Hindawi
Security and Communication Networks
Volume 2022, Article ID 9686049, 15 pages
https://doi.org/10.1155/2022/9686049

Research Article

A Lightweight Authentication Scheme Based on Consortium Blockchain for Cross-Domain IoT

Yujian Zhang,[1,2] **Yuhao Luo**,[2] **Xing Chen**,[2] **Fei Tong**,[1,2,3] **Yuwei Xu**,[1,2,3] **Jun Tao**,[1,2,3] **and Guang Cheng**[1,2,3]

[1]*Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing, China*
[2]*School of Cyber Science and Engineering, Southeast University, Nanjing, China*
[3]*Purple Mountain Laboratories, Nanjing, China*

Correspondence should be addressed to Yujian Zhang; yjzhang@seu.edu.cn

Received 4 September 2021; Revised 18 December 2021; Accepted 21 December 2021; Published 3 January 2022

Academic Editor: Wenjuan Li

Internet of Things (IoT) has been ubiquitous in both industrial and living areas, but also known for its weak security. Being as the first defense line against various cyberattacks, authentication is even more critical to IoT applications. Moreover, there has been a growing demand for cross-domain collaboration, leading to an increasing need for cross-domain authentication. Recently, certificate-based authentication schemes have been extensively studied. However, many of these schemes are not efficient in computation, storage, and communication, which are highly required in IoT. In this paper, we propose a lightweight authentication scheme based on consortium blockchain and design a cryptocurrency-like digital token to build trust. Furthermore, trust lifecycle management is performed by manipulating the amount of tokens. The comprehensive analysis and evaluation demonstrate that the proposed scheme is resistant to various common attacks and more efficient than competitor schemes in terms of storage, communication, and authentication cost.

1. Introduction

With the great power of bridging the physical world to the cyberspace, the Internet of Things (IoT) brings new paradigms for supporting diverse applications, such as smart cities, transportation, manufacturing, agriculture, and healthcare. According to a Statista report, the number of global IoT devices is forecast to triple from 8.74 billion in 2020 to more than 25.4 billion in 2030 [1]. The surging number suggests that IoT technology has been deeply involved in our daily lives. However, while enjoying the significant convenience brought by IoT, we need to be aware that IoT is still facing a bunch of challenges. The most critical one should be IoT security, since most IoT devices are limited in computation, storage, and network capacity; they have weak security and are more vulnerable to cyberattacks [2]. For example, in 2016, Mirai botnet took control of millions of IoT devices and launched an incredibly powerful Distributed Deny-of-Service (DDoS) attack [3]. Moreover, as reported by Gartner, 20% of organizations have experienced at least one IoT attack in the past three years [4]. Hence, there is an urgent demand for defensive strategies to protect IoT applications.

Serving as the first defense line against various attacks, authentication is the most effective way to prevent unauthorized access and other potential threats, which is particularly crucial in IoT scenarios [5]. Specifically, IoT authentication refers to a model for building trust in the identities of IoT devices through an unsecured network, such as Internet. With the wide range of deployment, IoT devices from different domains have an increasing need to communicate with each other for better collaboration, as shown in Figure 1. For example, smart vehicles administered by different local government agencies construct a Vehicular Ad hoc Network (VANET) for intelligent transportation applications. More industrial IoT devices from different factories are organized to work collaboratively to improve the productivity of manufacturing. For security concerns,

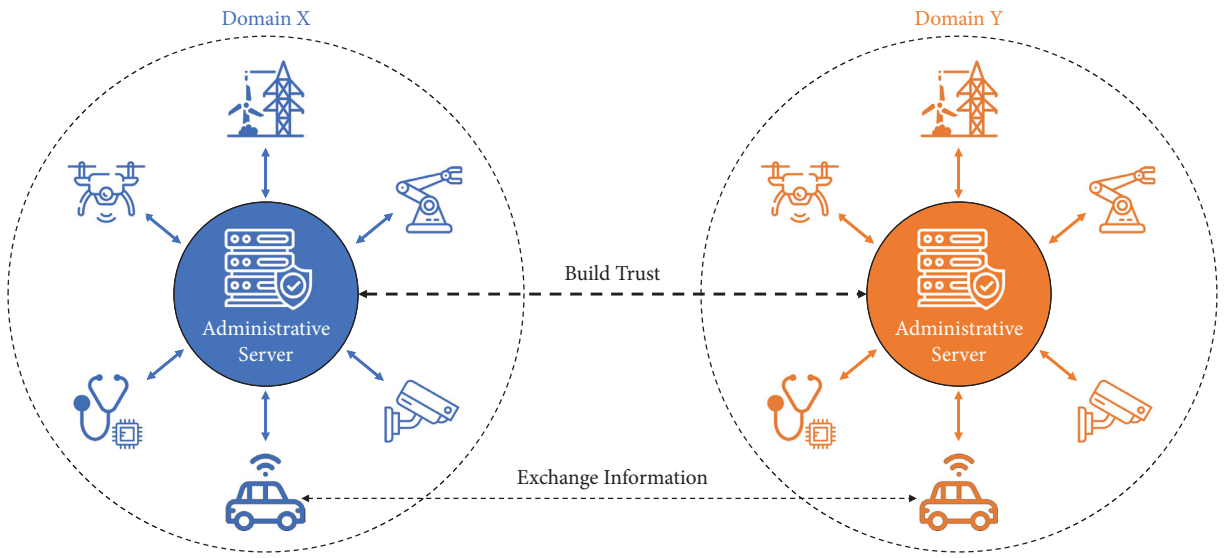

Figure 1: An example scenario of cross-domain IoT collaboration.

none of these domain administrators will allow their devices to be accessed from outside without authentication. However, building trust across different domains is a nontrivial task since it can refer to some sensitive data, which is more likely to be preserved within its own domain.

Existing IoT authentication methods are popularly based on the well-known Public Key Infrastructure (PKI) [6–14], which is built on top of asymmetric cryptography with public-private key pairs and uses certificates to authenticate the identities of individuals, devices, and other entities. Each PKI system depends on a Trusted Third Party (TTP) named Certificate Authority (CA) to provide the root of trust for all certificates. As long as these CAs are not compromised, IoT devices from outside domains can be authenticated by showing certificates signed by their administrative CAs. However, certificate management, including issuance, distribution, and revocation, requires significant efforts [13, 14] and is prone to misoperations [15]. To mitigate the management overhead in PKI-based methods, Identity-Based Cryptography (IBC) uses a publicly known string as the public key [16]. The private key is generated by another TTP, named Key Generation Center (KGC), which introduces a key-escrow problem and limits the mobility of IoT devices. Besides, both PKI-based and IBC-based methods rely on a TTP, which makes them suffer a threat of single-point failure.

As an emerging technology for building trust, blockchain has attracted an increasing interest in IoT authentication. Many current research studies incorporate blockchain into conventional authentication methods to avoid their drawbacks [17–20]. For example, CA transparency can be largely improved when certificate histories are recorded in the distributed public ledger of blockchain [17]. Considering the model of cross-domain authentication, some research studies employ delegated agencies (e.g., IBC [21, 22] and private blockchain [23]) for IoT devices to participate in the blockchain system. However, blockchain-based certificate authentication retains the issues in the PKI method, such as local resource requirements and management overhead. Although delegated agencies can mitigate IoT resource limitations, they re-introduce the risk of single-point failure and complicate the authentication infrastructure. On the contrary, blockchain technology is still under development due to its high latency and energy consumption, especially for public blockchain (e.g., Bitcoin [24] and Ethereum [25]). Therefore, employing blockchain for cross-domain IoT authentication still remains several challenges, such as improving efficiency on computation and storage for cryptographic keys, lightweight authenticity lifecycle management, and transparency of authorities.

Among different types of blockchain, consortium blockchain is a permissioned network, in which each entity needs a prior approval before joining an organization and works collaboratively across different organizations. Motivated by the fact that its structure well matches the model of cross-domain IoT, we propose a lightweight authentication scheme for cross-domain IoT based on consortium blockchain. The main idea is to utilize a digital token, named LiIDCoin (Lightweight IDentity Coin), to represent the trust of the identity of an IoT device or an entity (we will use them interchangeably in later sections). Different from most existing blockchain-based methods, such as CertCoin [18], AuthCoin [19], and CeCoin [20], which leverage blockchain to store public keys, challenge-response messages, and certificates, respectively, LiIDCoin is more like a cryptocurrency equipped with a set of coin operations to realize authentication process and lifecycle management. In this way, costs of computation, communication, and storage for authentication as well as lifecycle management efforts can be significantly reduced, which is highly preferable for IoT applications. This paper is an extend version of work published in [26]. We extend our previous work by adapting the method to IoT scenarios and conducting extensive experiments for detailed evaluations. The main contributions of this paper are summarized as follows.

