Blockchain-Assisted Secure Intra/Inter-Domain Authorization and Authentication for Internet of Things

Fei Tong, *Member, IEEE*, Xing Chen, Cheng Huang, *Member, IEEE*, Yujian Zhang, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Multidomain Internet of Things (IoT) is faced with serious domain interoperability (DI) and compatibility issues since different intradomain authorization and authentication (A&A) mechanisms are deployed without the consideration of interdomain A&A. This article proposes a blockchain-assisted scheme to achieve flexible intra- and inter-domain A&A simultaneously and seamlessly. Specifically, we first design a contract-based mutual access control agreement on top of a consortium blockchain, where domain managers can manage their access permission without any trusted parties. Based on the agreement, a secure and privacy-preserving authentication protocol is further proposed by tailoring one-out-of-many proof techniques, which enables IoT devices to anonymously access authorized IoT domains. We additionally design a voting-based protocol by using a threshold-based cryptosystem. The protocol allows domain managers to transparently audit resource access with the assistance of the blockchain. Detailed security analysis demonstrates that the proposed scheme achieves the security properties, such as DI, privacy protection, and accountability. Finally, we develop two proof-of-concept prototypes in a physical testbed and virtual machine, respectively, based on an open-source blockchain platform to show our scheme's efficiency in terms of computation and communication overhead.**

***Index Terms*—Authorization and authentication (A&A), blockchain, Internet of Things (IoT), intra/inter-domain, privacy protection (PP).**

I. Introduction

BASED on the high throughput and low-latency features of 5G networks [1], [2], numerous Internet of Things (IoT) applications are booming, with 14.6 billion devices already deployed to serve these applications in 2021 and expected to reach 30.2 billion by 2027 [3]. These IoT devices are able to efficiently collect and analyze data, bringing the tremendous commercial potential to various scenarios, such as smart healthcare, smart home, and smart city [4], [5]. Recent works show that the collaboration of different IoT application service providers can significantly improve the service quality and the productivity [6]. Considering that IoT application service providers generally have independent trusted domains [7], intra/inter-domain authorization and authentication (A&A) are essential security functions, by which IoT domain managers can manage their access permission for their valuable resources [8].

Traditional A&A schemes for IoT can be categorized into three types based on the techniques they used: 1) public-key infrastructure (PKI) [9]; 2) identity-based cryptography (IBC) [10]; and 3) certificateless public-key cryptography (CL-PKC) [11]. PKI-based schemes provide certificates by a trusted third-party called certificate authority (CA) to guarantee the authenticity and security of the device's identity. IBC-based schemes allow any public string to be used as an identifier of a device. IBC generates the private key corresponding to the device identifier through a trusted entity called key generation center (KGC). Different from IBC and PKI which realize A&A through trusted authority, CL-PKC-based schemes focus on using cryptographic methods, such as zero-knowledge proof and ring signature to realize certificate-free A&A.

For interdomain A&A, one of the major challenges lies in that it is difficult to be interoperable and compatible with each other in multiple IoT domains where different A&A solutions are deployed. As shown in Fig. 1, CA and KGC cannot assist in establishing trustworthiness between their devices based on locally independent A&A configurations. One of the prevailing approaches solves this issue by deploying an additional A&A scheme uniformly for all domains [12], [13], [14], [15], [16], [17], [18], [19]. Such an approach extends traditional authentication schemes to multidomain scenarios by erecting central or cloud servers with higher performance to manage intra/inter-domain A&A. However, with the explosion of IoT devices and applications in recent years, more complex and diverse multidomain environments present new challenges for A&A. One challenge is that a large number of IoT devices

Manuscript received 16 July 2022; revised 12 November 2022; accepted 2 December 2022. Date of publication 20 December 2022; date of current version 25 April 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61971131; in part by the "Zhishan" Scholars Programs, Southeast University; in part by the 2019 Industrial Internet Innovation and Development Project, Ministry of Industry and Information Technology under Grant 6709010003; and in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20190346. *(Corresponding author: Fei Tong.)*

Fei Tong is with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, Jiangsu, China, also with the Jiangsu Province Engineering Research Center of Security for Ubiquitous Network, Nanjing 211189, China, and also with Purple Mountain Laboratories, Nanjing 211111, Jiangsu, China (e-mail: ftong@seu.edu.cn).

Xing Chen and Yujian Zhang are with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, Jiangsu, China.

Cheng Huang and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Digital Object Identifier 10.1109/JIOT.2022.3229676

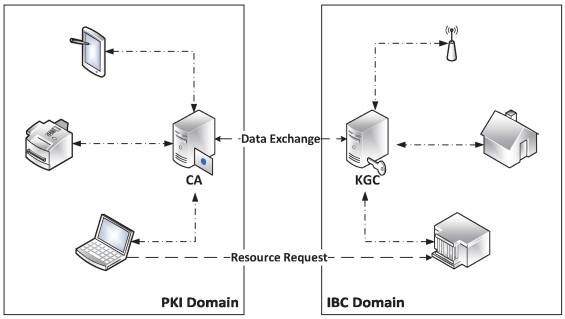


Fig. 1. Interdomain A&A case: two domains have different A&A configurations.

bring high management and deployment costs for intra/interdomain A&A solutions with centralized architectures, and the risk of a single point of failure increases [20]. From another point of view, due to unequal computing power and allocatable resources of IoT devices in different domains, it is difficult for traditional A&A schemes to be efficient and compatible in these domains simultaneously.

Recently, blockchain-based schemes have been proposed. Blockchain is a distributed database ledger with features such as decentralized trust (DT) and tamper-proofing. Depending on openness, there are three types of blockchain, including public blockchain, private blockchain, and consortium blockchain [21]. Consortium blockchain provides access rights neither completely open nor private but in between, which requires the nodes involved in maintenance to be authenticated and is commonly used in a scenario where multiple organizations collaborate. In addition, most consortium blockchain platforms provide smart contract service, which can wrap a series of on-chain operations into one atomic operation for automated, on-chain secure transactions [22]. Therefore, the architecture and functionality of consortium blockchain fit well with interdomain authentication scenario.

To tackle the aforementioned issues, this article proposes a blockchain-assisted scheme to achieve flexible intra/interdomain A&A simultaneously, enhancing domain interoperability (DI). Specifically, we first design an elaborate contract-based mutual access control agreement on top of the blockchain, which allows domain managers with different A&A configurations to authorize each other without the participation of any trusted parties. We also design an identity-based anonymous authentication protocol and seamlessly integrate it with the mutual access control agreement. By carefully tailoring one-out-of-many proofs, the protocol allows authorized IoT devices to access resources in different domains in an anonymous manner. In addition, we design a voting-based identity tracing protocol by using threshold-based encryption, to achieve the goal that malicious accesses can be audited by the administrators of the domains with the assistance of the blockchain. The major contributions of this article are listed as follows.

1) We propose a blockchain-based scheme to achieve flexible intra- and inter-domain A&A simultaneously and seamlessly. A contract-based mutual access control agreement is designed on top of the blockchain to achieve efficient authorization between domains.
2) Based on the agreement, an identity-based anonymous authentication protocol is further designed and seamlessly integrated to enable anonymous access between domains. Moreover, a voting-based identity tracing protocol is designed to guarantee that malicious access behavior can be audited.
3) We carry out a detailed security analysis on the proposed scheme and implement two proof-of-concept prototypes in a physical testbed and virtual machine, respectively, based on an open-source blockchain platform to show the practicality of our proposed scheme.

The remainder of this article is structured as follows. Section II discusses related work. Section III describes the system and adversarial models and states the design goals. Section IV introduces the building blocks and presents the detailed construction of our proposed A&A scheme. Section V analyzes the security of the scheme, and Section VI provides an implementation and evaluation of the effectiveness and efficiency of the scheme. Finally, Section VII concludes this article.

II. Related Work

Recent works on interdomain A&A for IoT can be roughly classified into two categories: 1) traditional A&A schemes and 2) domain interoperable A&A schemes.

A. Traditional A&A Schemes

The improvement of traditional A&A techniques in interdomain scenarios mostly takes the combination with blockchain as a breakthrough direction [23]. Wang et al. [24] applied PKI to the interdomain scenario, where they built a decentralized network by using the CAs of each domain as nodes of a consortium blockchain, thus avoiding the risk of a single point of failure. Li et al. [25] stored the certificates of nodes under WiFi network after hash conversion to a public ledger composed of blockchain, which achieves secure sharing of multidomain data. In addition, Qian et al. [26] redesigned the certificate format under PKI and improved the consensus algorithm in the blockchain, which enables fast A&A when the same device accesses the domain again. Ali et al. [27] even designed a multichain architecture through a global blockchain responsible for interdomain certificate management and a local blockchain for intradomain certificate management, with the two parts of the blockchain communicating and collaborating with each other. Matsumoto and Reischuk [20] mentioned that treating CAs as nodes that constitute the blockchain may lead to problems in the whole system due to potential attacks on them or their operation errors. Therefore, removing certificates from A&A has become another research direction. Shen et al. [6] and Jia et al. [28] both exploited a combination of blockchain and IBC to remove the reliance on CAs. They set up independent servers to form the blockchain, and such servers are only responsible for maintaining the blockchain and communicating with KGC, while KGC provides pseudonym generation services for the devices to ensure

privacy. Yao et al. [29] investigated the approach of CL-PKC to achieve interdomain A&A under the Internet of Vehicles, which works by equipping each vehicle with a trusted platform module (TPM) security chip, thus enabling complex cryptographic operations.

