 311121, China*
[5] *Westone Cryptologic Research Center, Beijing 100071, China*

Correspondence should be addressed to Fei Tang; tangfei@cqupt.edu.cn

Received 3 August 2020; Revised 3 September 2020; Accepted 21 September 2020; Published 17 October 2020

Academic Editor: Chunming Zhang

Identification schemes support that a prover who holding a secret key to prove itself to any verifier who holding the corresponding public key. In traditional identity-based identification schemes, there is a key generation center to generate all users' secret keys. This means that the key generation center knows all users' secret key, which brings the key escrow problem. To resolve this problem, in this work, we define the model of identity-based identification without a trusted party. Then, we propose a multi-authority identity-based identification scheme based on bilinear pairing. Furthermore, we prove the security of the proposed scheme in the random oracle model against impersonation under passive and concurrent attacks. Finally, we give an application of the proposed identity-based identification scheme to blockchain.

1. Introduction

In identification schemes, the user, playing the role of a prover, can identity itself to any verifier in a protocol in which the verifier begins by holding only the corresponding public key. One of the purposes of identification is to promote access control to resources, when an access privilege is linked to a particular identity.

There are a lot of research studies on identification schemes. The fundamental work of identification scheme [1] was proposed by Fiat and Shamir, named FS scheme. The authors described an identification scheme in which any user can prove its identity to other users. They combined zero-knowledge interactive proofs with identity-based schemes. The key of FS scheme is to assume that there is a trusted center, such as computer center, government, and credit card company. This center gives smart cards to users after checking their physical identities. The FS scheme is based on the factorization problem. Feige et al. [2] proposed another identification scheme, named FFS scheme, which is also based on the factorization problem. Okamoto [3] presented a three-move interactive identification scheme and proved that the scheme has the same security as the discrete logarithm problem. Schnorr's scheme [4] is one of the famous identification schemes. The GQ scheme which was proposed by Guillou and Quisquater [5] is based on the RSA-inversion problem. The formal proof of security for GQ and Schnorr schemes was realized by Bellare and Palacio [6]. They provided a proof for GQ scheme based on RSA-inversion assumption and a proof for Schnorr scheme based on one more discrete logarithm (OMDL) assumption. These two schemes are provably secure against impersonation under active and concurrent attacks. Girault [7] gave a modification of Schnorr's identification scheme, in which each user can select his own secret key but the center can not get it from the public key. Kim and Kim [8] proposed a new identification scheme based on bilinear Diffie-Hellman problem, which is secure against passive and active attacks.

In traditional identification schemes, we need a certificate authority (CA) to authenticate prover's public key in the setting of public key infrastructure (PKI). Shamir [9] introduced the notion of identity-based cryptography (IBC). The purpose of IBC is to simplify the management of certificates in PKI. Shamir pointed out that the key generation center (KGC) generates the corresponding secret key with the public identity and sends it to the user when he first joins in the system. Each user has a unique and meaningful identity as the public key and thus avoids the complicated certificate management problem. Then, Boneh and Franklin proposed an identity-based encryption (IBE) scheme [10], which is based on bilinear pairing. Since then, a large number of identity-based identification schemes have been proposed by using bilinear pairings.

The formal definition of identity-based identification (IBI) scheme was introduced by Kurosawa and Heng [11]. They constructed a transformation from any standard digital signature scheme to an IBI scheme. Then, in [12], they proposed two IBI schemes, one of which is provably secure against impersonation passive attacks, and the other is provably secure against impersonation active and concurrent attacks. The security model of IBI [11, 13] can be divided into three types, called security against impersonation under passive attacks, active attacks, and concurrent attacks, respectively. Then, Chin et al. [14] presented a provably secure IBI scheme in the standard model. The scheme of [14] is secure against impersonation under active and concurrent attacks based on one more computational Diffie-Hellman assumption. Barapatre and Rangan [15] proposed a general framework of IBI based on the identity-based key encapsulation mechanism. The scheme of [15] is secure against impersonation under active and concurrent attacks based on the q-bilinear Diffie-Hellman inversion assumption.

It is well known that, the IBI schemes suffer the key escrow problem, which means that we need a trusted KGC to generate all users' key. In order to solve this problem, in this work, we consider the IBI scheme without a trusted party. The main contributions of this work can be summarized as follows:

(1) We give the formal definition of IBI scheme in the multi-authority setting. In our definition, there are n authorities. The generation of users' secret key needs at least t authorities.

(2) We construct an IBI scheme with multiple authorities based on the BLS signature scheme [16]. The security of the proposed scheme is provably against impersonation under passive and concurrent attacks in the random oracle model.

(3) We consider the applications of the proposed multi-authority IBI scheme. We show that the scheme can be used to identification in blockchain.

The rest of this paper is organized as follows. In Section 2, we give the definitions of bilinear pairing and complexity assumptions. We also present the definition and security models of the IBI scheme in Section 2. Section 3 presents the details of IBI scheme. In Section 4, we prove the security of the proposed scheme. In Section 5, we describe the applications of the multi-authority IBI scheme in blockchain. Finally, we make a conclusion about this paper in section 6.

2. Preliminaries

In this section, we describe the relevant definitions and security models.

2.1. Bilinear Map and Complexity Assumptions. In the construction of our identity-based identification scheme, we use bilinear pairing as the basic tool. Therefore, we briefly introduce the concept of bilinear pairing.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups, where $\mathbb{G}$ is generated by an element g, i.e., $\mathbb{G} = \langle g \rangle$. Groups $\mathbb{G}$ and $\mathbb{G}_T$ have same prime order p. We say that $(e\colon \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T)$ is an admissible bilinear pairing if it satisfies the following properties:

(1) Bilinearity: $e(g^a, g^b) = e(g, g)^{\mathrm{ab}}$ for all $(a, b \in \mathbb{Z}_p)$.

(2) Nondegeneracy: there exists $(g^c, g^d \in \mathbb{G})$, for $(c, d \in \mathbb{Z}_p)$, such that $e(g^c, g^d) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ represents the identity element of the group $\mathbb{G}_T$.