We propose a framework for cross-domain IoT authentication based on consortium blockchain by exploiting their well-matched architectures and the underlying idea of cryptocurrency to build trust between entities

We design a novel data structure based on the unspent transaction output (UTXO) coupled with a suite of coin operations, including issuance, transfer, query, and revocation, to support authentication and lifecycle management

We perform comprehensive security analysis and implement a prototype on the HyperLedger Fabric (HLF) platform to validate the effectiveness and efficiency of the proposed scheme

The rest of this paper is organized as follows. Section 2 reviews recent literature on IoT authentication mechanisms, while Section 3 presents main preliminaries for the stated problem. Section 4 provides the framework of the proposed scheme, followed with the design details in Section 5. The security analysis and performance evaluation of the proposed scheme are given in Section 6 and Section 7, respectively. Finally, Section 8 concludes the work.

2. Related Work

A majority of studies have been focused on PKI-based methods since it can realize distributed public key authentication as well as offering good scalability [6]. Azees et al. proposed a certificate-based authentication scheme with conditional privacy preservation for VANETs [7]. Vijayakumar et al. utilized short-term certificates to achieve anonymous authentication in IoT-based Wireless Body Area Networks (WBANs) [8]. Li et al. pushed CA functions closer to IoT nodes to minimize attack surfaces and authentication latency for smart living [9]. To accelerate certificate validation, Wang et al. cached and maintained validated certificates in a memory pool made up by all IoT devices [10]. In view of the resource-limited nature, delegated nodes, such as gateways [11] and mobile agents [12], were designated to perform complex functionalities for IoT devices. Ma et al. built a consensus of trust for IoT devices by sharing certain information among distributed nodes [27]. Besides, Li et al. considered the revoked certificates [13] and further proposed certificate update protocols to concern the whole lifecycle [14]. Admittedly, the cryptographic and management overhead of PKI-based methods has never been ignorable. Although Identity-Based Cryptography (IBC) [16] and Certificateless Signature (CLS) [28] remove the certificate management, they necessitate the need for a Key Generation Center (KGC), which limits the mobility of IoT devices and still suffer a threat of single-point failure.

The emergence of blockchain technology with characteristics including unforgeability, redundancy, and transparency has revolutionized conventional authentication. Fromknecht et al. maintained domains and their associated public keys on the blockchain, named CertCoin, to build an alternative PKI [18]. Instead of public key bindings, Leiding *et al.* proposed an AuthCoin scheme to store challenge-response information for validation and authentication of public keys on the blockchain [19]. Qin et al. provided CeCoin services of multicertificate and identity assignment through blockchain transactions [20]. Hammi et al. created secure virtual zones (bubbles) for IoT authentication in between by manipulating bubble settings on a public blockchain [29]. For cross-domain scenarios, Jia et al. utilized IBC to authenticate local IoT entities and authorized them to access cross-domain resources through a threshold cryptographic algorithm on blockchain [21]. Li et al. deployed a private blockchain for each local domain and applied a cross-chain technology to the cross-domain IoT authentication through a public blockchain [23]. Shen et al. proposed a Blockchain-Assisted Secure Authentication (BASA) mechanism that integrated consortium blockchain with IBC to authenticate cross-domain industrial IoT devices for smart manufacturing [22]. Besides, benefiting from the public ledger, blockchain is able to build trust among entities and perform trust management in an open and distributed environment [30].

The above studies provide promising solutions for IoT authentications, but there still remains several challenges. First, the computational and storage efficiency should be improved, especially in solutions which employ complex cryptographic primitives for security and privacy. Second, optimization on authenticity lifecycle management is highly desired, especially in the process of revocation due to its negative impacts on system scalability and efficiency. Third, activities of authorities should be transparent and those of authentication should be accountable. Although blockchain has been introduced for auditing these activities or incorporated into authentication process, in this paper, we attempt to make an in-depth integration by exploring native features of consortium blockchain.

3. Preliminaries

This section presents problem statement, analysis of classical PKI-based solution, and design goals of this work.

3.1. Problem Statement. The problem studied in this paper is illustrated in Figure 2. Authentication is required between two entities, E_X^i and E_Y^j, which belong to domain X and Y, respectively. For the case that E_X^i is authenticated by E_Y^j, E_X^i is called the *clamiant* and E_Y^j is called the *verifier*. In each domain, there is a centralized agency, named Trusted Authority (TA), to play roles of both an administrator and an endorser. As an administrator, a TA validates entity enrollments and maintains their legitimacies. Once accepting an entity, TA is also responsible for endorsing for its identification through various forms, such as certificates. Such a kind of architecture is quite common in practice to facilitate effective management within a domain. Therefore, the problem studied in this paper is "*How to enable entities from different domains to authenticate each other in an IoT-applicable manner?*". In other words, the authentication infrastructure should be lightweight to meet the resource-limited and distributed nature of IoT devices.

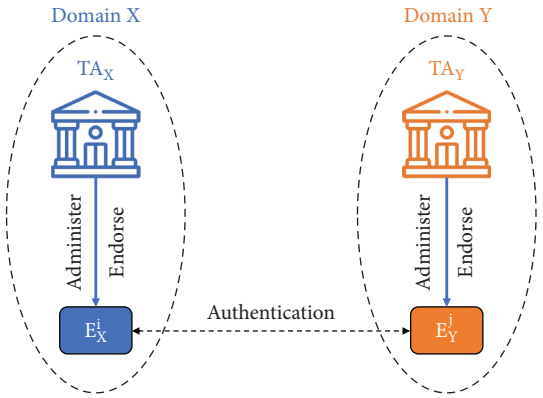

Figure 2: Problem overview of cross-domain IoT authentication.

Most authentication schemes heavily rely on the following cryptographic primitives, which are also assumed to supported by the IoT devices in this paper.

3.1.1. Digital Signature. A digital signature is a way of authenticating a digital data coming from a trusted source, which is typically represented by a tuple of three algorithms (Gen, Sign, Verify). The key generation algorithm $\text{Gen}(1^\lambda) \longrightarrow (\text{pk}, \text{sk})$ takes the security parameter 1^λ as input and outputs a pair of public/private keys (pk, sk). The signing algorithm $\text{Sign}(\text{sk}, m) \longrightarrow \sigma$ uses a private key sk to generate a signature σ on a message m, while the verification algorithm verifies the signature σ on message m through the public key pk, returning 1 if valid or $\perp$ otherwise.

3.1.2. Hash Function. A collision-resistant hash function $H(\cdot)$, defined by H: $\{0,1\}^* \longrightarrow \{0,1\}^l$, takes an arbitrary length string as input and outputs a fixed length (l bits) hash value, which is extremely hard to be generated from another different input. Hence, $H(\cdot)$ is widely used for digesting large data.

Table 1 lists the key notations used in this paper.

3.2. PKI-Based Solution. PKI provides a competitive solution to the stated problem by allowing TA to create, distribute, and revoke digital certificates for entity identification. Figure 3 demonstrates the working mechanism of PKI-based solution, in which E^i_X is authenticated by E^j_Y. First, E^i_X needs to register in TA_X to get a certificate $\text{Cert}_{E^i_X}$ represented by

$$\text{Cert}_{E^i_X} = \text{ID}_{E^i_X} \parallel \text{pk}_{E^i_X} \parallel \text{Date} \parallel \text{Issuer} \parallel \text{Algorithm} \parallel \text{Sig}_{\text{TA}_X}(\cdot), \tag{1}$$

where Date indicates the expiration time, Issuer represents its authority (e.g., TA_X), and Algorithm denotes cryptographic primitives used to generate the certificate. Then, certificate of TA_X, denoted as $\text{Cert}_{\text{TA}_X}$, should be preinstalled in E^j_Y as a trust anchor. The whole authentication process consists of four steps to not only validate the legitimacies of $\text{Cert}_{E^i_X}$ and $\text{Sig}_{E^i_X}$ once sent to E^j_Y in step ① but also confirm its validity through Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP) by steps ② and ③. Finally, step ④ gives the decision. Mutual authentication can be implemented by adding an inverse pass.

Table 1: Key notations.

Notations	Description
E^i_X	The *ith* entity of domain X
TA_X and TA_Y	Trusted authority of domain X and Y
pk_Z and sk_Z	Public key and private key of entity/TA Z
ID_Z and Cert_Z	Identity and certificate of entity/TA Z
$\text{Sig}_Z(\cdot)$	Signature function by entity/TA Z
$H(\cdot)$	Hash function $(0,1)^* \longrightarrow (0,1)^l$
$\parallel$	Concatenate operation
p	A large prime
Z_p	Curve base field, $Z_p = \{0, 1, \ldots, p-1\}$
$E(Z_p)$	Curve equation over Z_p
a, b	Parameters of curve equation, $a, b \in Z_p$
G	Curve base point
n	Curve order

Although PKI provides an effective scheme for cross-domain IoT authentication, it still needs improvement on performance and security. On the one hand, PKI certificate requires resources for cryptographic computation, storage, and communication, which could not be affordable for resource-critical IoT devices and certificate management costs' significant maintenance efforts. On the other hand, a certificate typically binds personal information with the public key, which could violate privacy preservation, and CA transparency still remains as a security issue in IoT scenarios.

3.3. Design Goals. A practical authentication scheme for cross-domain IoT authentication should accommodate the following requirements.

3.3.1. Lightweight. Local resources required to accomplish authentication should be as low as possible for IoT devices. Moreover, relevant management efforts are also expected to be mitigated.