B. Domain Interoperable A&A Schemes

Faced with the requirement for DI, Lan et al. [30] defined a group key exchange protocol for multiple domains and established a new A&A channel between PKI and IBC domains that use the same encryption settings. Yuan et al. [31] optimized the interdomain key protocol, thus enabling heterogeneous interdomain A&A between PKI domains and IBC domains under enterprise instant messaging (EIM) system. Lv et al. [32] used proxy signature techniques with the assistance of a cloud A&A server to achieve interdomain signature conversion between different cryptosystem trust domains. Besides, Xiaoxue and Wenping [33] redefined a unified set of encryption schemes between PKI and IBC domains, where the administration of the accessed domain has the private key of the accessing device, thus providing full management of the device's real identity.

To further optimize and solve the problem of a single point of failure that may be caused by centralization, there are also many existing works in recent years which utilize blockchain to achieve interdomain A&A for improving DI. The model proposed in [34] does not change the internal trust structure of each A&A domain. It designs a dedicated certificate that exists on the blockchain, called BCert, to build an interdomain A&A with high scalability and DI. Xuan et al. [35] designed a certificate-free interdomain scheme, which supports parameter differentiation based on certificateless cryptography and smart contracts on blockchain. Li et al. [36] utilized smart contracts to manage the public keys of devices and designed an interdomain key protocol that enables anonymous temporary interdomain A&A. Zhang et al. [37] even analyzed in depth the design points of interdomain systems for DI, which they called "complete cross-domain" and combined hash algorithms and smart contracts to achieve efficient and interoperable interdomain A&A.

III. Models and Design Goals

In this section, we first describe the system model and the adversarial model. Then, the design goals for interdomain IoT A&A system are presented accordingly.

A. System Model

An example shown in Fig. 2 is utilized to illustrate the system model. The model contains multiple domains, which may have different configurations for intradomain A&A. For example, domain *A* in Fig. 2 adopts PKI, while *B* adopts IBC. A consortium blockchain with a public ledger is maintained by a cluster of verification servers (VSs) from the participating domains. Our IoT A&A system involves the following entities.

1) *Proxy A&A Server (PAS):* The entity responsible for identity management within a domain uniformly is referred to as PAS. In reality, this could be a CA that manages certificates in a PKI scheme, a KGC that distributes keys in an IBC scheme, or a trust authority (TA) that uses a different A&A scheme. PAS will be owned and controlled by the administrator or service provider of a domain.
2) *Verification Server (VS):* VSs are additional dedicated servers introduced to enable interdomain A&A. All VSs form a consortium blockchain environment and have smart contracts deployed while having general server capabilities. All PASs of participating domains can send requests to VSs as members to query the blockchain and invoke smart contracts. Well-funded domains can build their own VS to join the system, while cost-constrained domains can choose to rent a cloud server. Only one VS is necessary for each participating domain.
3) *Device:* Devices are the entities that initiate A&A requests. Since each device under different domains is assigned different tasks, maybe there are low-computing-power and energy-limited IoT devices or high-computing-power smartphone entities. The devices in each domain are governed by the domain's PAS, which manages identity and provides trust endorsement for the devices.

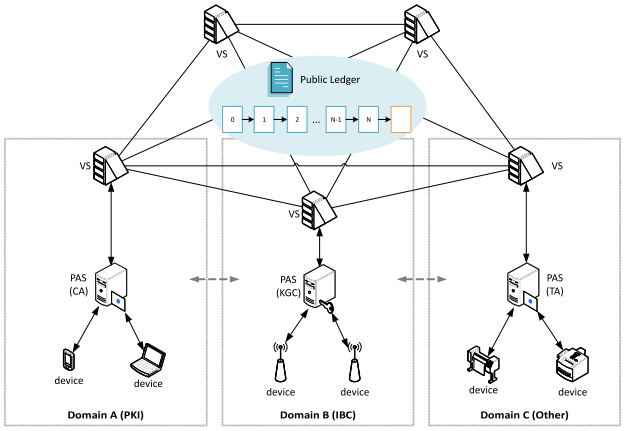


Fig. 2. Our system model.

B. Adversarial Model

In our aforementioned model, each PAS is honest and trustworthy to the devices in its own domain and has the highest level of security in intradomain A&A. The PASs from different domains cooperate with each other, through which those domains driven by resource access requirements can reach mutual access control agreement to allow access to each other. Despite the constraints of the mutual access control agreement, PAS and outer domain devices do not fully trust each other, and PAS may infer sensitive information about outer domain devices/PASs by collecting access logs. One may propose that outer domain devices/PASs can adopt anonymous techniques to protect their privacy. However, complete anonymity would also lead to security concerns, such as stealing unauthorized data through attacks, corrupting data from accessed domains, etc; moreover, the accessed domains cannot legally trace and punish anonymous devices or administrators of outer

domains, or block their future access. In addition, devices may also experience man-in-the-middle and replay attacks during interdomain A&A [38]. An adversary can affect A&A process by intercepting critical information to spoof the trusted domain or device, threatening the security, and availability of the system.

The VSs constituting the consortium blockchain are utilized to eliminate the reliance on the third-party trusted institutions. VSs consist of servers or cloud service providers from different domains following the literature on consortium blockchain [39], [40]. For the proper functioning of the consortium blockchain (successful execution of the consensus protocol), most VSs are honest and only a few ones may behave maliciously. Specifically, the number of malicious VSs should not exceed the half of the total number of VSs. Malicious VSs are able to launch a complicity attack on the system, where several VSs in multiple domains collude to break A&A of a domain. In addition, the data stored on the consortium blockchain is publicly accessible in the system. Therefore, the choice of the on-chain data also has a significant impact on the security.

C. Design Goals

Based on the system model and adversarial model, we set the following objectives to design our blockchain-assisted interdomain A&A scheme.

1) *DI:* The system can establish trust between trusted domains which have different A&A configurations, thus allowing devices from different domains to access resources across domains.
2) *DT:* The system requires DT building among multiple domains to avoid the single point of failure problem that centralization may bring and the high reliance on the credibility of the third-party trusted institutions.
3) *Domain Privacy:* Each domain needs to provide privacy protection (PP) for internal devices during interdomain A&A and resource access to avoid privacy violations by adversaries through attacks.
4) *Accountability (AC):* The PAS of the accessed domain can legally trace and punish outer domain anonymous devices/administrators with malicious behaviors, or block their future access.
5) *High Scalability:* A new trusted domain can easily and cost-effectively join the interdomain A&A scheme.
6) *High Computation Efficiency:* Resource-constrained IoT devices can be authenticated across domains in short time.

IV. Proposed Scheme

This section first briefly introduces the basic building blocks, and then explains the scheme details to provide reliable intra- and inter-domain A&A for devices, which includes system initialization, contract-based mutual access control agreement, intradomain authentication (INAU), anonymous cross-domain authentication, and voting-based identity tracing.

TABLE I
Notations Frequently Used in This Article

Parameter	Description
λ	a security parameter
G	a cyclic group of prime order q
q	a prime number
g	a generator of G
H	a cryptographic hash function
(vsk_i, vpk_i)	private/public key of PAS i for mutual access
(hsk_i, hpk_i)	private/public key of PAS i for pseudonym generation
(dsk_i, dpk_i)	private/public key of device i for inter-domain authentication
$LicsList$	an authorization list consists of vpks
c	ciphertext of vpk

A. Building Blocks

1) Threshold Cryptosystem: The threshold cryptosystem can safeguard a secret by dividing it into N parts and giving each part to a different server. The secret can be retrieved by using t or more than t servers, and fewer than t servers cannot collaborate to get any information about the secret. Further details can be found in [41].

2) Pedersen Commitment: The Pedersen commitment is of the form $\mathrm{Com}_{ck}(m; r) = c = g^r h^m$, where G is a group of prime order q whose generator is g, r is a random number chosen from Z_q^*, $h \in G, m \in Z_q$, and $ck = h$. They are perfectly hiding and computationally binding assuming the discrete logarithm (DL) assumption holds. In addition, Pedersen commitment is homomorphic, for all correctly generated ck, m, m', r, and r', the following equation holds:

$$\mathrm{Com}_{ck}(m; r) \cdot \mathrm{Com}_{ck}\left(m'; r'\right) = \mathrm{Com}_{ck}\left(m + m'; r + r'\right).$$

Further details can be found in [42].

3) Noninteractive Zero-Knowledge Proofs: Noninteractive zero-knowledge proofs (NIZKs) provide the prover with the ability to convince a verifier that a statement is true after sending it one way by generating a cryptographic proof without revealing the associated secret. Further proof generation methods based on Σ protocols for composite statements can be found in [43].

B. System Initialization

At the beginning of the whole system construction, some initialization operations are required among the participating domains, including the common configuration of the system, key negotiation and information sharing of the authentication scheme, and cryptographic algorithm for each domain. The notations frequently used are shown in Table I.

1) Common Configuration: The system generates a cyclic group G of prime order q and a generator g through a probabilistic polynomial time algorithm on input 1^λ, where λ is a security parameter. A hash function H is defined according to Fiat–Shamir heuristic [44]. The PAS of each domain obtains these public parameters. In addition, the current system is equipped with N VSs, which are responsible for verifying operations and forming blockchain.