(3) Computability: there is an efficient algorithm to compute $e(g^a, g^b)$ for all $(a, b \in \mathbb{Z}_p)$.

The security of our scheme relies on the following two difficult problems: Computational Diffie-Hellman (CDH) Problem and One More Discrete Logarithm (OMDL) Problem.

Definition 1 (CDH). Given (g, g^a, g^b) for some $(a, b \in \mathbb{Z}_p)$, it is hard to compute g^{ab}.

Definition 2 (OMDL) (see [17]). The definition of the OMDL problem is defined by the following experiment $\mathrm{Exp}^{\mathrm{omdl}}(k)$.

(i) Training: A polynomial-time adversary $\mathscr{A}$ makes n queries to the challenge oracle $\mathscr{B}(\cdot)$ and m queries to the Discrete-Logarithm (DL) oracle $\mathscr{DL}_{p,g}(\cdot)$. Let $(s_1, s_2, \ldots, s_n) \longleftarrow \mathscr{A}^{\mathscr{B}(\cdot), \mathscr{DL}_{p,g}(\cdot)}(p, g, e, \mathbb{G}, \mathbb{G}_T)$.

(ii) Output: If $(g^{s_1} = h_1) \wedge \cdots \wedge (g^{s_n} = h_n)$, where $(h_1, \ldots, h_n)$ are random points in $\mathbb{G}$ output by the challenge oracle $\mathscr{C}(\cdot)$, and $n < m$, where n denotes the number of queries to the DL oracle, then return 1. Otherwise, return 0.

We define the advantage of adversary $\mathscr{A}$ as $(\mathrm{Adv}_{\mathscr{A}}^{\mathrm{omdl}}(k) = \Pr[\mathrm{Exp}^{\mathrm{omdl}}(k)] = 1)$. We say that OMDL problem is hard if $\mathrm{Adv}_{\mathscr{A}}^{\mathrm{omdl}}(k)$ is negligible in k for any polynomial-time adversary.

2.2. Definition of Multi-authority IBI. An identity-based identification (IBI) scheme $\mathscr{IBI} = (\mathscr{S}, \mathscr{K}, \mathscr{P}, \mathscr{V})$ is specified by four probabilistic polynomial-time algorithms, called Setup, Key-generation, Proving, and Verification, respectively. On input security parameter k, $\mathscr{S}$ returns system public parameters and the master secret key. $\mathscr{K}$ is executed by the key generation center to generate a secret key corresponding to a given public identity. $\mathscr{P}$ and $\mathscr{V}$ are interactive algorithms that implement the prover and verifier. We call $(\mathscr{P}, \mathscr{V})$ an identification protocol.

As far as we know, there is no IBI scheme in the setting of multiple authorities. The standard IBI schemes have a key generation center to produce all users' secret key. Therefore, it is well known that identity-based cryptographic schemes have the key escrow problem. This work defines the notion of IBI scheme with multiple authorities. In our scheme, there has one more algorithm, Authority Setup, to generate all authorities' master secret keys. The notion of IBI scheme with multiple authorities is consists of the following algorithms:

(i) System-setup: This algorithm takes as input the security parameter k and outputs the system public parameter params.

(ii) Authority-setup: The authority setup algorithm is interactively executed by all authorities. On input the system public parameter params and identities $P_1, \ldots, P_n$, output their master secret keys $\mathrm{SK}_1, \ldots, \mathrm{SK}_n$.

(iii) Key-generation: User id makes queries to at least t authorities, $P_{i_1}, \ldots, P_{i_t}$, where $i_j \in [1, n]$ for key generation. Each authority P_{i_j} takes as inputs the system public parameter params, master secret key SK_{i_j}, and user's identity id and outputs user its partial key $\mathrm{psk}_{\mathrm{id}, i_j}$. Finally, user id can compute the secret key $\mathrm{sk}_{\mathrm{id}}$ by itself.

(iv) Identification: $\mathscr{P}$ receives as inputs (params), id, and $(\mathrm{sk}_{\mathrm{id}})$ and $\mathscr{V}$ receives as inputs (params), and (id)), where $\mathrm{sk}_{\mathrm{id}}$ is the secret key corresponding to the public identity id. After an interactive execution of $(\mathscr{P}, \mathscr{V})$, $\mathscr{V}$ outputs 1 (accept) or 0 (reject).

(v) Correctness: A legitimate $\mathscr{P}$ should always be accepted, i.e., $\langle \mathscr{P}(\mathscr{K}(\mathrm{msk}, \mathrm{id})), \mathscr{V} \rangle (\mathrm{params}, \mathrm{id}) \longrightarrow 1$.

2.3. Security Models. The accepted framework of security concepts for identification schemes was proposed by Feige et al. [2]. Then, the security definition for IBI scheme was presented in [11, 13]. This is an extension of the framework of [2]; that is, the three concepts of security for standard identification schemes are extended to IBI. Usually, we consider adversary goals, adversary capabilities or attacks. The adversary goal is impersonation that if the adversary interacts with the verifier playing the role of prover with identity id^* and can persuade the verifier to accept with a nonnegligible probability. To achieve this goal, the adversary can carry out various attacks. We consider three kinds of attacks, namely, passive attacks [2], active attacks [2], and concurrent attacks [6]. These attacks should take place and complete before the impersonation attempt.

Passive attacks are the weakest one of the above three kinds of attacks for IBI schemes. In passive attacks, the adversary does not interact with the prover. The adversary just eavesdrops and obtains a transcript of a conversation between the prover and verifier. The definition of passive attacks of IBI schemes is defined by the following game which is executed by an adversary $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$ and a challenger $\mathscr{C}$.

Definition 3 (Security against Impersonation under Passive Attacks). Let $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$ be an impersonation adversary with passive attacks (imp-pa).

(i) System-setup: The challenger $\mathscr{C}$ runs the system setup algorithm on input a security parameter k to generate system public parameters params. Then, $\mathscr{C}$ returns params to $\mathscr{A}$.