3.3.2. Security. Due to the open nature of IoT environment, the authentication scheme should be resistant against common cyberattacks.

3.3.3. Mobility. With the development of mobile computing and 5G technologies, there exhibits a growing demand for roaming, in which a IoT device needs to access the authentication service across its administrative domain.

4. Overview of the Proposed Scheme

This section presents the framework of the proposed scheme and describes the main components. Design details will be given in the next section.

4.1. Framework. The whole framework, as demonstrated in Figure 4, can be majorly divided into two layers, namely, entity layer and trust layer. The former represents IoT devices, each of which belongs to a specific domain. The latter layer provides a

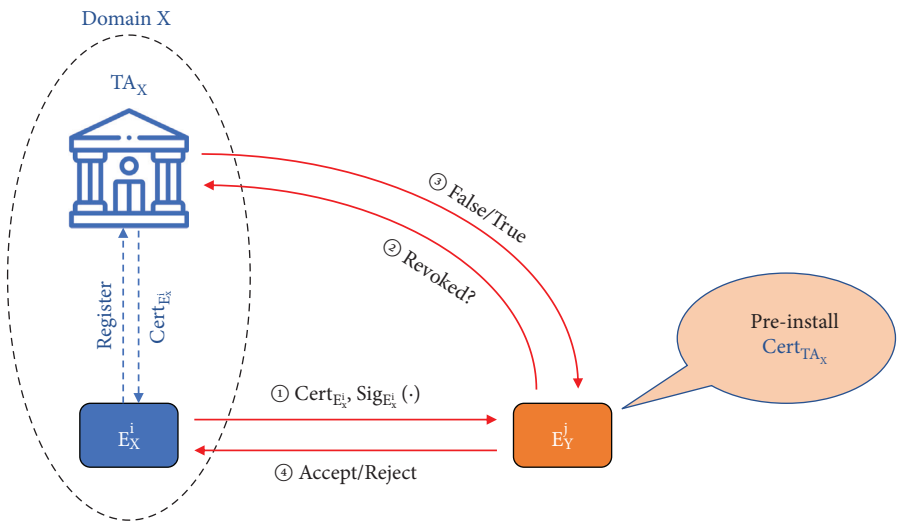

Figure 3: Unilateral authentication of PKI-based solution.

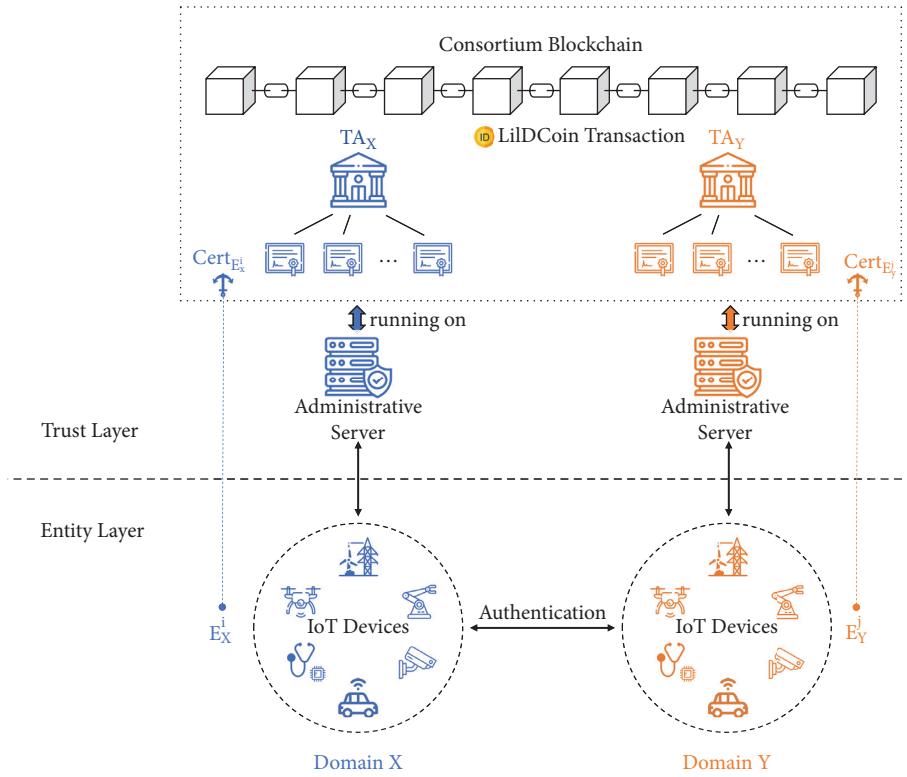

Figure 4: Layered framework of the proposed authentication scheme.

cross-domain authentication service built on top of consortium blockchain. Specifically, domain administrative servers are organized to create a Peer-to-Peer (P2P) network. Based on that, blockchain is deployed to build a consensus on entity states including their legitimacies and authentication activities. As consortium blockchain is introduced in the proposed scheme, a PKI system is incorporated to identify permissioned entities within the blockchain. In this way, each legal entity will be assigned with a domain-specific certificate, representing a trust anchor for the entity in the trust layer. For example, trust anchors of entity E_X^i and E_Y^j in Figure 4 are certificate $\text{Cert}_{E_X^i}$ and $\text{Cert}_{E_Y^j}$, respectively. It is worth mentioning that, although PKI is also employed as the trust facility in our scheme, it will be used internally inside the blockchain thus can largely benefit security enhancement and save management effort. Moreover, instead of explicitly exchanging certificates, the proposed scheme accomplishes a fast and secure identification through a transaction of specific digital token.

4.2. LiIDCoin: A UTXO Data Structure. We develop a digital token for identifying entity, named LiIDCoin, based on the unspent transaction output (UTXO) data structure,

which has been successfully applied in Bitcoin [24]. However, instead of free circulation in Bitcoin, LiIDCoin emphasizes token's specificity and one-time use.

Table 2 lists the key items of LiIDCoin data structure, which is composed of three parts: *basic*, *in*, and *out*. The *basic* is a common part in a blockchain transaction. Tx_id is a unique string to identify a LiIDCoin transaction. Timestamp denotes the submission time of the transaction. Script represents the operation name for generating the current transaction, which can be defined as a reference to a smart contract or a script. Signature is used to prove sender's agreement on the transaction. The *in* part of LiIDCoin describes the source of the token. Sender_id is a unique hash value to represent the address of token sender (e.g., $\mathrm{ID}_{\mathrm{TA}_X}$ or $\mathrm{ID}_{E_X^i}$). Note that such a hash value is generated as a pseudonym to hide the real identity of an entity once enrolled. Type is an entity-specific attribute to restrict tokens to being only used by their original owner, which can be implemented by binding it with some entity-specific information. Furthermore, endorser information should be retained for validating management operations. To this end, the certificate of an entity could be a good candidate. For purposes of effective storage and communication, we use a hash value of certificate. For a specific entity E_X^i, we have

$$\mathrm{Type}_{E_X^i} = H\left(\mathrm{Cert}_{E_X^i}\right), \tag{2}$$

which implies that such tokens can only be spent by E_X^i and are endorsed by its authority TA_X. In the *out* part, Recipient_id and Quantity represent the recipient address (e.g., $\mathrm{ID}_{E_Y^j}$ or $\mathrm{ID}_{\mathrm{TA}_X}$) and the amount of transferred tokens, respectively. For authentication, one token is sufficient to indicate the activity. Note that there can be more than one *out* field to represent multiple token recipients. For example, one token is spent for authentication while others remain in balance by being transferred to the owner itself.

4.3. Trust Layer. In the proposed scheme, the trust layer is the key infrastructure to provide cross-domain authentication service, which is accomplished by consortium blockchain underneath. It is a trusted distributed ledger composed of blocks encapsulating numbers of transactions, which record entity states of token ownership and transfer activities. The trust layer can be further divided into the following three sublayers.

4.3.1. Hardware. Administrative servers of different domains are organized to create the P2P network for blockchain. On the one hand, each server plays the role of a full node in blockchain network to store a complete copy of the ledger as well as running blockchain software (e.g., HLF). On the other hand, as located in its local domain, each administrative server is responsible for domain administration as a TA and handling all regional service requests, including those from roaming entities, to minimize network delays.

Table 2: Key items of the LiIDCoin data structure.

Item	Name	Description
Basic	Tx_id	Transaction ID
	Timestamp	Transaction submission time
	Script	Operation name
	Signature	Signature of sender
In	Sender_id	TA/Entity ID
	Type	Token type (hash of sender's certificate)
Out[..]	Recipient_id	TA/Entity ID
	Quantity	Token amount (1 for authentication)

4.3.2. Blockchain. Blockchain maintains a global distributed ledger of transactions, which carries the entire history of digital tokens. Furthermore, blockchain is responsible for building a consensus on token states among all peering servers. To realize prospective and automatic token state transition, smart contracts that implement authentication service logics are priorly installed on blockchain peers.

4.3.3. PKI. Since the consortium blockchain is permissioned, each entity has to be a member of an organization in the blockchain. Consequently, PKI is deployed internally for on-chain identification, providing features such as entity registration and certificate management. As long as these certificates are never delivered for any external use, security risks and management efforts can be significantly reduced. Thus, each entity can be anchored to a long-term certificate in the system.