2) Key Negotiation: PAS first randomly generates an asymmetric key pair (vsk, vpk) for mutual access. Specifically, PAS i randomly generates a private key $\mathrm{vsk}_i \in Z_q^*$, while the corresponding public key is derived from $\mathrm{vpk}_i = g^{\mathrm{vsk}_i}$.

TABLE II
FUNCTIONS DEFINED IN THE CONTRACT

Function	Description
Init	PAS i initializes the mutual access contract (STATE 0)
Apply	PAS j applies for mutual access (STATE 0→1)
Authorize	PAS i authorizes mutual access (STATE 1→2)
Confirm	PAS j ends the contract through authorization (STATE 2→3)

The PAS also needs to generate a pair of asymmetric keys with expiration dates for pseudonym generation. For PAS i, it will randomly generate the private key $\text{hsk}_i \in Z_q^*$ and the public key $\text{hpk}_i = g^{\text{hsk}_i}$. Then, PAS i chooses a $(t-1)$-order random polynomial $f(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$ for hsk_i, where $t = \lfloor N/2 \rfloor + 1$ and $f(0) = a_0 = \text{hsk}_i$. In this way, the private key hsk_i can be replaced by a set of masks: $\{m_j = f(j),\ j \in [1, N]\}$. Then, $D_k = g^{a_k}$ is computed. PAS i sends $\{\text{Verify}, \text{hpk}_i, m_j, \{D_k\}_{k=1}^{t-1}\}$ to the VS_j, $j \in [1, N]$ via encrypted transmission or private channel to request verification of the correctness of masks.

When VS_j receives a verification request, it computes $M_j = g^{m_j}$ and verifies whether the following equation holds:

$$M_j = \text{hpk}_i \cdot \prod_{k=1}^{t-1} (D_k)^{j^k}. \tag{1}$$

If the mask passes the verification, then VS_j saves $\{\text{hpk}_i, m_j, \{D_k\}_{k=1}^{t-1}\}$ locally and calls the smart contract to update the ledger data $\{\text{hpk}_i, \text{vote} + 1, \text{Timestamp}\}$ in the blockchain, where vote has an initial value of 0 in the ledger. After the above process, VS_j returns to notify PAS i that the operation is finished. PAS i queries the blockchain for the vote and Timestamp values corresponding to hpk_i. If vote is greater than $\lfloor N/2 \rfloor$, then the key pair $(\text{hpk}_i, \text{hsk}_i)$ is valid; otherwise, PAS does not have the ability to hide the identity for the device in interdomain authentication.

3) Domain Information Sharing: To ensure that authentication and communication between domains are not limited by the original configuration of the domain, the authentication scheme and cryptographic algorithms used by each domain need to be shared between PASs during system initialization. A possible format could be {IBC, Hash: SHA-256, EncryptScheme: RSA, ...,}. At the end of system initialization, the PAS of each domain shall collectively maintain a table of data containing information on the authentication schemes and cryptographic algorithms used by all participating domains.

C. Contract-Based Mutual Access Control Agreement

To achieve DT and DI, our scheme drives mutual access control agreements between PASs in different domains through the smart contract deployed on the consortium blockchain. Our smart contract contains four functions, as shown in Table II and Algorithm 1.

The function ***Init*** initializes the authorization list LicsList of each PAS and the current state of PAS (STATE). There are four states in total, indicating the different phases of how two PASs, e.g., PASs i and j shown in Fig. 3, to reach a mutual access control agreement. State 3 is the last one where the two

Algorithm 1: The Contract for Mutual Access

***Init*:**
- Intialize LicsList_i;
- Set STATE = 0;

***Apply*:**
- Upon receiving $(\text{vpk}_j, \text{LC}_j, \text{RDR}_j, \text{ADR}_j, \text{DS})$ from PAS j;
- Assert STATE == 0;
- Assert Check(LC_j);
- Store(vpk_j, RDR_j, ADR_j, DS);
- STATE = 1;

***Authorize*:**
- Upon receiving $(\text{vpk}_i, \text{RDR}_i, \text{ADR}_i)$ from PAS i;
- Assert STATE == 1;
- Assert $\text{RDR}_j \in \text{ADR}_i$;
- Assert $\text{RDR}_i \in \text{ADR}_j$;
- Add vpk_j to the LicsList_i;
- STATE = 2;

***Confirm*:**
- Upon receiving ('confirm') from PAS j;
- Assert STATE == 2;
- Add vpk_i to the LicsList_j;
- STATE = 3;

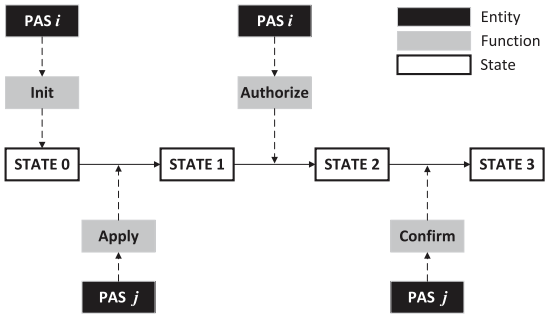


Fig. 3. State transitions in the contract.

PASs reach a mutual access control agreement. In Fig. 3, PAS i triggers ***Init*** at system initialization.

In the contract, when PAS j wants to reach a mutual access control agreement with PAS i, it needs to first submit to the blockchain its own mutual access public key vpk_j, legal credentials (LC_j), required data range (RDR_j), accessible data range (ADR_j) within its own domain, and the target domain (DS). By triggering the function ***Apply***, the blockchain verifies the legal correctness of LC_j (denoted as Check(·) in Algorithm 1), and saves this data [denoted as Store(·)] and updates the state after passing the verification.

After the successful state update, PAS i targeted by DS can trigger the function ***Authorize*** by submitting its own vpk_i, RDR_i and ADR_i. ***Authorize*** determines whether PAS i and PAS j match their mutual requested and accessible data ranges and adds vpk_j of PAS j to LicsList_i of PAS i if they meet the match mutually. PAS j can add vpk_i to its own LicsList_j

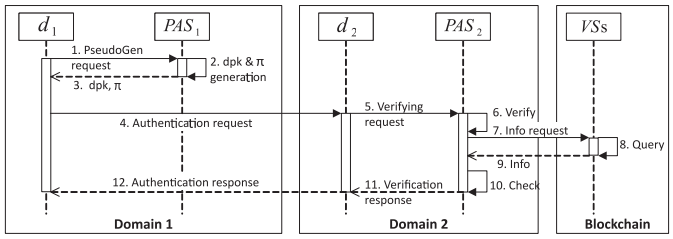


Fig. 4. Overview of the NIZK-based pseudonym generation process. d_1 in domain 1 deployed with PKI is authenticated by d_2 in domain 2 deployed with IBC.

by actively triggering the function ***Confirm*** through sending a "confirm" request. After a successful end of the smart contract, each PAS has a LicsList consisting of vpks of other PASs that have reached mutual access control agreements.

D. Intradomain Authentication

In the INAU phase, our scheme will not affect the authentication process of the scheme originally deployed in each domain. That is, the authentication of the device within the domain will be carried out according to the original scheme. For example, in a domain deployed with a PKI-based scheme, the devices affiliated to the domain request CA of the domain to complete the INAU. In another domain deployed with an IBC-based scheme, the affiliated devices send information to PAS (KGC) of the domain to get the corresponding private key for subsequent INAU. It is worth noting that in our proposed scheme, the devices that have completed the authentication in the domain still retain the relevant authority. Therefore, unlike other schemes in [6], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], and [37] that require reauthentication of all devices after being deployed, we can directly use the existing database, which makes it unnecessary to reauthenticate the devices which have already completed INAU. This not only provides excellent scalability but also greatly reduces the deployment cost and time overhead of the scheme.

E. Anonymous Interdomain Authentication

This section describes the interdomain authentication process in the proposed scheme in detail with an example shown in Fig. 4 for illustration. The system has been initialized before the process begins and a mutual access control agreement has been reached between domain 1 and domain 2. The device d_1 in domain 1 has completed INAU and has a valid certificate (e.g., X509 certificate) issued by CA of domain 1. In the example, we show how d_1 is authenticated by d_2 in domain 2 deployed with IBC. The process for d_1 to authenticate d_2 is the same.

1) NIZK-Based Pseudonym Generation: At the beginning of interdomain authentication, if d_1 does not generate a pseudonym or wants to change it, it can send a pseudonym generation request message to PAS_1 in the domain by using the intradomain credential (certificate in this case). The message should contain the request type identifier and the target access domain, e.g., $\{\text{PseudoGen}, \text{Domain}_2\}$.

After PAS_1 receives the request and verifies the identity of the device, it checks whether LicsList_2 contains vpk_1 of domain 1, which means that whether there is a mutual access control agreement built with the target domain, Domain_2. Also, PAS_1 checks if its own hpk_1 is valid. If all steps are correct, PAS_1 uses the cryptographic algorithm which is adopted by domain 2 for INAU, to generate a random asymmetric key pair $(\text{dsk}_1, \text{dpk}_1)$ for d_1, where dpk_1 is used as the interdomain identity for d_1. PAS_1 chooses a random number $r \in Z_q$ and encrypts its mutual access public key vpk_1 as follows:

$$c = \left(\text{hpk}_1{}^r, g^r \cdot \text{vpk}_1\right) = (u, v). \tag{2}$$

Then, PAS_1 generates an NIZK proof, π, which proves three properties.