(ii) Authority-setup: The challenger $\mathscr{C}$ runs the authority setup algorithm to generate master secret keys $\mathrm{SK}_1, \mathrm{SK}_2, \ldots, \mathrm{SK}_n$ for all authorities $P_1, P_2, \ldots, P_n$.

(iii) Queries: $\mathscr{A}$ can issues some queries as follows:

  (1) Master secret key queries: $\mathscr{A}$ issues a request for some authorities P_i for their master secret key. For such a request, $\mathscr{C}$ transmits SK_i to $\mathscr{A}$.

  (2) Key generation queries: $\mathscr{A}$ issues some key generation queries id_i. $\mathscr{C}$ then returns the corresponding private key $\mathrm{sk}_{\mathrm{id}_i}$ as the answer.

  (3) Transcript queries: $\mathscr{A}$ can issue some transcript queries on id. In passive attacks, $\mathscr{C}$ returns the transcripts T which denotes the conversations between the valid prover id and other verifiers.

(iv) Challenge: $\mathscr{A}$ chooses a challenge identity id^*. Then, $\mathscr{A}$ plays the role of a cheating prover, trying to convince any verifier.

We define that adversary $\mathscr{A}$ succeeds in impersonating if it can make the verifier accepts. The advantage of an imp-pa adversary $\mathscr{A}$ denoted by $\mathrm{ADV}^{\mathrm{imp-pa}}_{\mathscr{IBI},\mathscr{A}}(k)$. We say that IBI scheme is secure against impersonation under passive attacks if $\mathrm{Adv}^{\mathrm{imp-pa}}_{\mathscr{IBI},\mathscr{A}}(k)$ is negligible in k for any imp-pa adversary.

Different from passive attacks, in the active and concurrent attacks, the adversary first plays the role of the cheating verifier, interacting with the honest prover multiple times, trying to extract some useful information. Then it plays role of cheating prover, interacting with the honest verifier, trying to persuade the honest verifier to accept. It is easy to see that the security notions of active and concurrent attacks are stronger than the notion of passive attacks. Generally, we pursue stronger security notion for crytographic schemes, such as [18, 19].

Active attacks are a special case of concurrent attacks. In the active attacks, the next round of attack is carried out after one attack is completed, that is, the interaction is one by one. In the concurrent attacks, however, the adversary can interact with multiple different prover "replicas" concurrently. The replicas all have the same secret key but are initialized with independent coins and maintain their own state. Apparently, security against impersonation under concurrent attack implies security against impersonation under active attack.

Difinition 4 (Security against Impersonation under Concurrent Attacks). An impersonation under concurrent attacks (imp-ca) adversary $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$ is a pair of randomized polynomial-time algorithms, which denotes the cheating verifier and the cheating prover, respectively. The definition of the concurrent attacks of IBI schemes is defined by the following game which is played by a concurrent adversary and challenger $\mathscr{C}$.

(i) System-setup: The challenger $\mathscr{C}$ runs the system setup algorithm on input k to generate system public parameters params. Then, $\mathscr{C}$ sends params to different replicas of prover $\mathscr{P}$ and adversary $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$.

(ii) Authority-setup: The challenger $\mathscr{C}$ runs the authority setup algorithm to generate master secret keys $\mathrm{SK}_1, \mathrm{SK}_2, \ldots, \mathrm{SK}_n$ for all authorities $P_1, P_2, \ldots, P_n$.

(iii) Queries: $\mathscr{A}$ can issues some queries as follows:

(1) Master secret key queries: $\mathscr{A}$ issues a request for some authorities P_i for their master secret key. For such a request, $\mathscr{C}$ transmits SK_i to $\mathscr{A}$.

(2) Key generation queries: $\mathscr{A}$ issues some key generation queries id_i. $\mathscr{C}$ then returns the corresponding private key $\mathrm{sk}_{\mathrm{id}_i}$ as the answer.

(3) Identification training: $\mathscr{A}$ first plays the role of a cheating verifier to execute the identification protocols with the honest prover id. In concurrent attacks, the adversary $\mathscr{A}$ can issue the identification protocol at any time regardless of whether the last protocol is end or not. The difference between concurrent attack and active attack is that the active adversary only can issue a new identification protocol after the end of the last protocol. We denote the transcript of i-th protocol as T_i.

(iv) Challenge: Finally, adversary plays the role of a cheating prover $\widehat{\mathscr{P}}$ to execute the identification protocol with a valid verifier $\mathscr{V}$ to try to convince that he is the valid prover.

We define that adversary $\mathscr{A}$ succeeds in impersonating if it can make the verifier accepts. The advantage of an imp-ca adversary $\mathscr{A}$ denoted by $\mathrm{Adv}^{\mathrm{imp-ca}}_{\mathscr{IBI},\mathscr{A}}(k)$. We say that IBI scheme is secure against impersonation under concurrent attacks if $\mathrm{Adv}^{\mathrm{imp-ca}}_{\mathscr{IBI},\mathscr{A}}(k)$ is negligible in k for any imp-ca adversary.

3. The Proposed Scheme

In this section, we give our multi-authority IBI scheme without a trusted party. Generally speaking, in traditional IBI schemes, there is a trusted party for the generation and distribution of user secret keys. To address the problem of no trusted party, we utilize distributed key generation (DKG) protocol to generate user secret keys. DKG was proposed by Gennaro et al. [20]. The core idea of DKG is (t, n) threshold secret sharing. The concept of secret sharing was introduced by Shamir [21]. Secret sharing is used to share a secret among a group of participants, each of whom has partial information about secret. (t, n) threshold secret sharing means that at least t participate among n participants can reconstructed the secret value.

In the DKG protocol, the participants jointly choose and generate a random secret share s. Each participant P_i chooses a random share s_i, and then a random secret share s can be recovered by at least t participants. At the end of the protocol, the public key can be defined as $y = g^s$. There is no trusted party, who owns the secret value s in the secret sharing scheme. The secret value s can only be reconstructed by the cooperation of at least t participants.