4.4. Entity Layer. The entity layer contains diverse IoT devices that have basic capabilities of processing encryption and network communication, such as smart vehicles, drones, and manufacturing robots. Note that some of them (e.g., vehicles and drones) have a vital requirement for mobility, which implies location-fixed delegated nodes or key centers could not be available when such IoT devices move into a foreign domain. Therefore, the authentication scheme should be feasible to address the mobility requirement.

By anchoring to a PKI certificate in the blockchain, each entity essentially has a roamable account for authentication service as long as it can connect to any regional administrative server. The private key for generating the certificate is the root of entity control over all LiIDCoins associated with the corresponding address. Moreover, identity lifecycle management is implemented by manipulating the amount of LiIDCoins, rather than key managements (e.g., renewal and revocation) in conventional schemes.

5. Design Details

This section elaborates the design details of the proposed authentication scheme.

5.1. Initialization. The initialization phase is executed by the trust layer when the consortium blockchain is deployed on administrative servers from different domains. The main job is to select system parameters, including hash function and

Elliptic Curve Cryptography (ECC), which is quite similar to the setup phase in the PKI system. The hash function $H(\cdot)$ can be SHA-1 or more secure version. Meanwhile, a non-singular elliptic curve in the following form is selected:

$$\left\{(x,y) \in \left(Z_p\right)^2 | y^2 \equiv x^3 + ax + b (\mathrm{mod} p), 4a^3 + 27b^2 \equiv 0 (\mathrm{mod} p)\right\} \cup \{\mathcal{O}\}, \quad (3)$$

where p is a large prime, $a, b \in Z_p$ are two constants, and $\mathcal{O}$ is the point at infinity. Then, a base point G is picked on the curve whose order is n such that $n \cdot G = \mathcal{O}$.

The above parameters make a parameter tuple $\{H(\cdot), \{p, a, b, G, n\}\}$, which will be published both inside and outside the blockchain. Note that, as long as the hash function and ECC are secure, publishing these parameters will not affect the resistance ability against cryptography attacks. Furthermore, according to the concept of consortium blockchain, to join in the system, all participants, including domain administrators (e.g., TA_X and TA_Y), need to generate a public-private key pair by

$$\mathrm{pk} = \mathrm{sk} \cdot G, \mathrm{sk} \in Z_p, \quad (4)$$

where sk is the private key randomly selected from Z_p and pk is the public key. At the end of the initialization phase, each domain administrative TA publishes its public key inside the blockchain while keeping its private key as secret.

5.2. Off-Chain Registration. Each entity needs to register at its administrative TA to get engaged in the blockchain-based authentication service by the following off-chain steps.

(1) Firstly, an entity, such as E_X^i, gets the public parameter tuple from the blockchain and generates a public-private key pair $(\mathrm{sk}_{E_X^i}, \mathrm{pk}_{E_X^i})$ according to equation (4). Then, it submits its real identity and public key $\mathrm{pk}_{E_X^i}$ to its administrator TA_X through an internal and secure channel.

(2) Secondly, TA_X verifies E_X^i's real identity to generate a certificate $\mathrm{Cert}_{E_X^i}$ as its trust anchor within the blockchain and then calculates E_X^i's address on the blockchain by $\mathrm{ID}_{E_X^i} = H(\mathrm{pk}_{E_X^i})$. Besides, TA_X activates E_X^i's authentication service through issuing a certain amount of LiIDCoins to the address $\mathrm{ID}_{E_X^i}$, which will be detailed in Section 5.3.1.

(3) Finally, TA_X returns the address value $\mathrm{ID}_{E_X^i}$ to E_X^i as a pseudonym for the rest phases. So far, the entity has completed the registration and only needs to keep the private key $\mathrm{sk}_{E_X^i}$ in the local device. It is worth noting that the certificate has never been exposed to the outside of the blockchain.

5.3. On-Chain Operations. After finishing registration, the entity obtains the authentication service through a set of on-chain operations on LiIDCoins, which covers the whole lifecycle of its identity.

5.3.1. Issue. As mentioned above, TA_X issues a certain amount of LiIDCoins to activate a validated entity E_X^i during the registration phase. This is implemented by a transaction from TA_X to E_X^i in the following form of UXTO:

$$\begin{cases} \mathrm{basic} = \mathrm{tx\_id} \parallel t \parallel \mathrm{issue} \parallel \mathrm{Sig}_{\mathrm{TA}_X}(\cdot) \\ \mathrm{in} = \mathrm{ID}_{\mathrm{TA}_X} \parallel H\left(\mathrm{Cert}_{E_X^i}\right) \\ \mathrm{out} = \mathrm{ID}_{E_X^i} \parallel Q_{\mathrm{issue}} \end{cases}, \quad (5)$$

where tx_id is a transaction ID generated by the blockchain, t is the timestamp when the transaction is created, issue indicates the operation name, $\mathrm{Sig}_{\mathrm{TA}_X}(\cdot)$ shows the approval for the transaction from TA_X, $\mathrm{ID}_{\mathrm{TA}_X}$ and $\mathrm{ID}_{E_X^i}$ denote that the transaction is launched by TA_X and sent to E_X^i, $H(\mathrm{Cert}_{E_X^i})$ is used to define the entity-specific type of these LiIDCoins, and Q_{issue} is the quantity of issued LiIDCoins. According to the queuing theory, Q_{issue} can be calculated by

$$Q_{\mathrm{issue}} = \lambda W, \quad (6)$$

where λ is the arrival rate of LiIDCoin transactions and W is the processing time per transaction.

5.3.2. Transfer. A successful LiIDCoin transfer from E_X^i to E_Y^j reveals two facts: on one side, *claimant* E_X^i has the ability to launch a transfer endorsed by TA_X, which is equivalent to showing a certificate signed by its CA in PKI, thus claiming the legitimacy of E_X^i. On the other side, such a transfer activity is recorded as a transaction on the blockchain, which will be confirmed by the *verifier* E_Y^j as well as be used for auditing purposes. The UXTO for the transfer operation is formulated by

$$\begin{cases} \mathrm{basic} = \mathrm{tx\_id} \parallel t \parallel \mathrm{transfer} \parallel \mathrm{Sig}_{E_X^i}(\cdot) \\ \mathrm{in} = \mathrm{ID}_{E_X^i} \parallel H\left(\mathrm{Cert}_{E_X^i}\right) \\ \mathrm{out} = \begin{bmatrix} \mathrm{ID}_{E_Y^j} \parallel 1 \\ \mathrm{ID}_{E_X^i} \parallel Q_{\mathrm{balance}} - 1 \end{bmatrix} \end{cases}, \quad (7)$$

where transfer indicates the operation name, $\mathrm{Sig}_{E_X^i}(\cdot)$ is the transaction approval from E_X^i, $\mathrm{ID}_{E_X^i}$ in in part denotes LiIDCoins in this transaction are coming from E_X^i while two recipients exist in the out part, with 1 for E_Y^j and the rest for the change, respectively, and Q_{balance} is the number of LiIDCoins owned by E_X^i before this transaction.

As aforementioned, LiIDCoin has a one-time use property, which is essentially different from other digital tokens. This is implemented by the Type attribute, which is instantiated as $H(\mathrm{Cert}_{E_X^i})$ for entity E_X^i. Before the transaction is actually recorded in the blockchain, it will be validated to ensure that the Type value conforms to the declared sender by verifying its signature in the UTXO and recalculating the hash value of its certificate. For example, supposing that E_Y^j intends to spend the LiIDCoin that it has just received from E_X^i in equation (7), E_Y^j needs to fill the in part as $\mathrm{ID}_{E_Y^j} \parallel H(\mathrm{Cert}_{E_X^i})$. Note that Type is a read-only attribute once a LiIDCoin is issued by the TA, which implies

that E_Y^j cannot alter it to its own hash value (e.g., $H(\text{Cert}_{E_Y^j})$). In this case, the Type value will not conform to the signature for this transaction, leading to a rejection by the blockchain system.

5.3.3. Query. The query operation is quite straightforward but essential to confirm the authenticity of an entity. Given an index (e.g., tx_id) of a transaction, this operation provides required information about the transaction from the blockchain, which is fairly useful for the LiIDCoin recipient to verify sender's claim. The simplest return value of the query operation can be True/False, indicating the validation result of the transaction.

5.3.4. Revoke. This operation is equivalent to the certificate revocation in PKI, which is the key way to control the trust lifetime of an entity. However, most conventional schemes employ a black list, such as CRL, which requires extra storage, computation, and communication costs. In contrast, the proposed scheme achieves the goal through revoking all LiIDCoins of the entity by the following UTXO:

$$\begin{cases} \text{basic} = \text{tx\_id} \parallel t \parallel \text{revoke} \parallel \text{Sig}_{\text{TA}_X}(\cdot) \\ \text{in} = \text{ID}_{E_X^i} \parallel H\left(\text{Cert}_{E_X^i}\right) \\ \text{out} = \text{ID}_{\text{TA}_X} \parallel Q_{\text{balance}} \end{cases}, \tag{8}$$

where revoke is the operation name and $\text{Sig}_{\text{TA}_X}(\cdot)$ denotes the approval from TA_X. The sender and the recipient of Q_{balance} LiIDCoins are E_X^i and TA_X, respectively. As a result, E_X^i's LiIDCoin is cleared and E_X^i is no longer able to launch any new transaction, which implies it cannot be authenticated by others. This kind of transaction is allowed by the blockchain because TA_X's endorser role for E_X^i can be proved by investigating certificate $\text{Cert}_{E_X^i}$ and the Type attribute.