1) c is a valid ciphertext that is encrypted under the pseudonym generation public key hpk_1.
2) The ciphertext is an encryption of vpk_1 which is in LicsList_2.
3) dpk_1 is the identity of d_1 and has not been tampered with.

The three properties can be written mathematically as follows:

$$\begin{aligned}\mathcal{R} = \Big\{\big((\text{hpk}_1, \text{dpk}_1, \text{LicsList}_2, c), (\text{vsk}_1, r)\big) : \text{vsk}_1 \in Z_q^* \\ \wedge\, \text{vpk}_1 = g^{\text{vsk}_1} \in \text{LicsList}_2 \subset G \\ \wedge\, c = \left(\text{hpk}_1{}^r, g^r \cdot \text{vpk}_1\right)\Big\}.\end{aligned} \tag{3}$$

After that, PAS_1 returns $\{\text{dpk}_1, \text{hpk}_1, \pi\}$ to d_1. When d_1 receives the message, it can send $\{\text{Auth}, \text{dpk}_1, \text{hpk}_1, \pi\}$ to d_2 to request authentication, where Auth is the request type identifier. Upon receiving a message with the Auth identifier, d_2 forwards it to PAS_2 for verification. PAS_2 verifies the proof π. If the verification is passed, then a query request containing hpk_1 is initiated and sent to the blockchain. The query is based on hpk_1, and the query result is sent to and checked by PAS_2. If hpk_1 does not expire and vote is greater than $\lfloor N/2 \rfloor$, PAS_2 returns a successful verification result to d_2; otherwise, or any of the above steps fails, PAS_2 returns d_2 with a failed result. Then, d_2 knows that whether d_1 passes the authentication based on the result returned by PAS_2 and returns the result to d_1. If the authentication is successful, d_2 saves dpk_1 as the public key for subsequent communication with d_1, which can be alternatively done by PAS_2 if d_2 is a resource-constrained device.

2) Lightweight and Fast Authentication: Note that the above NIZK-based pseudonym generation operation is not required for every authentication. If d_1 already has a valid pseudonym dpk_1 and does not need to be replaced, then a lightweight and fast authentication between d_1 and d_2 is performed. Specifically, assume that M is the message that d_1 wants to deliver to d_2. d_1 first submits M to PAS_1 to ask for its signature over M [denoted as $\text{Sig}(M)$] using dsk_1 based on the cryptographic algorithm adopted by domain 2 (note that a high-performance d_1 can perform the signature operation by itself using dsk_1). Then d_1 sends $\{\text{dpk}_1, \text{Sig}(M), M\}$ to d_2, and d_2 then sends a query and verification request to PAS_2 based on the identity of the received message dpk_1 (high-performance devices can locally query dpk_1 and perform the

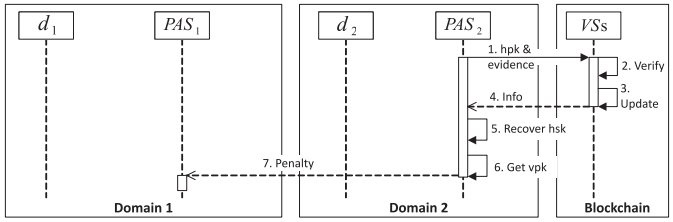


Fig. 5. Overview of voting-based identity tracing process. PAS_2 in domain 2 finds the source domain and penalizes it based on the pseudonym.

verification of the signature). If the signature passes the verification, d_2 encrypts the reply message for d_1 using dpk_1 and returns the encrypted message to d_1. Then, d_1 can decrypt the message using dsk_1. Similarly, resource-constrained devices can hand over the relevant encryption/decryption operations to their corresponding PASs for execution. With the above operations, d_1 and d_2 can achieve lightweight and fast anonymous authentication and resource interaction.

F. Voting-Based Identity Tracing

When risky behavior of an accessing device is detected, the PAS of the accessed domain in our scheme is able to trace the source domain of the device based on the pseudonym and penalize the source domain. Fig. 5 shows an example for illustration, where PAS_2 sends hpk_1, which is bound to the malicious device pseudonym dpk_1, and associated evidence to each VS via a secure path. After a VS, say VS_j, confirms the received evidence, it uses the received hpk_1 to look up the locally stored $\{\text{hpk}_1, m_j, \{D_k\}_{k=1}^{t-1}\}$ and returns them to PAS_2 through the secure path, while invoking the smart contract to update the vote corresponding to the aforementioned hpk_1, which makes a penalty by decreasing vote by one. PAS_2 verifies the correctness of the received group of masks using (1). After receiving the $t = \lfloor N/2 \rfloor + 1$ set of masks from t VSs, the private key hsk_1 can be retrieved based on the threshold technique [41] by

$$\text{hsk}_1 = \sum_{j=1}^{t} \left(m_j \cdot \prod_{k=1, k \neq j}^{t} \frac{k}{k-j} \right). \tag{4}$$

After getting the private key hsk_1, PAS_2 successfully traces the source domain vpk_1 of the malicious device by

$$\text{vpk}_1 = \text{Dec}_{\text{hsk}_1}(c) = v \cdot u^{-\frac{1}{\text{hsk}_1}}. \tag{5}$$

PAS_2 queries its own LicsList_2 based on vpk_1, identifies the source domain, and thus makes a financial or operational penalty to the PAS of that domain, as well as broadcasting it to the other domains in the system. After this process, the pseudonym generation public key hpk_1 of the penalized PAS is invalidated, because the corresponding vote on the blockchain of hpk_1 is fewer than $\lfloor N/2 \rfloor$. Before a new identity-hiding key can be regenerated and verified, the penalized PAS needs to guarantee that the malicious device is rectified in that domain and notify the other domains of this rectification.

V. Security Analysis

This section first analyzes the proposed NIZK proof π to show its completeness, special soundness, and special honest verifier zero knowledge. Then, we analyze how this scheme achieves domain privacy, AC, and secure A&A.

A. Security Analysis of NIZK Proof

An NIZK proof based on the Σ protocol needs to satisfy three properties [45].

1) *Completeness:* If the prover ($\mathcal{P}$) knows a witness w for the statement s, then it should be able to convince the verifier ($\mathcal{V}$).
2) *Special Soundness:* If $\mathcal{P}$ does not know a witness w for s, it should not be able to convince $\mathcal{V}$. Formally, there exists an extractor that can extract w for any x and any pair of accepting conversations on input x with different random challenges ch and ch'. Moreover, the protocol is n-special sound if the number of accepting conversations is n.
3) *Special Honest Verifier Zero Knowledge (SHVZK):* The protocol should not reveal any information about the w of $\mathcal{P}$. Formally, there exists a probabilistic polynomial time simulator $\mathcal{S}$ that is able to output, for input x and a random challenge ch, an accepting conversation with the same probability distribution as the conversation between the honest $\mathcal{P}$ and the honest $\mathcal{V}$ on input x.

Lemma 1: The Σ protocol for the relation (3) has perfect completeness, special soundness, and SHVZK.

Proof (Completeness): To simplify the description, we first define the operation $\text{Enc}_{\text{hpk}_1}(m : r) = (\text{hpk}_1^r, g^r \cdot m)$. Based on Pedersen commitment to multiple elements, we define $ck = (h_1, \ldots, h_n)$, $(h_1, \ldots, h_n \in G)$, and $\text{Com}_{ck}(m_1, \ldots, m_n)$, which returns the error symbol when $(m_1, \ldots, m_n) \notin Z_q^n$, otherwise the value is $g^r \prod_{i=1}^{n} h_i^{m_i}$. Moreover, we assume LicsList_2 is a list which contains N' ($N' \leq N$) vpks (denoted as $\{\text{vpk}_0, \text{vpk}_1, \ldots, \text{vpk}_{N'-1}\}$) and the index of vpk_1 is ℓ. We pad the list with copies of the last vpk to make N' reach $N = n^m$, keeping the values of n and m.

Without loss of generality, let PAS_1 work as the $\mathcal{P}$. It first chooses four random numbers $\alpha, \beta, \gamma, \epsilon \in Z_q$ and generates $ek = g^{\tau}$ for $\tau \in Z_q^*$. Then, it computes

$$\begin{aligned} d &= \text{Enc}_{ek}\left(g^{\text{vsk}_1}; \beta\right) \\ A &= \text{Enc}_{\text{hpk}_1}\left(g^{\alpha}; \gamma\right), B = \text{Enc}_{ek}\left(g^{\alpha}; \epsilon\right) \\ c_0 &= d \cdot \text{Enc}_{ek}\left(\text{vpk}_0{}^{-1}; 0\right), \ldots \\ c_{N-1} &= d \cdot \text{Enc}_{ek}\left(\text{vpk}_{N-1}{}^{-1}; 0\right). \end{aligned} \tag{6}$$

Next, a set of random numbers $\dot{\alpha}, \dot{\beta}, \dot{\gamma}, \dot{\epsilon}, a_{j,1}, \ldots, a_{j,n-1} \in Z_q$ are chosen by PAS_1, where $j = 0, \ldots, m-1$ and $\forall j : a_{j,0} = -\sum_{i=1}^{n-1} a_{j,i}$. After that, PAS_1 computes the following elements:

$$\begin{aligned} \dot{A} &= \text{Com}_{ck}\left(a_{0,0}, \ldots, a_{m-1,n-1}; \dot{\alpha}\right) \\ \dot{B} &= \text{Com}_{ck}\left(\delta_{\ell_0,0}, \ldots, \delta_{\ell_{m-1},n-1}; \dot{\beta}\right) \end{aligned}$$

$$\dot{C} = \mathrm{Com}_{ck}\left(\left\{a_{j,i}\left(1-2\delta_{\ell_j,i}\right)\right\}_{j,i=0}^{m-1,n-1}; \dot{\gamma}\right)$$
$$\dot{D} = \mathrm{Com}_{ck}\left(-a_{0,0}^2, \ldots, -a_{m-1,n-1}^2; \dot{\epsilon}\right) \quad (7)$$

where $\delta_{\ell,i}$ is the Kronecker delta [46], which has a value of 1 when $\ell = i$ and 0 for the rest. Meanwhile, $\delta_{\ell,i}$ is equal to $\prod_{j=0}^{m-1}\delta_{\ell_j,i_j}$, where $\ell = \sum_{j=0}^{m-1}\ell_j n^j$ and $i = \sum_{j=0}^{m-1} i_j n^j$ are the n-ary representations of ℓ and i, respectively, and ℓ_j is the index of the only 1 in the sequence $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,n-1})$.