The construction of our scheme refers to two article by Lin et al. [22] and Tang et al. [23]. Lin et al. proposed a threshold multi-authority attribute-based encryption scheme. In their scheme, they use (t, n) threshold secret sharing to get the system secret key a_0. Each authority only has the share a_{i0} about secret a_0. Therefore, the system secret key a_0 is unknown to any authority. Tang et al. proposed an efficient multi-authority authentication scheme for electronic health records system based on blockchain.

3.1. Construction.

3.1. Construction. The construction of the scheme is outlined below:

(i) System-setup: Given the security parameter k as input, generates prime p randomly to establish the system parameters. First of all, it chooses two multiplication cycles $\mathbb{G}$ and $\mathbb{G}_T$ with some prime order p, and a bilinear map $(e\colon \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T)$. Let g be a generator of the group $\mathbb{G}$. Next, it chooses a cryptographic hash function $H\colon \{0,1\}^* \longrightarrow \mathbb{G}$. The system parameters are params $= \{p, g, e, \mathbb{G}, \mathbb{G}_T, H, n, t\}$, where n is the number of authorities in the system, and t is the threshold value which denotes the number of authorities to generate secret key for users.

(ii) Authority-setup: In this algorithm, all authorities take public parameters params and their identities $P_1, \ldots, P_n$ as inputs and establish their master secret keys $\mathrm{SK}_1, \ldots, \mathrm{SK}_n$. It consists of the following two phases:

(a) Phase 1 (generation of the master secret key): Each authority generates the public key and private key, as well as the master public key of the system.

(1) Each authority P_i selects at random a polynomial $\mathscr{F}_i(x) \in \mathbb{Z}_p^*$ of degree $(t-1)$:

$$\mathscr{F}_i(x) = \sum_{j=0}^{t-1} a_{ij} x^j. \quad (1)$$

(2) P_i calculates $(A_{ik} = g^{a_{ik}})$ for $(k = 0, 1, \ldots, t-1)$ and then broadcasts A_{ik}.

(3) P_i computes secret value $y_{\mathrm{ij}} = \mathscr{F}_i(P_j)$ for $(j = 1, 2, \ldots, n)$, and then sends y_{ij} secretly to authority P_j for $j \neq i$.

(4) P_j verifies the equation $g^{y_{ij}} = \prod_{k=0}^{t-1}(A_{ik})^{P_j^k}$ holds or not. If it holds, the secret sharing from P_i is valid. Otherwise, P_j broadcasts a complaint against P_i.

(5) If authority P_i is complained, then it needs to broadcast values y_{ij} that satisfy the equation. If the disclosed y_{ij} still does not match, P_i has to keep proving itself to be honest until the equation is true.

(6) P_j computes its own private key $\mathrm{SK}_j = \sum_{i=1}^{n} y_{\mathrm{ij}}$ and calculates its own public key $\mathrm{PK}_j = g^{\mathrm{SK}_j}$. The master secret key s can be recovered by any t values in $\mathrm{PK}_1, \ldots, \mathrm{PK}_n$.

(b) Phase 2 (generation of master public key): According to the above phase, each authority has broadcasted values $\mathrm{PK}_i = g^{\mathrm{SK}_i}$ for $(i = 1, 2, \ldots, n)$ which can verified publicly. Therefore, the master public key can be computed as

$$y = \prod_{i=1}^{n} \mathrm{PK}_i^{\prod_{j=1, j\neq i}^{t}\left(P_j/P_j - P_i\right)}. \quad (2)$$

After the above two phases, each authority adds parameters y and $(P_i, \mathrm{PK}_i)_{i=1}^{n}$ to the parameters params: $= \{p, g, e, \mathbb{G}, \mathbb{G}_T, H, y, n, t, (P_i, \mathrm{PK}_i)_{i=1}^{n}\}$.

(iii) Key-generation: User id_i makes key-generation request to at least t authorities. Then, the authority generates the corresponding partial secret key and sends it to the user. After receiving the partial secret key, the user can verify its correctness using the public key of the corresponding authority. Finally, user id_i computes his secret key $\mathrm{sk}_{\mathrm{id}_i}$ by himself.

(1) Phase 1 (generation of partial secret key): Each authority P_j computes a value $\mathrm{psk}_{\mathrm{id}_i, j} = H(\mathrm{id}_i)^{\mathrm{SK}_j}$ and secretly transmits it to user id_i.

(2) Phase 2 (verification of partial secret key): After receiving the partial secret key $\mathrm{psk}_{\mathrm{id}_i, j}$ from authority P_j, the user id_i verifies the equation $e(\mathrm{psk}_{\mathrm{id}_i, j}, g) = e(H(\mathrm{id}_i), \mathrm{PK}_j)$ holds or not. If it holds, then the partial secret key is correct. Otherwise, the user exposes the partial secret key and requests other authorities to authenticate it. The authority P_j needs to retransmit the correct value to satisfies the equation.

(3) Phase 3 (generation of secret key): After receiving all partial secret keys, the user id_i computes his own secret key as

$$\mathrm{sk}_{\mathrm{id}_i} = H(\mathrm{id}_i)^s = \prod_{j=1}^{t} \mathrm{psk}_{\mathrm{id}_i, j}^{\prod_{k=1, k\neq j}^{t}\left(P_k/P_k - P_j\right)}. \quad (3)$$

(iv) Identification: We consider two types of identification protocols which corresponding to the passive attack and concurrent (or active) attack, respectively.

(a) Identification protocol against passive attacks:

(1) The prover id_i selects $(r \in \mathbb{Z}_p^*)$ randomly, computes $U = H(\mathrm{id}_i)^r \in \mathbb{G}$, and sends U to verifier.

(2) The verifier chooses $(c \in \mathbb{Z}_p^*)$ randomly and sends it to prover id_i.