In summary, LiIDCoin is able to represent the whole lifecycle of entity identification, which can be easily activated and deactivated through issuing and revoking LiIDCoins, respectively. Besides, entity states are published and updated in a synchronous manner through the blockchain ledger, without any intermediate support (e.g., certificate and CRL), which saves significant management efforts.

5.4. Authentication Protocol. Based on the above coin operations, the proposed authentication scheme is implemented by three rounds of request-response communications. The whole process is illustrated in Figure 5, where claimant E_X^i from domain X proves its identity to *verifier* E_Y^j in domain Y. Note that the blockchain system itself is assumed to be secure since it is beyond the scope of this paper. In this case, we only illustrate essential payload in each request/response packet.

The whole authentication process is accomplished by the following six steps. In step ①, E_X^i sends a request to its regional server TA_X for launching a LiIDCoin transfer transaction by providing *verifier*'s ID (e.g., $\text{ID}_{E_Y^j}$) and *claimant*'s approval (e.g., $\text{Sig}_{E_X^i}(\text{ID}_{E_Y^j})$). Upon receiving the request, TA_X checks its validity by an ingoing authentication algorithm (Algorithm 1). If the request is valid, the transaction is submitted to the blockchain for updating the ledger, and a transaction ID tx_id is returned to E_X^i in step ②. Based on the transaction, E_X^i formally sends an authentication request framed with tx_id to E_Y^j in the following form:

$$\text{Auth}_{i,j}(\text{Payload}) = \text{ID}_i \parallel \text{Timestamp} \parallel \text{Payload} \parallel \text{Sig}_i(\cdot), \tag{9}$$

where $\text{Auth}_{i,j}$ represents an authentication request from *claimant i* to *verifier j*, Payload indicates the message carried in the packet, and Timestamp is a point in time, used to prevent replay attack. Furthermore, the whole packet is signed by the *claimant* in case of spoofing attack. To determine whether the *claimant* is trustworthy, E_Y^j forwards $\text{Auth}_{i,j}$ to TA_Y to confirm the existence and the validity of the transaction by an outgoing authentication algorithm (Algorithm 2 during steps ④ and ⑤). Finally, a decision of acceptance or rejection is immediately sent back to E_X^i in step ⑥.

Algorithm 1 depicts the key process of approving a new LiIDCoin transaction. Once receiving a transfer request from E_X^i, its certificate $\text{Cert}_{E_X^i}$ is retrieved from the blockchain system (line 1) and then E_X^i's public key $\text{pk}_{E_X^i}$ is extracted from $\text{Cert}_{E_X^i}$ (line 2). If the signature in the request can be verified using $\text{pk}_{E_X^i}$, a transfer UXTO is created and submitted to the blockchain to generate the transaction (lines 3–7). Note that generate Transaction represents a metafunction provided by the blockchain for generating a transaction. Finally, the transaction ID tx_id is returned if no error occurs (line 6), otherwise error_code is provided (line 8).

On the other side, the process of validating a given authentication request is shown in Algorithm 2. First, the signature in the request is verified (lines 1–3). If the signature is valid, the UTXO of the transaction indicated by tx_id is retrieved from the blockchain by another metafunction query Transaction (line 4). Then, both the sender and the recipient in the UTXO are verified to conform to the *claimant* and the *verifier*, respectively (line 5). Furthermore, the time frame of the authentication is checked by $|\text{UTXO.Timestamp} - T_{\text{current}}| < \Delta T$ (line 6), where T_{current} is the current time and ΔT is the maximum allowable delay. After these investigations, True/False will be returned to indicate the judgment of the transaction (lines 7 and 11).

Besides, E_Y^j maintains a local database to store transaction IDs that have been used for authentication. So far, the identity of E_X^i has been successfully authenticated by E_Y^j, without transmitting any certificate or CRL files.

5.5. Mutual Authentication and Key Agreement. Mutual authentication refers to two entities authenticating each other's identities before formal communication is established. This can be implemented by adding an inverse pass, as shown in Figure 6. After E_X^i proves its identity to E_Y^j through transaction $\text{tx\_id}_{i,j}$, the two entities exchange their roles, with E_Y^j the *claimant* and E_X^i the *verifier*, which means E_Y^j also proves its identity to E_X^i through another transaction $\text{tx\_id}_{j,i}$ following the above protocol.

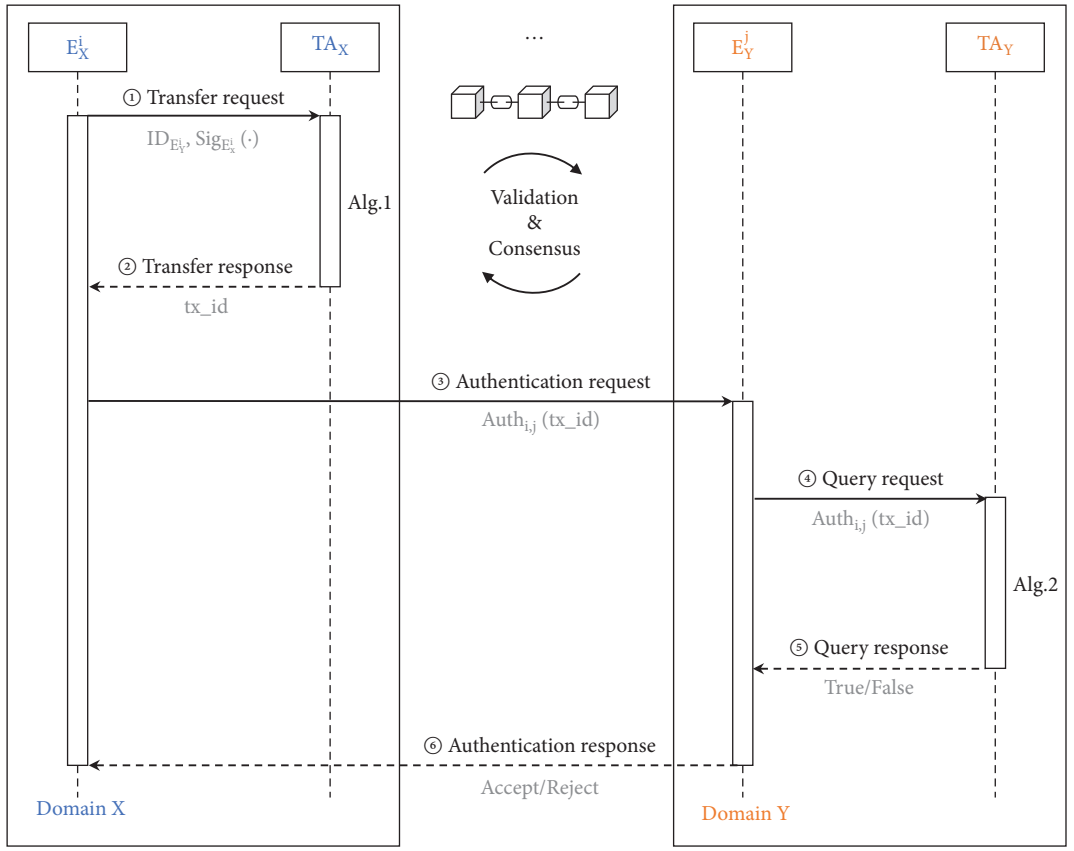

Figure 5: Authentication process of the proposed scheme.

Input: $ID_{E_Y^j}$ and $Sig_{E_X^i}(\cdot)$
Output: tx_id or error_code.
(1) Get E_X^i's certificate $Cert_{E_X^i}$ in the blockchain
(2) Extract E_X^i's public key $pk_{E_X^i}$ from $Cert_{E_X^i}$
(3) **if** $Sig_{E_X^i}(\cdot)$ is valid by using $pk_{E_X^i}$ **then**
(4) Construct a transfer UTXO from E_X^i to E_Y^j
(5) tx_id←generate Transaction (UXTO)
(6) **return** tx_id
(7) **end if**
(8) **return** error_code

Algorithm 1: Ingoing authentication.

Input: $Auth_{i,j}$ and $ID_{E_X^i}$
Output: True or False
(1) Get E_X^i's certificate $Cert_{E_X^i}$ in the blockchain
(2) Extract E_X^i's public key $pk_{E_X^i}$ from $Cert_{E_X^i}$
(3) **if** $Auth_{i,j}.Sig_{E_X^i}(\cdot)$ is valid by using $pk_{E_X^i}$ **then**
(4) UTXO←query Transaction ($Auth_{i,j}$.tx_id)
(5) **if** UTXO.Sender_id == $ID_{E_X^i}$ && UTXO.Recipient_id == $Auth_{i,j}.ID_{E_Y^j}$ **then**
(6) **if** $|UTXO.Timestamp - T_{current}| < \Delta T$ **then**
(7) **return** True
(8) **end if**
(9) **end if**
(10) **end if**
(11) **return** False

Algorithm 2: Outgoing authentication.