PAS_1 calculates the challenge $ch = \mathrm{H}(A\|B\|\dot{D}\|\mathrm{dpk}_1)$ based on Fiat–Shamir heuristic [44]. Then, it calculates $\forall j, i{:}f_{j,i} = \delta_{\ell_j,i}\cdot ch + a_{j,i}$, $z_A = \dot{\beta}\cdot ch + \dot{\alpha}$, $z_C = \dot{\gamma}\cdot ch + \dot{\epsilon}$. Afterward, it calculates $\dot{G}_k = \prod_{i=0}^{N-1} c_i^{p_{i,k}}\cdot \mathrm{Enc}_{ek}(1;\rho_k)$, where random numbers $\rho_k \in Z_q$ and $k = 0, \ldots, m-1$. $\{p_{i,k}\}_{k=0}^{m-1}$ are some coefficients depending on ℓ and $a_{j,i}$ in the following polynomial and can be computed by PAS_1 independently of x [45]:

$$\begin{aligned} p_i(x) &= \prod_{j=0}^{m-1}\left(\delta_{\ell_j,i}x + a_{j,i}\right) \\ &= \prod_{j=0}^{m-1}\delta_{\ell_j,i_j}x + \sum_{k=0}^{m-1} p_{i,k}x^k \\ &= \delta_{\ell,i}x^m + \sum_{k=0}^{m-1} p_{i,k}x^k. \end{aligned} \quad (8)$$

Then, PAS_1 computes the elements $z = \beta\cdot ch^m - \sum_{k=0}^{m-1}\rho_k\cdot ch^k$, $z_s = \mathrm{vsk}_1\cdot ch + \alpha$, $z_a = r\cdot ch + \gamma$, $z_b = \beta\cdot ch + \epsilon$. Finally, PAS_1 sends the proof π as follows:

$$\pi = \left\{ek, ck, c, n, m, d, A, B, \dot{A}, \dot{B}, \dot{C}, \dot{D}, \left\{\dot{G}_k\right\}_{k=0}^{m-1}, \left\{f_{j,i}\right\}_{j=0,i=1}^{m-1,n-1}, z_A, z_C, z, z_s, z_a, z_b\right\}. \quad (9)$$

Upon receiving the proof π, PAS_2 first checks hpk_1, $ek \in G^*$, $d, A, B \in G^2$, $ck, \dot{A}, \dot{B}, \dot{C}, \dot{D}, \dot{G}_0, \ldots, \dot{G}_{m-1} \in G$ and $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C, z, z_s, z_a, z_b \in Z_q$. Next, PAS_2 computes $ch = H(A\|B\|\dot{D}\|\mathrm{dpk}_1)$ and $\forall j{:}f_{j,0} = ch - \sum_{i=1}^{n-1} f_{j,i}$, and verifies the proof π by checking whether the following equations hold:

$$\begin{aligned} &c^{ch}A = \mathrm{Enc}_{\mathrm{hpk}_1}\left(g^{z_s}; z_a\right),\ d^{ch}B = \mathrm{Enc}_{ek}\left(g^{z_s}; z_b\right) \\ &\dot{B}^{ch}\dot{A} = \mathrm{Com}_{ck}\left(f_{0,0}, \ldots, f_{m-1,n-1}; z_A\right) \\ &\dot{C}^{ch}\dot{D} = \mathrm{Com}_{ck}\left(\left\{f_{j,i}\left(ch - f_{j,i}\right)\right\}_{j,i=0}^{m-1,n-1}; z_C\right) \\ &\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{m} f_{j,i_j}}\cdot\prod_{k=0}^{m-1}\dot{G}_k^{-ch^k} = \mathrm{Enc}_{ek}(1;z). \end{aligned} \quad (10)$$

If any of the above elements or equations does not hold, the proof verification fails. Otherwise, the proof π passes the verification of PAS_2. Therefore, the completeness of the proof holds.

(Special Soundness): We assume that the extractor inputs z_a, z_b, z_s and z_a', z_b', z_s' to different challenges ch and ch'. Then, it can get

$$\begin{aligned} c^{ch-ch'} &= \mathrm{Enc}_{\mathrm{hpk}_1}\left(g^{z_s - z_s'}; z_a - z_a'\right) \\ d^{ch-ch'} &= \mathrm{Enc}_{\mathrm{hpk}_1}\left(g^{z_s - z_s'}; z_b - z_b'\right) \end{aligned} \quad (11)$$

and thus obtain $\mathrm{vsk}_1 = (z_s - z_s')/(ch - ch')$, $r = (z_a - z_a')/(ch - ch')$ through the verification equations. Next, the extractor needs to extract the index of vpk_1 in $\mathrm{LicsList}_2$. Suppose there are $m+1$ ($m > 1$) different accepting responses $(f_{j,i}^{(0)}, z^{(0)}), \ldots, (f_{j,i}^{(m)}, z^{(m)})$ with respect to $m+1$ different challenges $ch^{(0)}, \ldots, ch^{(m)}$. Similarly, the extractor can extract $\dot{B}$, $\dot{A}$ and $\delta_{\ell_j,i}, a_{j,i}$ for them. Then, it can compute the polynomials (8) and derive $\tilde{G}_k$ values from equation $c_\ell^{ch^m}\cdot\prod_{k=0}^{m-1}\tilde{G}_k^{ch^k} = \mathrm{Enc}_{ek}(1;z)$. Consider a Vandermonde matrix with row e for $(1, ch^{(e)}, \ldots, ch^{(e)^m})$. Since the values of $ch^{(e)}$ are distinct, this matrix is invertible. The extractor can obtain a linear combination $\theta_0, \ldots, \theta_n$ of the rows producing the vector $(0, \ldots, 0, 1)$ and deduce

$$c_\ell = \prod_{e=0}^{m}\left(c_\ell^{\left(x^{(e)}\right)^m}\cdot\prod_{k=0}^{m-1}\tilde{G}_k^{\left(x^{(e)}\right)^k}\right)^{\theta_e} = \mathrm{Enc}_{ek}\left(1; \sum_{e=0}^{m}\theta_e z^{(e)}\right).$$

It shows $\mathrm{vpk}_1 \in \mathrm{LicsList}_2$ and the index ℓ. Therefore, the proof has $(m+1)$-special soundness.

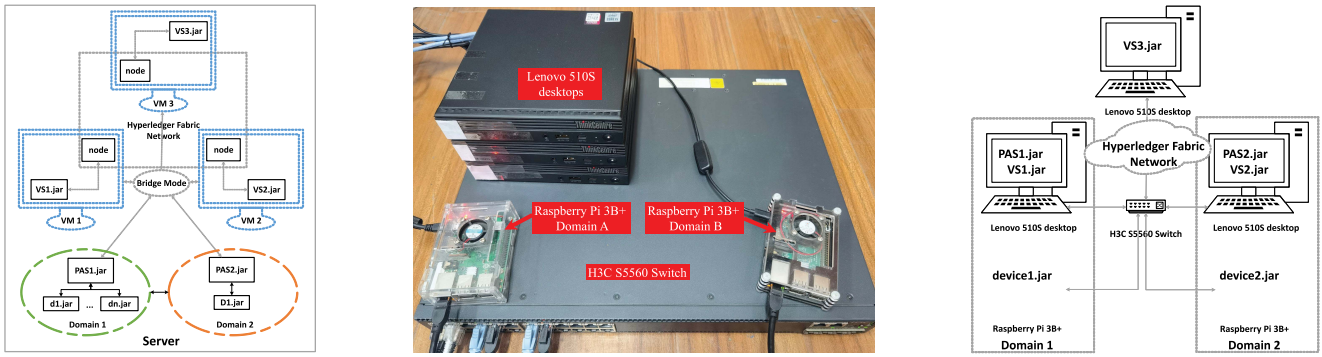
Special Honest Verifier Zero Knowledge: We construct an SHVZK simulator $\mathcal{S}$ and give a random challenge ch. It randomly selects $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C, z, z_s, z_a, z_b \in Z_q$, $\dot{C}, \dot{B}, \dot{G}_1, \ldots, \dot{G}_{m-1} \in G$, $d \in G^2$ and computes

$$\begin{aligned} f_{j,0} &= ch - \sum_{i=1}^{n-1} f_{j,i} \\ \dot{A} &= \mathrm{Com}_{ck}\left(f_{0,0}, \ldots, f_{m-1,n-1}; z_A\right)\dot{B}^{-ch} \\ \dot{D} &= \mathrm{Com}_{ck}\left(\left\{f_{j,i}\left(ch - f_{j,i}\right)\right\}_{j,i=0}^{m-1,n-1}; z_C\right)\dot{C}^{-ch} \\ A &= \mathrm{Enc}_{\mathrm{hpk}_1}\left(g^{z_s}; z_a\right)c^{-ch} \\ B &= \mathrm{Enc}_{ek}\left(g^{z_s}; z_b\right)d^{-ch}. \end{aligned} \quad (12)$$