(3) The prover id_i computes $(V = \mathrm{sk}_{\mathrm{id}_i}^{r+c} \in \mathbb{G})$ and returns it to verifier.

(4) The verifier checks $(e(V, g) = e(U, y) \cdot e(H(\mathrm{id}_i)^c, y))$ holds or not. If it holds, outputs accept; otherwise, outputs reject.

(b) Identification protocol against active and concurrent attacks:

(1) The prover id_i blinds the secret key $\mathrm{sk}_{\mathrm{id}_i}$. Let $\widetilde{\mathrm{sk}}_{\mathrm{id}_i} = \mathrm{sk}_{\mathrm{id}_i}^z$, where $(z \in \mathbb{Z}_p^*)$ is the blinding factor.

(2) The prover id_i randomly selects an integer $(r \in \mathbb{Z}_p^*)$, computes $(X = e(H(\mathrm{id}_i), y)^r)$, and sends X and $\widetilde{\mathrm{sk}}_{\mathrm{id}_i}$ to verifier.

(3) The verifier chooses a random integer $(c \in \mathbb{Z}_p^*)$ and sends it to prover id_i.

(4) The prover id_i computes $t = r + cz \pmod{p}$ and sends t to verifier.

(5) The verifier checks $(e(H(\mathrm{id}_i), y)^t = X \cdot e(\widetilde{\mathrm{sk}}_{\mathrm{id}_i}, g)^c)$ holds or not. If it holds, outputs accept; otherwise, outputs reject.

3.2. Correctness. The correctness of the identification protocol against passive attacks can be verified by the following equation:

$$\begin{aligned} e(V, g) &= e\left(\mathrm{sk}_{\mathrm{id}_i}^{r+c}, g\right) \\ &= e\left(H(\mathrm{id}_i)^{s\cdot(r+c)}, g\right) \\ &= e\left(H(\mathrm{id}_i)^{r+c}, g^s\right) \\ &= e\left(H(\mathrm{id}_i)^r, y\right) \cdot e\left(H(\mathrm{id}_i)^c, y\right) \\ &= e(U, y) \cdot e\left(H(\mathrm{id}_i)^c, y\right). \end{aligned} \quad (4)$$

The correctness of the identification protocol against concurrent attacks can be verified by the following equation:

$$\begin{aligned} e(H(\mathrm{id}_i), y)^t &= e(H(\mathrm{id}_i), y)^{r+cz} \\ &= e(H(\mathrm{id}_i), y)^r \cdot e(H(\mathrm{id}_i), y)^{cz} \\ &= X \cdot e\left(H(\mathrm{id}_i)^z, g^s\right)^c \\ &= X \cdot e\left(H(\mathrm{id}_i)^{zs}, g\right)^c \\ &= X \cdot e\left(\widetilde{\mathrm{sk}}_{\mathrm{id}_i}, g\right)^c. \end{aligned} \quad (5)$$

4. Security Proofs

In this section, we prove the security of the proposed multi-authority IBI scheme.

As said above, the proposed scheme is based on the distributed key generation technique [20] and a centralized IBI scheme. It seems that the security of the scheme directly holds based on the securities of the two schemes. It is not true because in the security proof of IBI scheme we need to embed the challenge instance to a fixed element y which is one of the public parameters. However, the value y which is generated by the distributed key generation technique [20] is randomly in the beginning.

To resolve this problem, we use the proof framework of [23] which introduced the approach of hybrid games for this kind of schemes. The core technique of [23] is that define three games. The first game corresponds to the honest execution of the security proof. Then, in the second game, we set the master key as $y\colon = g^{as}$ where a is the exponent of the CDH or OMDL instance and s is the master secret key randomly generated by all authorities, respectively. No one knows a and s. In the last game, the challenger plays the role of all authorities, and thus it knows the value s. Then, we can prove that the advantage of any probabilistic polynomial time (PPT) adversary in the first game is close to the another two games. Hence, if we can prove the advantage of any PPT adversary in the last game which corresponds to the proof of centralized IBI scheme is negligible, then we can obtain the security result that the advantage of any PPT adversary of the multi-authority IBI scheme is also negligible. Therefore, in this work, we only prove the security of the centralized IBI scheme. Please refer to [23] for details of the proof technique which describes the security from centralized scheme to the multi-authority setting.

Theorem 1. *The proposed multi-authority IBI scheme is secure against impersonation under passive attack in the random oracle model assuming that the CDH problem is hard.*

Proof. Let $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$ be a polynomial-time imp-pa impersonator that tries to break the IBI scheme. Let $\mathscr{C}$ be a challenger that tries to break the BLS signature scheme under chosen message attack. $\mathscr{C}$ takes as input k, generates public parameters $(p, g, e, \mathbb{G}, \mathbb{G}_T, H)$, where $(H\colon \{0,1\}^* \longrightarrow \mathbb{G})$ is a hash function modeled as a random oracle. $\mathscr{C}$ chooses $(x \in \mathbb{Z}_p^*)$, computes $(y = g^x \in \mathbb{G})$, and then gives system public parameters params $= (p, g, e, \mathbb{G}, \mathbb{G}_T, H, y)$ to adversary $\mathscr{A}$.

If $\mathscr{A}$ makes a key generation query on id_i. $\mathscr{C}$ then returns the corresponding private key $\mathrm{sk}_{\mathrm{id}_i}$ as the answer. If $\mathscr{A}$ makes a transcript query on id_j. Then $\mathscr{C}$ chooses $(c_j \in \mathbb{Z}_p^*)$, $(V_j \in \mathbb{G})$ randomly and computes U_j such that $e(V_j, g) = e(U_j, y) \cdot e(H(\mathrm{id}_j)^{c_j}, y)$. $\mathscr{C}$ then gives (U_j, c_j, V_j) to $\mathscr{A}$ as the transcript. Finally, $\mathscr{A}$ chooses a challenge identity id^*.