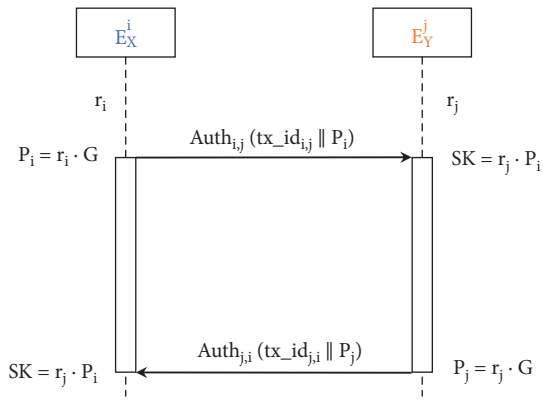

Figure 6: Mutual authentication and session key exchange.

Upon a successful mutual authentication, the two entities should be able to negotiate a session key for secure communication, which is called key agreement. Based on the elliptic curve in the system setup, we use the Ephemeral Elliptic Curve-based Diffie–Hellman (ECDHE) mechanism for key negotiation and incorporate it into the authentication process. ECDHE is a key agreement protocol that allows two entities to compute the same session key by exchanging their public keys based on the ECC algorithm. Specifically, the two entities obtain the parameter tuple $\{p, a, b, G, n\}$ from the blockchain. As shown in Figure 6, E_X^i and E_Y^j randomly pick two secret keys $r_i, r_j \in Z_p$, respectively. Then, E_X^i computes its public key $P_i = r_i \cdot G$ and sends it to E_Y^j through $\text{Auth}_{i,j}$. Based on that, E_Y^j gets the session key by $\text{SK} = r_j \cdot P_i = r_j \cdot r_i \cdot G$. Similarly, in the inverse pass for authentication, E_Y^j sends its public key $P_j = r_j \cdot G$ to E_X^i; thus, E_X^i obtains the same session key by $\text{SK} = r_i \cdot P_j = r_i \cdot r_j \cdot G$.

The ECDHE mechanism provides perfect forward secrecy by using the ephemeral keys. Moreover, these keys are exchanged through the authentication process; thus, security concerns (e.g., man-in-the-middle attack and replay-attack) can be addressed along with the proposed scheme. Details about security analysis will be given in Section 6.

5.6. LiIDCoin Recycling. The one-time use property of LiIDCoins will make them exhausted eventually, which can be easily tackled by issuing new LiIDCoins to the entity based on its trust evaluation. This process is called LiIDCoin recycling (or renewal), which can be performed in two modes, including real-time mode and periodic mode. In the real-time mode, once a LiIDCoin is spent for the authentication, a new one will be immediately issued back for supplementation. In this case, the entity will always have sufficient LiIDCoins for authentication service, but trust evaluation complemented with each renewal process could increase system workload. In case of that, the periodic mode allows entities to be supplemented at a specific time interval, and LiIDCoin exhaustion should also be considered. In practice, the two modes can be selected according to specific applications.

6. Security Analysis

In this section, we analyze the security requirements that the proposed scheme can accommodate. Note that the cryptographic primitives and the blockchain are assumed to be secure since they are beyond the scope of this paper. Details of the security analysis are given as follows.

6.1. Single Registration. For ease use, each entity needs to register only once before it enjoys the authentication service. According to the proposed scheme, an entity enrolls into the system at its administrative TA through an off-chain registration. The entity submits its real identity and public key during the registration. Once they are approved, the TA generates a certificate and a certain amount of LiIDCoin for the entity in the blockchain. After that, the entity can launch requests without any further registration, and the request messages can be signed by the entity using it private key independently. Therefore, the proposed scheme is able to provide a one-off registration.

6.2. Forgery Attack. The adversary spoofs the signature of a legitimate entity and sends it to others. In the proposed scheme, the private key is generated and preserved by each entity itself. As long as it is not compromised, the cryptographic theory can guarantee that it is infeasible to fake a valid signature without knowing the private key.

6.3. Man-in-the-Middle Attack. The adversary secretly intercepts communications or alters them between two entities. In our proposal, the authentication request is represented by $\text{Auth}_{i,j}$ (tx_id), in which the *claimant* and the *verifier* are already determined in the form of a transaction in the blockchain before it is sent. The adversary cannot insert itself into a communication without interacting with the blockchain. Once it happens, the malicious behavior will be revealed by the blockchain ledger.

6.4. Replay Attack. The adversary fraudulently delays or resends a request to the receiver. In the proposed scheme, a timestamp is introduced to the authentication request along

with a time threshold, to ensure the particular request cannot be processed more than once. Even within the threshold, a used request can be easily detected by looking up a local database that stores the latest accepted ones.

6.5. Identity Revealing Attack. In this attack, the adversary attempts to reveal the real identity of a target entity. According to the proposal, the real identity of each entity is only required during the registration phase and an address is then assigned to the entity as a pseudonym for all later operations. Moreover, the identity-pseudonym mapping is exclusively maintained by its administrative TA. As long as the TA is not compromised, the privacy of the entity is preserved.

6.6. Deny-of-Service Attack. The adversary attempts to overload the system to prevent it from serving clients. In our proposal, the authentication service is mainly implemented by transferring LiIDCoins between entities. The number of LiIDCoins owned by each entity is limited and each transaction is recorded in the blockchain. In this case, the adversary needs to pay a price and risk its reputation to launch such an attack. Moreover, abnormal transactions can be detected from the server side.

6.7. Authority Abuse Attack. In this attack, a TA arbitrarily issues tokens to illegal entities or revoke tokens of legal entities. In the proposed scheme, all the activities of issuing and revoking tokens are transparent and verifiable in the blockchain. Based on that, abuse behaviors of a TA affect its reputation in the consortium; thus, it could be eliminated from the system.

7. Implementation and Evaluation

In this section, we implement our scheme and conduct experiments to evaluate its performance in terms of functionality, storage cost, communication cost, and authentication efficiency.

7.1. Experimental Setup. Two administrative domains are simulated in the experiments. In each domain, there are one TA server and one IoT device, respectively. In addition, as we implement our scheme on the HyperLedger Fabric platform and use the Raft algorithm for consensus, we deploy a third server to satisfy the requirement for an odd number of ordering servers. Figure 7(a) illustrates the logical architecture of the simulation. Therefore, we have three Lenovo 510S desktops with Intel Core i5 CPU@2.9 GHz and 8 GB memory running on CentOS v8.3 for TA servers, and two Raspberry Pi 3B+ with Broadcom Cortex-A53 CPU@1.4 GHz and 1 GB memory as IoT devices, as shown in Figure 7(b). All desktops and IoT devices are connected to their subnetworks through a H3C S5560 switch, which supports multiple Virtual Local Area Networks (VLANs) and network routing. Furthermore, we deploy HLF v2.0 as the consortium blockchain and implement smart contracts (also known as chaincode) using Node.js. For the client side, cryptographic primitives and authentication protocol are implemented by *Python*.

We compared our scheme with four classic competitors, including PKI, CertCoin [18], AuthCoin [19], CeCoin [20], and BASA [22]. For comparison purposes, we use the same cryptographic parameter settings. Specifically, we use the *prime256v1* elliptic curve to create the public/private keys and Elliptic Curve Digital Signature Algorithm (ECDSA) for generating signatures. In this case, the lengths of the private key and the signature are 256 bits and 512 bits, respectively.

7.2. Functionality. We compared the functionalities that have been achieved (indicated by 0) or not achieved (indicated by ×) in Table 3. All schemes have the functions of registration and validation, only AuthCoin and BASA can hardly renew and revoke identity proofs since the former inherits the spirits from Web of Trust (WoT) and the latter employ a pseudoanonymous identity for each authentication. Besides, blockchain-based schemes realize the function of transparency except for AuthCoin since it only stores challenge-response messages in the blockchain. Among these schemes, only BASA and the proposed scheme have the function of lightweight client, which is critically important for IoT scenarios. However, since BASA is based on IBC, the KGC introduces a key-escrow problem as well as a restriction on mobility. More results will be given in the following experiments to support the above conclusion.

7.3. Storage Cost. The comparison results of storage cost for the six authentication schemes are listed in Table 4. Note that we only count in the most necessary data that should be stored in local device for authentication. As we used 256 bit ECC keys, we have the size of private key $S_{sk} = 32$ bytes while the size of certificate S_{Cert} could be various according to its contents; in this experiment, $S_{Cert} = 557$ bytes. The total rough costs of the six schemes are also given in Table 4.

For PKI, we assume that OCSP is used for certificate status validation, otherwise it will cost extra storage to maintain CRL files. In CertCoin, each entity is associated with two pairs of public-private keys for privacy preservation. Among the above schemes, LiIDCoin cost the least storage since only the private key is kept in local device while others (e.g., certificates) are maintained in the blockchain servers.