Similarly, it can also compute $\dot{G}_0$ from the last verification equation. Obviously, in both simulations and real proofs, $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C, z, z_s, z_a, z_b, \dot{C}, \dot{B}, \dot{G}_1, \ldots, \dot{G}_{m-1}$ and d are independent, uniformly random, and uniquely determine $\{f_{j,0}\}_{j=0}^{m-1}, \dot{A}, \dot{D}, A, B, G_0$, so the simulation is perfect and has quasi-unique responses. ■

B. Analysis of Security Properties

1) Domain Privacy: For the risk of privacy-invasive malicious behavior of PAS, our scheme implements domain-level anonymity in the interdomain authentication process. The public information of PAS for each domain is vpk and hpk. vpk can identify a domain and form a LicsList with the vpk of other PASs that have reached mutual access control agreements. hpk is a random element generated by PAS and sent to VS to update the blockchain ledger. It is not visible to other PASs in this process, and the data updated by VSs on the ledger does not contain any identification of this PAS, so hpk is not bound to the real identity of the PAS. In the interdomain authentication phase, we successfully hide the information of vpk by using an effective NIZK proof. Taking Fig. 4 for example, the message sent by the authentication initiator d_1 contains only dpk_1, hpk_1, and π. dpk_1 is a random element generated by PAS_1 based on the encryption algorithm adopted by domain 2, and similar to hpk_1, it does not contain



(a) (b) (c)

Fig. 6. Physical and virtual prototypes. (a) Virtual prototype. (b) Physical prototype. (c) Corresponding topology of the physical prototype shown in (b).

any information about domain 1. None of the parameters contained in π relate to domain 1. Therefore, there is no way for d_2 and PAS_2 in domain 2 to know which domain the message sender d_1 comes from. In addition, the PAS of each domain can generate new hpk through the *Key Negotiation* process (as shown in Section IV-B2), and the device can also request new dpk and π from PAS when a pseudonym needs to be changed. This ensures that it is difficult for PAS to infer sensitive information about any visitors or their domains from their records.

2) Accountability: Our scheme uses the threshold technique [41] to design the voting-based identity tracing, which solves the security problem of not being able to trace and punish malicious devices due to complete anonymity. PAS can submit evidence related to malicious devices to VSs. After voting by VSs, PAS uses (4) to recover the decryption private key hsk to get vpk. PAS has previously verified the correctness of NIZK proof π corresponding to the malicious device dpk, which indicates that PAS has authorized the domain where the malicious device is located. Thus, PAS can query vpk from LicsList and use it to obtain information of the domain where the malicious device is located, so as to initiate a punishment against that domain. On the other hand, for achieving a possible collusion attack on the auditing process, the conspirators need to collude with more than half of malicious VSs in the whole system to prevent the correct restoration of hsk and thus deceive the auditing of PAS. However, such a collusion is difficult to be realized in reality.

3) Secure A&A: With the benefit of the introduction of consortium blockchain, our scheme eliminates the dependence on the third-party organizations in interdomain A&A, allowing direct A&A between two domains. To prevent security risks to the domain caused by the opening data on the blockchain, the data we store on the consortium blockchain does not contain secrets and privacy. For man-in-the-middle attacks that devices may encounter in interdomain authentication, the message contents sent by both authentication parties during the authentication process are public and tamper-proof because of the use of asymmetric keys and NIZK proofs. When an attacker intercepts the message content and tampers with it, the modified message will not pass the receiver's authentication. On the other hand, if an attacker replays the authentication request from d_1 to d_2, i.e., $\{\text{Auth}, \text{dpk}_1, \text{hpk}_1, \pi\}$ (as mentioned in Section IV-E1), it cannot decrypt the data returned by d_2, which is encrypted by using dpk_1.

TABLE III
FUNCTION COMPARISON WITH EXISTING WORKS

Scheme \ Function	INAU	CSAU	DT	DI	PP	AC
IRBA [28]	✓	✓	✓	✗	✓	✗
CAKA [36]	✓	✓	✓	✓	✓	✗
HIAS [32]	✓	✓	✗	✓	✓	✗
BASA [6]	✓	✓	✓	✗	✓	✗
BlockCAM [24]	✓	✓	✓	✗	✗	✗
BTCAS [37]	✓	✓	✓	✓	✗	✗
Ours	✓	✓	✓	✓	✓	✓

VI. PERFORMANCE EVALUATION

A. Functionality

We first compare the functional differences between our scheme and existing works in Table III, including INAU, interdomain authentication (CSAU), DT, DI, PP, and AC. Note that although most existing works also consider PP and provide anonymity mechanisms, their definitions are different from ours. In the existing work, the server side of the accessed domain can still track the data of the access device, while our scheme makes the access device anonymous to both the server and the device side of the accessed domain.

B. Scheme Implementation and Experiments

In order to evaluate the performance of the proposed scheme, we implement two proof-of-concept prototypes in a physical testbed and virtual machine, respectively, both using Java and the open-source federated chain formed based on Hyperledger Fabric v2.3 [47]. The topology of the virtual prototype is shown in Fig. 6(a) and contains two trusted domains and several intradomain devices. The whole prototype runs on a host server equipped with i9 7900X CPU @3.30 GHz and 128-GB RAM. The entity operations of the PAS and IoT devices are running directly in the server, and those of three VSs are running in three virtual machines (VMs) built based on VMware Workstation 16 Pro [48] equipped with 8-GB memory. The three VMs are working in the same local area network (LAN) by setting the network to the bridge mode.

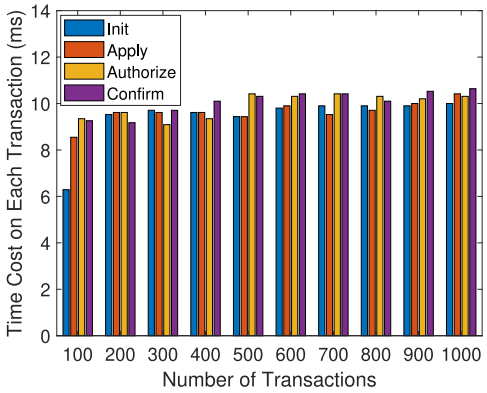


Fig. 7. On-chain cost of core functions defined in the contract for mutual access.

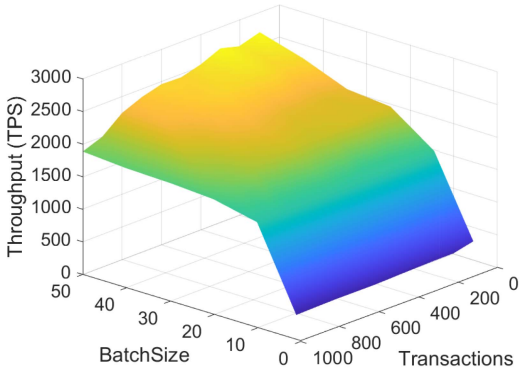


Fig. 8. On-chain performance of the contract-based mutual access control agreement.

TABLE IV
EXECUTION TIME OF OUR SCHEME IN EACH ENTITY (UNIT: MILLISECOND)

Entity	PR	AU	PR+AU
d_1	35.842	28.251	64.093
PAS_1	156.855	11.259	168.114
d_2	9.719	24.179	33.898
PAS_2	150.300	9.885	160.185
VS_j	18.675	/	18.675

TABLE V
EXECUTION TIME COMPARISON (UNIT: MILLISECOND)

Scheme	IoT devices	Servers	Blockchain
Our Scheme	97.991	328.299	18.675
BTCAS	460.676	61.299	58.095
BASA	5,243.236	186.337	51.015

The physical prototype shown in Fig. 6(b) contains two trusted domains (each containing a PAS and an IoT device) and a server cluster consisting of three VSs. The corresponding topology is shown in Fig. 6(c). The prototype consists of three Lenovo 510S desktops equipped with i5 10400 CPU @2.9 GHz and 8-GB RAM, and two Raspberry Pi 3B+ devices equipped with Broadcom Cortex-A53 CPU @1.4 GHz and 1-GB RAM. The three desktops are responsible for running the PASs and VSs in the trusted domains, and the Raspberry Pi 3B+ devices are working as IoT devices. All machines are connected in the same LAN through an H3C S5560 switch. In both prototypes, all entity programs are packaged as independent Jar files. The HTTP protocol is adopted for the communication between entities.

In addition, we implement BASA [6] and BTCAS [37] for performance comparison with our scheme under the same settings and security level. Their function comparison is shown in Table III. BASA is a representative work for A&A based on IBC and blockchain. It provides secure and effective PP with lower computation and communication overhead than PKI- and other IBC-based A&A works. BTCAS achieves lightweight and DI for A&A by combining hashing algorithms and blockchain. It has extremely low computation and communication overhead but does not provide anonymity. All intradomain keys are generated uniformly using RSA-1024, while SHA1withRSA is used for signature verification of messages.