Now, $\mathscr{A}$ plays the role as the cheating prover and interacts with challenger $\mathscr{C}$. $\mathscr{A}$ can still issues some key generation queries and transcript queries in this phase, with the restriction that the query on the challenge identity id^* is not allowed. $\mathscr{C}$ runs $\mathscr{A}$ to get the response U. After receiving U, $\mathscr{C}$ selects $(c \in \mathbb{Z}_p^*)$ randomly, runs $\mathscr{A}$ to get its response V and verifies the equation $e(V, g) = e(U, y) \cdot e(H(\mathrm{id}^*)^c, y)$ holds or not. If the equation holds, $\mathscr{C}$ runs $\mathscr{A}$ again with the same state but with different challenge value $(c' \in \mathbb{Z}_p^*)$, obtains its response V', and verifies the equation $e(V', g) = e(U, y) \cdot e(H(\mathrm{id}^*)^{c'}, y)$ hold or not. If the equation holds, $\mathscr{C}$ outputs $(V/V')^{(c-c')^{-1}}$ as a forgery. Since we have $(V = \mathrm{sk}_{\mathrm{id}^*}^{r+c})$ and $V' = \mathrm{sk}_{\mathrm{id}^*}^{(r+c')}$. Thus, $\mathrm{sk}_{\mathrm{id}^*} = (V/V')^{(c-c')^{-1}}$ is a valid signature on id^*. □

Theorem 2. *The proposed multi-authority IBI scheme is secure against impersonation under concurrent attack in random oracle model assuming that the OMDL problem is hard.*

Proof. Let $\mathscr{A} = (\widehat{\mathscr{V}}, \widehat{\mathscr{P}})$ be a polynomial-time imp-ca impersonator that tries to break the identity-based identification scheme. Let $\mathscr{C}$ be an OMDL challenger. We assume that $\widehat{\mathscr{V}}$ never repeats a request. $\mathscr{C}$ takes as input k and generates public parameters $(p, g, e, \mathbb{G}, \mathbb{G}_T)$. $\mathscr{C}$ chooses $(x \in \mathbb{Z}_p^*)$ and computes $(y = g^x \in \mathbb{G})$ and then outputs params $= (p, g, e, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, y)$ as system public parameters. $\mathscr{C}$ returns params to adversary $\mathscr{A}$.

If $\mathscr{A}$ makes a key generation query on id_i. $\mathscr{C}$ then returns the corresponding private key $\mathrm{sk}_{\mathrm{id}_i}$ as the answer.

Now, $\mathscr{A}$ makes a identification training. First, challenger $\mathscr{C}$ queries its challenge oracle $\mathscr{B}(\cdot)$ to obtain a challenge point $(W_0 = g^{r_0} \in \mathbb{G})$, where $(r_0 \in \mathbb{Z}_p^*)$. $\mathscr{C}$ now chooses an arbitrary identity $(\mathrm{id} \in \{0,1\}^*)$. $H\colon \{0,1\}^* \longrightarrow \mathbb{G}$ is hash function viewed as a random oracle. We set it as follows. $\mathscr{C}$ chooses a random $(l \in \mathbb{Z}_p^*)$ and sets $(H(\mathrm{id}) = g^l \in \mathbb{G})$. If $(\mathrm{id}' \neq \mathrm{id})$, $\mathscr{C}$ chooses a random $(l' \in \mathbb{Z}_p^*)$ and sets $H(\mathrm{id}') = g^{l'} \in \mathbb{G}$. $\mathscr{C}$ next computes $(\widetilde{\mathrm{sk}}_{\mathrm{id}} = W_0^{lx})$ and sends it to adversary $\mathscr{A}$. Since $(W_0 = g^{r_0})$ for random $(r_0 \in \mathbb{Z}_p^*)$, $(\widetilde{\mathrm{sk}}_{\mathrm{id}} = W_0^{lx} = g^{r_0 lx} = H(\mathrm{id})^{xr_0})$ for random $(l, x \in \mathbb{Z}_p^*)$. Now, $\mathscr{C}$ simulates an interaction between $\widehat{V}$ and the prover replicas as follows. A random tape R_i is chosen for prover replicas i. $\mathscr{C}$ then initializes prover replicas i with (params, R_i). $\mathscr{C}$ first queries its challenge oracle $B(\cdot)$ to get the response W_i. $\mathscr{C}$ computes $X_i = e(W_i^{lx}, g)$ and sends this to $\widehat{\mathscr{V}}$. Since $(W_i = g^{r_i})$ for random $(r_i \in \mathbb{Z}_p^*)$, we have $(X_i = e(W_i^{lx}, g) = e(g^{r_i lx}, g) = e(g^{r_i l}, g^x) = e(g^l, g^x)^{r_i} = e(H(\mathrm{id}), y)^{r_i})$. $\widehat{V}$ chooses random $(c_i \in \mathbb{Z}_p^*)$ and returns to $\mathscr{C}$. $\mathscr{C}$ makes the query $W_i W_0^{c_i}$ to its discrete log oracle $\mathrm{DL}_{p,g}(\cdot)$ and get the response t_i. $\mathscr{C}$ sends t_i to $\widehat{\mathscr{V}}$. $\widehat{\mathscr{V}}$ verifies the equation $e(H(\mathrm{id}), y)^{t_i} = X_i \cdot e(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g)^{c_i}$ hold or not. The correctness of the equation is as follows:

$$\begin{aligned} e(H(\mathrm{id}), y)^{t_i} &= e\left(g^l, g^x\right)^{r_i + c_i r_0} \\ &= e\left(g^l, g^x\right)^{r_i} \cdot + e\left(g^l, g^x\right)^{c_i r_0} \\ &= X_i \cdot e\left(g^{lxr_0}, g\right)^{c_i} \\ &= X_i \cdot e\left(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g\right)^{c_i}. \end{aligned} \tag{6}$$