7.4. Communication Cost. We evaluate the communication cost of the authentication scheme by accumulating all key payloads during the authentication process. Since CertCoin, AuthCoin, and CeCoin have not been implemented yet, we only compared our scheme with PKI and BASA in this experiment. As shown in Figures 3 and 5, PKI has four steps while LiIDCoin has six steps. The original work of BASA has three components in the server side and requires 24 steps for one unilateral authentication. To fit the topology in this experiment, we deployed all server-side components of BASA on the same physical server; thus, we only consider communication cost between the IoT device and the server. The size of entity ID,

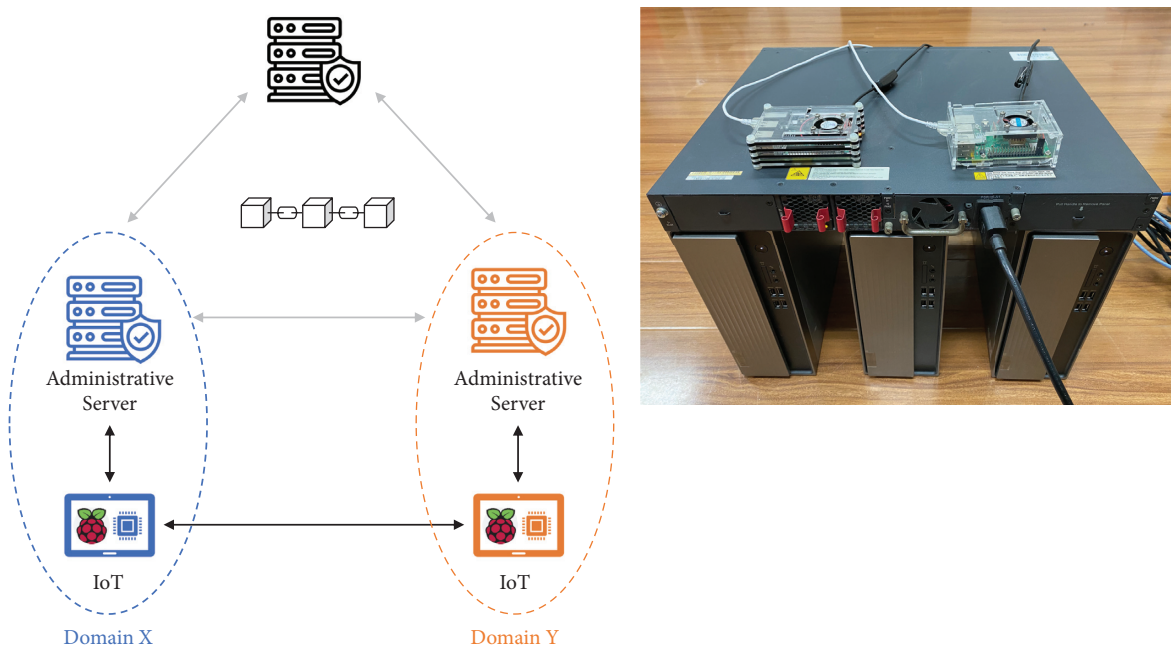

Figure 7: (a) Logical architecture and (b) physical architecture of the simulated cross-domain IoT.

Table 3: Comparison of functionality.

Function	PKI	CertCoin	AuthCoin	CeCoin	BASA	LiIDCoin
Registration	✓	✓	✓	✓	✓	✓
Renewal	✓	✓	×	✓	×	✓
Revocation	✓	✓	×	✓	×	✓
Validation	✓	✓	✓	✓	✓	✓
Transparency	×	✓	×	✓	✓	✓
Privacy preservation	×	×	×	×	✓	✓
Lightweight client	×	×	×	×	✓	✓
Mobility	✓	✓	✓	✓	×	✓

Table 4: Comparison of storage cost.

Scheme	Total cost	Total rough cost (byte)
PKI	$2S_{Cert} + S_{sk}$	1146
CertCoin	$2(S_{pk} + S_{sk})$	128
AuthCoin	$S_{pk} + S_{sk}$	64
CeCoin	$S_{Cert} + S_{sk}$	589
BASA	$2S_{sk}$	64
LiIDCoin	S_{sk}	32

signature, timestamp, random number, cryptographic key, transaction ID, certificate ID, and binary result (Yes/No or True/False) is denoted as S_{ID}, S_{Sig}, S_{TS}, S_N, S_{Key}, S_{Tx_id}, and S_{Bin}, the values of which are 32, 64, 4, 4, 32, 32, and 1 byte(s), respectively. We use the same certificate in the above experiment; thus, we have $S_{Cert} = 557$ and $S_{Cert\_ID} = 20$. The rough cost of PKI authentication is 643 bytes, in which certificate occupies the majority. BASA costs 430 bytes in communication; it is worth mentioning that when the server-side components are deployed in a distributed way, this value would be even more. By contract, the communication overhead of LiIDCoin is only 394 bytes. Details are listed in Table 5.

7.5. Authentication Efficiency. The efficiency of the whole authentication process is another critical metric for evaluating an authentication scheme. Since the proposed scheme is closely associated with the on-chain operations (e.g., transfer and query), we firstly investigate the server-side efficiency, which is normally described by the Transactions Per Second (TPS) of the blockchain. We launched 10 000 transactions for each operation with respect to different transaction numbers encapsulated in a block. As illustrated in Figure 8, TPS of each operation increases and then becomes stable as the block-encapsulated transactions grows. Being as the most complex transaction, Transfer is less

Table 5: Comparison of communication cost.

Step	PKI	BASA	LiIDCoin
①	$S_{Cert} + S_{Sig}$	$2S_{ID} + S_{Sig} + S_N$	$S_{ID} + S_{Sig}$
②	S_{Cert_ID}	$S_{Key} + S_{Sig}$	S_{Tx_id}
③	S_{Bin}	$S_N + S_{ID} + S_{Sig}$	$S_{ID} + S_{TS} + S_{Tx\_id} + S_{Sig}$
④	S_{Bin}	$S_N + S_{ID} + S_{Sig}$	$S_{ID} + S_{TS} + S_{Tx\_id} + S_{Sig}$
⑤	—	S_{Bin}	S_{Bin}
⑥	—	S_{Bin}	S_{Bin}
Total rough cost (byte)	643	430	394

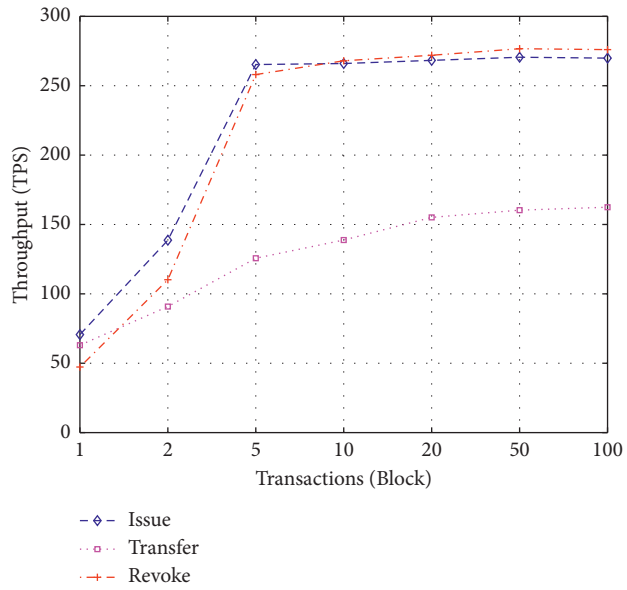

Figure 8: Throughput of LiIDCoin operations.

efficient than the other two operations. It is worth mentioning that the throughput can be further improved by employing high-performance servers as well as other optimization strategies [31].

When the maximum throughput is selected, the time costs of on-chain operations are listed in Table 6. Although Transfer is the most time-consuming operation, it is still less than 7 ms. Note that Query is a read-only operation, which is not related to transaction settings.

Besides, the overall time cost of an authentication scheme is highly affected by the required cryptographic primitives. Table 7 lists time costs of the basic cryptographic operations used in this experiment. Each data is an average value of 100 measurements. It can be observed that time cost of cryptographic computations on the server is far lower than that on the IoT device.

Based on this knowledge, we compared the whole authentication process of LiIDCoin with that of BASA and PKI in this experiment. We conduct authentications by using BASA, PKI, and LiIDCoin, respectively, each for 100 times. The average time costs of the three schemes are illustrated in Figure 9, in which LiIDCoin is the most time efficient, costing only 72 ms on average for one unilateral authentication. We also investigate time consumption of the main operations in each scheme. Since all cryptographic operations in PKI need to be conducted by the IoT devices while

Table 6: Time cost of on-chain operations.