1) Computation Cost: Computation cost is evaluated in terms of execution time and consists of two parts: 1) authorization and 2) authentication.

For the authorization part, we perform ten groups of experiments for each core function shown in Algorithm 1 in the virtual prototype based on the default configuration of Hyperledger Fabric v2.3, with the number of concurrent invocations increased from 100 to 1000 by a step of 100. Every experiment in each group is run for 20 trials, and the results are averaged over all trials and shown in Fig. 7. As can be seen in the figure, the time cost of the four core functions increases with the number of concurrent transactions, but which can still be concluded in about 10.5 ms on the chain even when the number of concurrent transactions reaches 1000. To further evaluate the impact of block generation in the blockchain on the transaction throughput of this smart contract, we adjust the block size parameter, i.e., *BatchSize*, in Hyperledger Fabric v2.3 and conduct experiments of the same order of magnitude. Fig. 8 shows the variation of transaction throughput [unit: transactions per second (TPS)] of this smart contract as the number of concurrent transactions increases with different BatchSize. As BatchSize increases, the throughput on the blockchain grows rapidly at first and then becomes stable. It should be noted that the execution of each smart contract (i.e., mutual access control agreement) is an operation done between two domains, and the execution of n transactions on the chain is an operation done two-by-two between $2n$ domains. Therefore, the variation of this part is the impact of the number of domains on the system performance.

For the authentication part, we evaluate each scheme in two parts, PR and AU, representing the preparation phase before authentication (including pseudonym generation, etc.) and the interdomain authentication phase, respectively. We run 200 times of one-way interdomain authentication for each of the three schemes in the physical prototype shown in Fig. 6(b). The average computation cost of each entity in our scheme is shown in Table IV. Based on the results in this table, we show the sum of the execution time of the two IoT devices (i.e., $64.093 + 33.898$) and that of the two PAS servers (i.e., $168.114 + 160.185$) in Table V, in comparison with the other two schemes. As we can see from Table V, our scheme has the lowest computation cost at the device side, which indicates

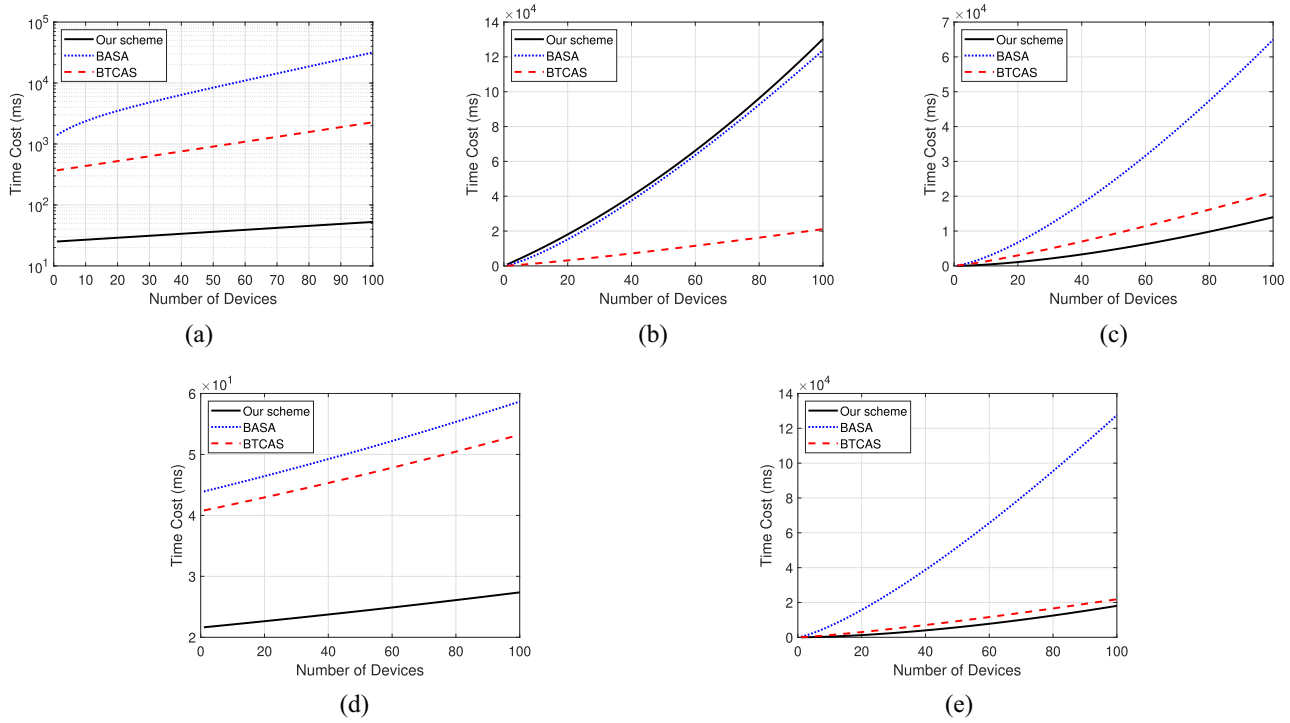


Fig. 9. Comparison of the time consumption of our scheme with other schemes for each entity under different phases (PR: preparation; AU: authentication). (a) Average cost on devices for PR (the y-axis is in the logarithmic scale). (b) Cost on server for PR. (c) Cost on blockchain for PR. (d) Average cost on devices for AU. (e) Cost on server for AU.

that our scheme can be well adapted to energy-constrained IoT devices. The computation cost at the server side is larger than the other two schemes, which is due to the fact that our design transfers the generation and verification of zero-knowledge proofs from the device side to the service side. It should be noted that in reality trusted domains generally have higher performance servers and the overhead incurred by our scheme at the server side is acceptable. An additional note is that since BASA's pseudonym generation task in the PR part is performed by the device side, it requires bilinear pairing operations at each turn, which makes it a significant computation cost to the Raspberry Pi 3B+ devices.

To further evaluate the performance of our scheme under heavy tasks, we execute experiments with concurrent accesses varying from 1 to 100 IoT devices in the virtual prototype shown in Fig. 6(a) (containing two trusted domains). The computation cost at the server and blockchain side is defined as the difference between the authentication start time of the first device and the authentication completion time of the last device. Each IoT device records its computation cost, and the results are averaged. We perform 100 simulations and fit the curve based on the average. The results are shown in Fig. 9. Since the blockchain side of our scheme does not require the computation at the AU phase, we do not compare it with other schemes for fairness. From Fig. 9(a) and (d), we can see that as the number of concurrently requested devices increases, the device-side overhead in our scheme increases very slightly, both being lower than BTCAS and BASA. It should be clarified that the significant increase in the average device cost of BASA in Fig. 9(a) is caused by the large number of devices performing high-overhead computations simultaneously and the insufficient computing capability of the virtual prototype. For the server side, our scheme can approach the performance of BASA in the PR phase [Fig. 9(b)] and outperform other schemes in the AU phase [Fig. 9(e)] for a large number of devices with concurrent accesses. For the performance of the blockchain side in the PR phase [Fig. 9(c)], the computation cost of our scheme is also lower than other schemes since it only requires query operations to be performed in this phase.

2) Communication, Storage, and Deployment Overheads: We also evaluate the communication, on-chain storage, and system deployment overheads. The communication overhead of our scheme contains mainly dpk, hpk, and NIZK proof π, which are forwarded twice from server to device and from device to outer domain server. For a LicsList of size $N = n^m$, π contains $n+1$ elements in G, $m+4$ ciphertexts, 4 Pederson commitments, and $m(n-1)+6$ elements in Z_q. The group and field elements used in this article are approximately the same size, so the communication overhead is $m(n+1)+n+21$ elements in total, i.e., $(n+1)\log_n N+n+21$ elements. Considering the case where the security parameter occupies 1024 bits and the LicsList contains 1024 vpks, the total communication overhead of our scheme is about 13 kB if n is set to 2.

For the on-chain storage overhead, the size of the interaccess contract in our experiments is 11 kB, and the data stored in the ledger varies with the number of domains and the size of its LicsList. Given 1025 domains, with each LicsList containing 1024 vpks, the amount of data stored in the ledger is about 129 MB. It is worth mentioning that we can also further optimize the on-chain data storage through existing techniques, e.g., IPFS [49].

In addition, we analyze the deployment cost theoretically. Our scheme consists of a modified INAU server and a cluster of blockchain servers. It allows the participating domains to outsource the blockchain part to service providers to reduce the cost of intradomain deployment. Also, different from existing works, which may require all participating domains to reauthenticate all devices, our proposal directly utilizes the existing INAU database. New domains can join the multidomain system running our scheme directly and have the ability to handle interdomain A&A, which reduces the deployment time and implementation cost of the scheme.

VII. Conclusion

In this article, we have proposed a blockchain-assisted intra/inter-domain A&A scheme for IoT. The scheme achieves DT and provides a secure way for the domains with different A&A schemes to authorize access to each other. Moreover, it can not only protect the privacy of devices and domains but also allow legitimate auditing of malicious devices outside the domain. The scheme is quite suitable for resource-constrained IoT devices as the on-device authentication cost is low, which is independent of the complexity of the authentication policy. As a generic solution, the proposed scheme can be easily deployed in many IoT applications with multiple domains to enhance their security and DI. For future work, we intend to explore some real-world multidomain IoT applications to deploy our scheme, so that the proposed scheme can be improved more in practicability.