After performing the above simulations, $\widehat{\mathscr{V}}$ outputs some state information and stops interaction. Now, $\mathscr{C}$ attempts to extract the discrete logarithm r_0 of challenge point W_0. Then using this value, $\mathscr{C}$ can further compute the discrete logarithm $r_1, r_2, \ldots, r_n$ of other challenge points $(W_1, W_2, \ldots, W_n)$. To do so, $\mathscr{C}$ runs $\widehat{\mathscr{P}}$ in state St obtaining X, selects a random $(c \in \mathbb{Z}_p^*)$, and runs $\widehat{\mathscr{P}}$ to get its response t. $\mathscr{C}$ then verifies the equation $(e(H(\mathrm{id}), y)^t = X \cdot e(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g)^c)$ holds or not. If the equation holds, $\mathscr{C}$ runs $\widehat{\mathscr{P}}$ again with the same state St but with different challenge value $(c' \in \mathbb{Z}_p^*)$, obtains its response t' and verifies the equation $e(H(\mathrm{id}), y)^{t'} = X \cdot e(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g)^{c'}$ hold or not. If the equation holds, $\mathscr{C}$ computes $((t - t')/(c - c')(\bmod p))$. We show that $((t - t')/(c - c')(\bmod p))$ is the discrete logarithm of W_0. Observing that

$$\begin{aligned} e\left(g^{\mathrm{lx}(t-t'/c-c')}, g\right) &= e\left(g^{\mathrm{lx}(t-t')}, g\right)^{(1/c-c')} \\ &= \left(e\left(g^l, g^x\right)^t e\left(g^l, g^x\right)^{-t'}\right)^{(1/c-c')} \\ &= \left(e(H(\mathrm{id}), y)^t e(H(\mathrm{id}), y)^{-t'}\right)^{c-c'} \\ &= \left(X \cdot e\left(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g\right)^c \left(X \cdot e\left(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g\right)^{c'}\right)^{-1}\right)^{c-c'} \\ &= \left(e\left(\widetilde{\mathrm{sk}}_{\mathrm{id}}, g\right)^{c-c'}\right)^{(1/c-c')} \\ &= e\left(W_0^{\mathrm{lx}}, g\right) \\ &= e\left(g^{r_0 \mathrm{lx}}, g\right). \end{aligned} \tag{7}$$

From the above equation, we obtain $(r_0 = ((t - t')/(c - c'))(\bmod p))$. We now can further compute $(r_i = t_i - c_i r_0 \ (\bmod p))$ for $(i = 1, 2, \ldots, n)$. Finally, $\mathscr{C}$ outputs $(r_1, r_2, \ldots, r_n)$. □

5. Applications

The proposed IBI scheme provides a good solution for scenarios where there is no trusted center, such as blockchain. Hence, in this section, we consider the application of the multi-authority IBI scheme in blockchain.

Blockchain technology was introduced by Nakamoto [24] in Bitcoin. Blockchain as the underlying technology of Bitcoin is essentially a type of distributed ledger. It can avoid the single point of failure. The advantages of blockchain are decentralization, anonymity, trustworthiness, and so on. According to different application scenarios and participants, blockchain can be divided into three categories, including public blockchain, consortium blockchain, and private blockchain [25]. In public blockchain, everyone can read and send transactions and everyone could join in the consensus process. For private blockchain, the node coming from a specific organization can be allowed to enter into the consensus process. The consortium blockchain is between public blockchain and private blockchain. It is a specific blockchain with authorized nodes. The consensus process is controlled by authorized nodes. The consortium blockchain is a community composed of n member organizations, and each member runs a node. Only with the confirmation of (2/3) of the member organizations can each block take effect. At present, many researchers are trying to utilize blockchain in different fields, such as healthcare [26, 27], Internet of things (IoT) [28], and so on. This will make data freedom from ideal to reality, and these data providers will become the buliders and users of blockchain. In order to achieve the security of data sharing and privacy protection and confirm that data usage is legitimate, it is necessary to reach a consensus on the identification to ensure the authenticity of the identity on the chain.

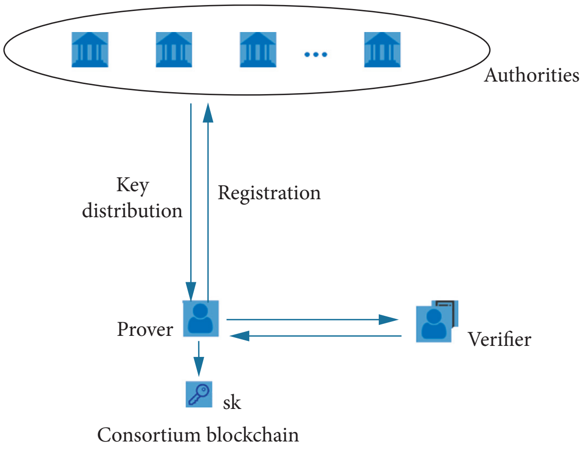

Figure 1: The identification protocol of blockchain based on multi-authority IBI scheme.

Traditional IBI schemes are centralized which have a trusted party to generate and distribute users' key. However, the main feature of blockcahin is decentralization. Traditional IBI schemes are suitable for single authority instead of multiple authorities. There is no trusted party in blockchain. At the same time, we cannot build a trusted party in blockchain. Distrbuted identification is a way to address this problem. In distributed identification, we do not need to rely on the trusted third party for secret key generation and distribution and identity management.

Our multi-authority IBI scheme can provide a good solution for consortium blockchain. We describe the application of our IBI scheme in consortium blockchain. In the consortium blockchain, we can divide nodes into two types, authority and user. The member of consortium blockchain plays the role of the authority, and the user is assumed by other nodes that join the consortium blockchain. The identification protocol for blockchain based on multi-authority IBI scheme is as shown in Figure 1.

(1) System-setup: In the beginning, all n authorities are cooperating to initialize the consortium blockchain system. In this phase, they generate the public parameters according to the security parameters. Meanwhile, all master secret keys can be generated by themselves. Finally, public parameters are published to all users in this system, and the master secret keys are secretly kept by all authorities.