Operation	Average time (ms)
Issue	3.71
Transfer	6.16
Query	4.31
Revoke	3.62

those in LiIDCoin can be partly performed in the TA servers, the latter scheme consumes less time than the former in terms of cryptographic operations. Meanwhile, although BASA also migrates cryptographic operations from IoT devices to powerful severs, it introduces more cryptographic operations and require more steps to perform one single authentication. In contrast to PKI, LiIDCoin requires extra time for blockchain operations (e.g., Transfer and Query). However, according to Table 6, time consumption of these operations are acceptable. Besides basic operations, there are many other steps, including network transmission, for completing the authentication. In PKI, the step of certificate status validation tends to take up a significant part of the time cost, which is coherent to the investigative findings in OCSP applications [32]. Compared to PKI, BASA requires much more communication due to its numerous steps, while LiIDCoin spends less time in network transmission since it has moderate steps and fully utilizes regional network for communications.

7.6. Discussion and Limitations. In the evaluation, we investigate various metrics of the proposed method LiIDCoin, including functionality, storage cost, communication cost, and authentication efficiency from both the server side and the client side. Compared with the traditional PKI, LiIDCoin is more efficient in terms of storage, communication and client-side authentication, and trust management overhead is largely mitigated. In comparison with the blockchain-based competitor BASA, LiIDCoin is not based on IBC, thus does not have key-escrow problems, and is more feasible for mobile IoT applications. Besides, it costs less communication bandwidth and computation resources.

As our work is still an early study in the field of blockchain-based IoT authentication, there remains some open challenges that can be considered. Firstly, as the infrastructure of authentication service, blockchain still faces some issues, such as server-side storage cost, energy cost, and scalability. Secondly, in the current proposal, the entity ID is used during it is entire lifecycle, which may suffer from

Table 7: Time cost of cryptographic operations.

Operation	Average time (ms)	
	Server (Lenovo PC)	IoT (Raspberry Pi)
Sign	2.04	21.98
Verify	0.86	15.53

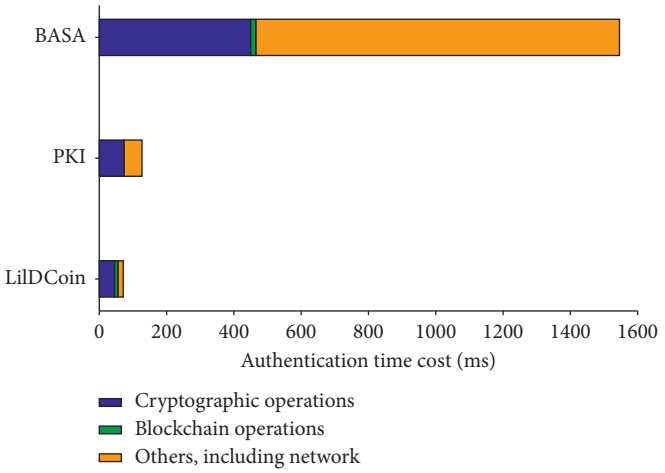

Figure 9: Time cost of the whole authentication process.

a linkability attack. Thus, privacy issues can be further enhanced. Thirdly, there is a need to optimize trust management in segments of LiIDCoin recycling and activity auditing. We plan to explore the above challenges based on the current implementation in our future work.

8. Conclusion

In this paper, we proposed a lightweight authentication scheme based on consortium blockchain by designing a cryptocurrency-like digital token named LiIDCoin to represent entity trust. In this way, the authenticity of an entity can be proved by providing evidence of its transaction capability in the blockchain. Moreover, lifecycle management can be carried out by manipulating the amount of LiIDCoins. We conducted an overall analysis to demonstrate the security requirements that the proposed scheme satisfies. Furthermore, we implemented our scheme on the HLF platform and compared it with several competitor schemes. Experimental results reveal that our scheme is more efficient in terms of storage, communication, and authentication efficiency.

In future work, we would like to conduct extensive experiments on real-world IoT applications as well as extending our scheme to more cross-domain authentication scenarios, such as eduroam (education roaming). Privacy enhancement and fine-grained trust management can also be considered.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Natural Science Foundation of Jiangsu Province of China (Grant no. BK20190346), the National Natural Science Foundation of China (Grant no. 61971131), the Fundamental Research Funds for the Central Universities (Grant no. 3209002102C3), and the 2019 Industrial Internet Innovation and Development Project, Ministry of Industry and Information Technology of China (Grant no. 6709010003). Fei Tong was also supported by "Zhishan" Scholars Programs of Southeast University.

References

[1] "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030," https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[2] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[3] M. Antonakakis, T. April, M. Bailey et al., "Understanding the Mirai botnet," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1093–1110, USENIX Association, Vancouver BC Canada, 16 August 2017.

[4] P. Middleton, R. Contu, B. Pace, and S. Alaybeyi, "Forecast: IoT security, worldwide, 2018," Technical Report G00351051, Gartner Research, USA, Feb 2018.

[5] M. b. Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: a survey," *Computer Networks*, vol. 148, pp. 283–294, 2019.

[6] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[7] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[8] P. Vijayakumar, M. S. Obaidat, M. Azees, S. H. Islam, and N. Kumar, "Efficient and secure anonymous authentication with location privacy for IoT-based WBANs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2603–2611, 2020.

[9] J. Li, J. Jin, L. Lyu et al., "A fast and scalable authentication scheme in IoT for smart living," *Future Generation Computer Systems*, vol. 117, pp. 125–137, 2021.

[10] M. Wang, C. Qian, X. Li, and S. Shi, "Collaborative validation of public-key certificates for IoT by distributed caching," in *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*, pp. 847–855.

[11] S. Malani, J. Srinivas, A. K. Das, S. Kannan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, 2019.

[12] S. P. Gochhayat, C. Lal, L. Sharma et al., "Reliable and secure data transfer in IoT networks," *Wireless Networks*, vol. 26, no. 8, pp. 5689–5702, 2020.

[13] Li Duan, Y. Li, and L. Liao, "Flexible certificate revocation list for efficient authentication in IoT," in *Proceedings of the 8th International Conference on Internet of Things (IoT)*, pp. 1–8, ACM, Santa Barbara California USA, 15 October 2018.

[14] L. Duan, Y. Li, and L. Liao, "Non-interactive certificate update protocol for efficient authentication in IoT," *Future Generation Computer Systems*, vol. 113, pp. 132–144, 2020.

[15] S. Matsumoto and R. M. Reischuk, "IKP: turning a PKI around with decentralized automated incentives," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P)*, pp. 410–426, IEEE, San Jose, CA, USA, May 2017.

[16] O. Salman, S. Abdallah, I. H. Elhajj, C. Ali, and A. Kayssi, "Identity-based authentication scheme for the Internet of Things," in *Proceedings of the 2016 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–3, IEEE, June 2016.

[17] Z. Wang, J. Lin, Q. Cai, Q. Wang, J. Jing, and D. Zha, "Blockchain-based certificate transparency and revocation transparency," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, 2020.

[18] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: a namecoin based decentralized authentication system," Technical Report 6.857 Class Project, Massachusetts Institute of Technology, MA, USA, 2014.

[19] L. Benjamin, H. Clemens, T. Mundt, and S. Rashidibajgan, "AuthCoin: validation and authentication in decentralized networks," in *Proceedings of the 10th Mediterranean Conference on Information Systems (MCIS)*, pp. 1–14, University of Nicosia, Paphos, Cyprus, September 2016.

[20] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, and W. Shi, "CeCoin: a decentralized PKI mitigating MitM attacks," *Future Generation Computer Systems*, vol. 107, pp. 805–815, 2020.

[21] X. Jia, N. Hu, S. Su et al., "IRBA: an identity-based cross-domain authentication scheme for the Internet of Things," *Electronics*, vol. 9, no. 4, pp. 1–21, 2020.

[22] M. Shen, H. Liu, L. Zhu et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[23] D. Li, Y. Jia, X. Gao, and N. Al-Nabhan, "Research on multidomain authentication of IoT based on cross-chain technology," *Security and Communication Networks*, vol. 2020, Article ID 6679022, 12 pages, 2020.

[24] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," http://www.bitcoin.org/bitcoin.pdf.

[25] "Ethereum," 2021, https://ethereum.org/.

[26] Y. Zhang, F. Tong, Y. Xu, J. Tao, and G. Cheng, "A privacy-preserving authentication scheme for VANETs based on consortium blockchain," in *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC)*, pp. 1–6, IEEE, Victoria, BC, Canada, November 2020.

[27] Z. Ma, L. Liu, and W. Meng, "Towards multiple-mix-attack detection via consensus-based trust management in IoT networks," *Computers & Security*, vol. 96, pp. 1–21, Article ID 101898, 2020.

[28] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3701–3711, 2018.

[29] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: a decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126–142, 2018.

[30] W. Meng, W. Li, and J. Zhou, "Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration," *Information Fusion*, vol. 70, pp. 60–71, 2020.

[31] C. Gorenflo, S. Lee, L. Golab, and K. Srinivasan, "Fastfabric: scaling hyperledger fabric to 20,000 transactions per second," in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 455–463, Wiley Online Library, Massachusetts, Amherst, USA, February 2020.

[32] T. Chung, J. Lok, B. Chandrasekaran et al., "Is the Web ready for OCSP must-staple?" in *Proceedings of the 2018 Internet Measurement Conference (IMC)*, pp. 105–118, ACM, Boston MA USA, October 2018.