References

[1] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[2] X. Shen et al., "Data management for future wireless networks: Architecture, privacy preservation, and regulation," *IEEE Netw.*, vol. 35, no. 1, pp. 8–15, Jan./Feb. 2021.

[3] *Ericsson Mobility Report*, Ericsson, Stockholm, Sweden, Nov. 2021.

[4] C. Huang et al., "Blockchain-assisted transparent cross-domain authorization and authentication for smart city," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17194–17209, Sep. 2022.

[5] R. Lohiya and A. Thakkar, "Application domains, evaluation data sets, and research challenges of IoT: A systematic review," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8774–8798, Jun. 2021.

[6] M. Shen et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.

[7] S. Valtolina, L. Ferrari, and M. Mesiti, "Ontology-based consistent specification of sensor data acquisition plans in cross-domain IoT platforms," *IEEE Access*, vol. 7, pp. 176141–176169, 2019.

[8] K. A. Awan, I. U. Din, M. Zareei, M. Talha, M. Guizani, and S. U. Jadoon, "HoliTrust-a holistic cross-domain trust management mechanism for service-centric Internet of Things," *IEEE Access*, vol. 7, pp. 52191–52201, 2019.

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," IETF, RFC 5280, 2008.

[10] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Adv. Cryptol.*, 2001, pp. 213–229.

[11] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Adv. Cryptol.*, 2003, pp. 452–473.

[12] J. Wu, M. Dong, K. Ota, M. Tariq, and L. Guo, "Cross-domain fine-grained data usage control service for industrial wireless sensor networks," *IEEE Access*, vol. 3, pp. 2939–2949, 2015.

[13] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 633–645, Jul./Aug. 2016.

[14] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 6, pp. 754–764, Jun. 2010.

[15] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2014, pp. 2728–2733.

[16] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for IoT systems," in *Proc. Workshop IoT Challenges Mobile Ind. Syst.*, 2015, pp. 37–42.

[17] J. Liu, Y. Xiao, and C. P. Chen, "Authentication and access control in the Internet of Things," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, 2012, pp. 588–592.

[18] S. R. Moosavi et al., "SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways," *Procedia Comput. Sci.*, vol. 52, pp. 452–459, Dec. 2015.

[19] B. Ndibanje, H.-J. Lee, and S.-G. Lee, "Security analysis and improvements of authentication and access control in the Internet of Things," *Sensors*, vol. 14, no. 8, pp. 14786–14805, 2014.

[20] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 410–426.

[21] M. Pilkington, "Blockchain technology: Principles and applications," in *Research Handbook on Digital Transformations*. Cheltenham, U.K.: Edward Elgar Publ., 2016.

[22] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, vol. 310, 2016, pp. 1–4.

[23] X. Shen et al., "Blockchain for transparent data management toward 6G," *Engineering*, vol. 8, pp. 74–85, Jan. 2022.

[24] W. Wang, N. Hu, and X. Liu, "BlockCAM: A blockchain-based cross-domain authentication model," in *Proc. IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, 2018, pp. 896–901.

[25] C. Li, Q. Wu, H. Li, and J. Liu, "Trustroam: A novel blockchain-based cross-domain authentication scheme for Wi-Fi access," in *Wireless Algorithms, Systems, and Applications*. Cham, Switzerland: Springer, 2019, pp. 149–161.

[26] S. Qian, L. Chen, and S. Wang, "PKI cross-domain authentication scheme based on advanced PBFT algorithm," *Chin. J. Netw. Inf. Security*, vol. 6, no. 4, pp. 37–44, Aug. 2020.

[27] G. Ali et al., "xDBAuth: Blockchain based cross domain authentication and Authorization framework for Internet of Things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020.

[28] X. Jia, N. Hu, S. Su, S. Yin, and C. Zhang, "IRBA: An identity-based cross-domain authentication scheme for the Internet of Things," *Electronics*, vol. 9, no. 4, p. 634, 2020.

[29] Y. Yao, X. Chang, J. Mišić, V. B. Mišić, and L. Li, "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3775–3784, Apr. 2019.

[30] X. Lan, J. Xu, H. Guo, and Z. Zhang, "One-round cross-domain group key exchange protocol in the standard model," in *Proc. Inf. Security Cryptol.*, 2017, pp. 386–400.

[31] C. Yuan, W. Zhang, and X. Wang, "EIMAKP: Heterogeneous cross-domain authenticated key agreement protocols in the EIM system," *Arab. J. Sci. Eng.*, vol. 42, pp. 3275–3287, Feb. 2017.

[32] Y. Lv, W. Liu, and Z. Wang, "Heterogeneous cross-domain identity authentication scheme based on proxy resignature in cloud environment," *Math. Problems Eng.*, vol. 2020, no. 6, pp. 1–12, 2020.

[33] L. Xiaoxue and M. Wenping, "CDAKA: A provably-secure heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS," *J. Med. Syst.*, vol. 42, no. 8, pp. 1–15, 2018.

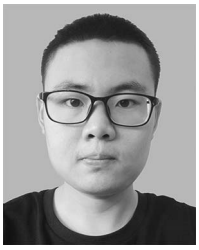
[34] J. Liu, Y. Liu, Y. Lai, R. Li, S. Wu, and S. Mian, "Cross-heterogeneous domain authentication scheme based on blockchain," *J. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 92–100, 2021.

[35] S. Xuan, H. Xiao, D. Man, W. Wang, and W. Yang, "A cross-domain authentication optimization scheme between heterogeneous IoT applications," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–14, Sep. 2021.

[36] G. Li, Y. Wang, B. Zhang, and S. Lu, "Smart contract-based cross-domain authentication and key agreement system for heterogeneous wireless networks," *Mobile Inf. Syst.*, vol. 2020, pp. 1–16, Sep. 2020.
[37] H. Zhang, X. Chen, X. Lan, H. Jin, and Q. Cao, "BTCAS: A blockchain-based thoroughly cross-domain authentication scheme," *J. Inf. Security Appl.*, vol. 55, pp. 1–10, Dec. 2020.
[38] K. Zhao and L. Ge, "A survey on the Internet of Things security," in *Proc. Int. Conf. Comput. Intell. Security*, 2013, pp. 663–667.
[39] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
[40] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-Health systems via consortium blockchain," *J. Med. Syst.*, vol. 42, no. 8, pp. 1–18, 2018.
[41] D. Boneh et al., "Threshold cryptosystems from threshold fully homomorphic encryption," in *Proc. Adv. Cryptol.*, 2018, pp. 565–596.
[42] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. Annu. Int. Cryptol. Conf.*, 1991, pp. 129–140.
[43] S. Agrawal, C. Ganesh, and P. Mohassel, "Non-interactive zero-knowledge proofs for composite statements," in *Proc. Adv. Cryptol.*, 2018, pp. 643–673.
[44] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. Conf. Theory Appl. Cryptogr. Techn.*, 1986, pp. 186–194.
[45] J. Groth and M. Kohlweiss, "One-out-of-many proofs: Or how to leak a secret and spend a coin," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2015, pp. 253–280.
[46] E. W. Weisstein, "Kronecker delta." 2002. [Online]. Available: https://mathworld.wolfram.com/
[47] "HyperLedger fabric." Accessed: Jun. 20, 2022. [Online]. Available: https://www.hyperledger.org/
[48] "VMware workstation 16 pro." Accessed: Jun. 20, 2022. [Online]. Available: https://www.vmware.com/products/workstation-pro.html
[49] J. Benet, "IPFS-content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

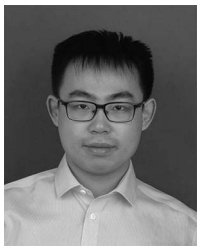
Fei Tong (Member, IEEE) received the M.S. degree in computer engineering from Chonbuk National University, Jeonju, South Korea, in 2011, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 2016.

He is currently an Associate Professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China, also with the Jiangsu Province Engineering Research Center of Security for Ubiquitous Network, Nanjing, and also with Purple Mountain Laboratories, Nanjing, Jiangsu. From 2016 to 2018, he was a Postdoctoral Research Fellow with the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include Internet of Things and ubiquitous networking intelligence and security.



Xing Chen received the B.S. degree in computer science and technology from Jilin University, Changchun, China, in 2020. He is currently pursuing the M.S. degree with the School of Cyber Science and Engineering, Southeast University, Nanjing, China.

His research interests are in the area of blockchain technology, authentication security, and applied cryptography.



Cheng Huang (Member, IEEE) received the B.Eng. and M.Eng. degrees in information security from Xidian University, Xi'an, China, in 2013 and 2016 respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2020.

He is currently a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests are in the areas of security and privacy in connected vehicles, databases, and blockchain.

Yujian Zhang received the B.S., M.S., and Ph.D. degrees from Southeast University, Nanjing, China, in 2006, 2010, and 2018, respectively.

He is currently an Assistant Professor with the School of Cyber Science and Engineering, Southeast University. His research interests include blockchain and system security.

Dr. Zhang received several awards and honors when pursuing degrees at Southeast University, including the Outstanding Undergraduate Award, the Outstanding Graduate Award, and the Best Doctoral Dissertation Award.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks.

Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals, Ontario, in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the President of the IEEE Communications Society. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and the Member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.