(2) User-registration: When a user wants to join the system, he submits his enrollment request to at least t authorities. Then, the system assign an unique recognizable identity id and corresponding partial secret key $\mathrm{psk}_{\mathrm{id},j}$. Eventually, user verifies the validity of the partial secret key and computes its own secret key $\mathrm{sk}_{\mathrm{id}}$.

(3) Identification: Finally, in some cases, the user needs to prove that he is a legitimate user of this system. Then he can use the identification protocol of the IBI scheme.

6. Conclusion

In this paper, we propose an identity-based identification scheme without trusted party, which is provably secure in the random oracle model. Our scheme takes advantage of distributed key generation to generate the user's secret key. By interacting with at least t authorities, a legal user can generate his/her secret key. Thus, it avoids any one authority being a single-point bottleneck on security. The security analysis results show that our identity-based identification scheme is secure against impersonation under passive and concurrent attacks. Finally, we apply the proposed scheme to the blockchain.

Data Availability

No data were used during the study.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Key Research and Development Program (No. 2017YFB0802300), in part by the National Natural Science Foundation of China (Nos. 61702067, 61972124), in part by the Zhejiang Provincial Natural Science Foundation of China (No. LY19F020019), in part by the key project of science and technology research program of Chongqing Education Commission of China (No. KJZD-K201803701, in part by Chongqing Natural Science Foundation (No. cstc2019jcyj-msxmX0551), and in part by the science and technology research program of Chongqing Municipal Education Commission (No. KJQN201903701).

References

[1] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," *Advances in Cryptology—CRYPTO 1986, LNCS*, vol. 263, pp. 186–194, Springer-Verlag, Berlin, Germany, 1987.

[2] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

[3] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," *Advances in Cryptology-CRYPTO 1992, LNCS*, vol. 740, pp. 31–53, Springer, Berlin, Germany, 1993.

[4] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[5] L. Guillou and J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory," *Advances in Cryptology—EUROCRYPT 1988, LNCS*, vol. 330, pp. 123–128, Springer-Verlag, Berlin, Germany, 1989.

[6] M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks," *Advances in Cryptology—CRYPTO 2002 2002, LNCS*, Springer-Verlag, vol. 2442, pp. 162–177, Berlin, Germany, 2002.

[7] M. Girault, "An identity-based identification scheme based on discrete logarithms modulo a composite number," *Advances in Cryptology—EUROCRYPT 1990, LNCS*, vol. 473, pp. 481–486, Springer, Berlin, Germany, 1991.

[8] M. Kim and K. Kim, "A new identification scheme based on the bilinear Diffie-Hellman problem," in *Proceedings of the 7th Australian Conference on Information Security and Privacy-ACISP 2002*, pp. 362–378, Melbourne, Australia, July 2002.

[9] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology-CRYPTO 1984, LNCS*, vol. 196, pp. 47–53, Springer-Verlag, Berlin, Germany, 1985.

[10] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Advances in Cryptology—CRYPTO 2001, LNCS*, vol. 2139, pp. 213–229, Springer-Verlag, Berlin, Germany, 2001.

[11] K. Kurosawa and S. H. Heng, "From digital signature to ID-based identification/signature," *Public Key Cryptography—PKC 2004, LNCS*, vol. 2947, pp. 248–261, Springer-Verlag, Berlin, Germany, 2004.

[12] K. Kurosawa and S. H. Heng, "Identity-Based identification without random oracles," in *Proceedings of the Computational Science and Its Applications—ICCSA 2005, International Conference, LNCS*, vol. 3481, Springer, Singapore, 2005.

[13] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," *Advances in Cryptology—EUROCRYPT 2004*, Springer-Verlag, vol. 3027, pp. 268–286, Berlin, Germany, 2004.

[14] J. J. Chin, S. H. Heng, and B. M. Goi, "An efficient and provable secure identity-based identification scheme in the standard model," *Public Key Infrastructure—EuroPKI 2008, LNCS*, vol. 5057, pp. 60–73, Springer, Berlin, Germany, 2008.

[15] P. Barapatre and C. P. Rangan, "Identity-based identification schemes from ID-KEMs," *Security, Privacy and Applied Cryptography Engineering—SPACE 2013, LNCS*, vol. 8204, pp. 111–129, Springer, Berlin, Germany, 2013.

[16] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Advances in Cryptology—ASIACRYPT 2001, LNCS*, Vol. 2248, Springer, Berlin, Germany, 2001.

[17] J. Baek, R. Safavi-Naini, and W. Susilo, "Universal designated verifier signature proof (or how to efficiently prove knowledge of a signature)," *Lecture Notes in Computer Science 2005, LNCS*, Springer-Verlag, vol. 3788, pp. 644–661, Berlin, Germany, 2005.

[18] J. Chang, H. Wang, F. Wang, A. Zhang, and Y. Ji, "RKA security for identity-based signature scheme," *IEEE Access*, vol. 8, pp. 17833–17841, 2020.

[19] J. Mo, Z. Hu, and Y. Lin, "Cryptanalysis and Security improvement of two authentication schemes for healthcare systems using wireless medical sensor networks," *Security and Communication Networks*, vol. 2020, Article ID 5047379, 11 pages, 2020.

[20] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Advances in Cryptology—EUROCRYPT 1999, LNCS*, vol. 1592, pp. 295–310, Springer, Berlin, Germany, 1999.

[21] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[22] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Progress in Cryptology—INDOCRYPT 2008, LNCS*, Vol. 5365, Springer, Berlin, Germany, 2008.

[23] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019.

[24] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, BN Publishing, New York City, NY, USA, 2008.

[25] Z. Zheng, S. Xie, H. Dai, and H. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, Honolulu, HI, USA, June 2017.

[26] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[27] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu, "A decentralizing attribute-based signature for healthcare blockchain," in *Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, Hangzhou, China, pp. 1–9, July 2018.

[28] H. Huang, X. Chen, Q. Wu, X. Huang, and J. Shen, "Bitcoin-based fair payments for outsourcing computations of fog devices," *Future Generation Computer Systems*, vol. 78, pp. 850–858, 2018.