Efficient and Privacy-preserving Blockchain-based Multi-factor Device Authentication Protocol for Cross-domain IIoT

Yan Zhang, Bing Li, Jiaxin Wu, Bo Liu, *Senior Member, IEEE*, Rui Chen, and Jinke Chang

***Abstract*—Industrial Internet of Things (IIoT) has emerged as a prospective technology that improves the productivity and automation level for industrial applications. Devices from cooperative IIoT domains will communicate and collaborate on the increasingly complicated manufacturing tasks. To secure cross-domain device collaborations, we propose combining the blockchain with multi-factor authentication. Because the multi-factor authentication conforms to IIoT devices' operation modes and brings higher security levels, and the blockchain technology contributes to building trust among different domains. However, this combined usage still has limitations in terms of the potential loss of factor attack, the storage overhead on the blockchain, and the contradiction between efficiency and privacy preservation. Motivated by these facts, in this paper, we develop a privacy-preserving blockchain-based multi-factor device authentication protocol for cross-domain IIoT. Specifically, multiple factors are additionally encoded by the hardware fingerprint into random numbers, before being transformed into key materials. The blockchain only stores each domain's dynamic accumulator, which accumulates derived key materials for devices, thereby reducing the overhead. Moreover, the on-chain accumulator is leveraged to efficiently verify the unlinkable identities of cross-domain IIoT devices. The security of our protocol is formally proved, and the security features and functionalities are respectively discussed. A proof-of-concept prototype was implemented to prove the efficiency and reliability. The comparison results indicate that the on-chain storage is greatly reduced. Finally, the smart contract's performance was evaluated to show scalability.**

***Index Terms*—Industrial Internet of Things, blockchain, cross-domain IIoT, multi-factor authentication, security and privacy.**

I. Introduction

INDUSTRIAL Internet of Things (IIoT) has recently emerged as a promising machine-oriented platform for industrial areas. Industrial assets (e.g., IIoT devices, resources, and systems) are interconnected through machine-to-machine links to assist the advent of digital and smart manufacturing [1]. The operational processes are combined with information technology to activate remote real-time access, flexible data collection and sharing, and on-demand command, etc [2]. As a result, various innovative IIoT applications have been spawned, including healthcare service, production management, food supply chain, and public security [3], etc. To efficiently finish the increasingly complicated production tasks, the cloud-based network enables IIoT devices from different cooperative domains to flexibly exchange information and collaborate. Hence, the cross-domain collaboration in IIoT could significantly improve the productivity, thereby becoming a prospective industrial production mode [4].

However, implementing the cross-domain collaboration in IIoT is a non-trivial task, since there exist non-negligible security, privacy, and trust problems. IIoT devices deployed in open areas may be threatened by physical, cloning attacks as well as the malicious impersonation. And, the public channel that transmits the sensitive information is prone to active or passive attack, such as replay attack or eavesdropping. Furthermore, entities in each domain would only trust their domain administrator, and it becomes an open problem to build trust among different domains. Therefore, the significant role of cross-domain device authentication in securing the cross-domain collaborations is evident, as it can effectively build trust among entities in different domains, make them authenticate each other, and establish a trustworthy session key to protect the public channel.

Multi-factor authentication protocols have recently been proposed by simultaneously combining multiple factors (e.g., PIN, password, hardware, biometrics) from uses and devices, to provide better security protection for IIoT systems [5]. This security mechanism conforms to IIoT applications well, as finishing production tasks not only needs devices to operate independently, but also requires the human-device interaction. The multi-factor authentication is usually deployed in a closed domain, where a trusted domain administrator is relied on to issue and distribute certificates (or keys) for devices and servers [6], [7]. However, it is difficult for the administrator of one single domain to build trust among different domains as well as be fully trusted by all entities in other domains. Therefore, the multi-factor authentication cannot be directly applied to cross-domain IIoT scenarios.

The prospective blockchain technology is essentially a distributed and decentralized public digital ledger [8]. Data will be trustfully synchronized among peer nodes from serval different domains in the form of transactions, needless of relying on a trusted central authority. This feature has been leveraged by the existing security mechanisms [9]–[16] to build trust among different domains. Therefore, it is reasonable to combine the blockchain with multi-factor authentication.

Bing Li (Corresponding Author) is with the School of Microelectronics, School of Cyber Science and Engineering, Shenzhen Research Institute, and Advanced Cloud-System Research Center, Southeast University, Nanjing, 210000, China. Author email: bernie_seu@seu.edu.cn. This paper was supported by National Natural Science Foundation of China 62174150 and Shenzhen Science Technology and Innovation Commission 20170817115500476.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3176192, IEEE Internet of Things Journal

However, when applying the combination of blockchain and multi-factor authentication to secure device collaborations and communications in cross-domain IIoT, the following issues still need to be addressed:

- The potential loss of factor attack: To prevent the known attacks [17] to the multi-factor database at server-side, the state-of-the-art studies [5], [6] proposed mapping factors to random numbers, and stored them in the form of public keys. However, these multi-factor protection methods have not considered the potential loss of factor attack existing in IIoT scenarios. Devices usually need to undertake automatic production tasks without the human-machine interaction. As a result, IIoT devices should independently provide sensitive factors (e.g PIN, serial number, MAC address) for authentication. The requirement of these device factors would lead to the loss of factor attack, as the attacker could derive these factors from the captured IIoT devices by power analysis [18]. These compromised factors would be further used to perform the malicious impersonation. Therefore, the resistance to the loss of factor attack should be included in the protection method for multiple factors.
- The contradiction between efficiency and privacy preservation: Usually, there are two ways of leveraging the blockchain to develop cross-domain authentication protocols. The first category registers public keys in the blockchain ledger in advance [10], [12]. The server could retrieve these keys by querying the blockchain ledger as needed. Another category transmits one-time pseudo public keys in the form of new transactions during the authentication [4], [11], [14]. The advantage of the first category is its high efficiency, since the server only needs to query the local copy of the ledger, needless of proposing new transactions. However, this category only encodes real identities into public keys to ensure anonymity. It is still possible for attackers to link different messages to the queried public key as well as the requestor device. The second category is privacy-preserving to support unlinkability by sending a one-time public key for each request, but the efficiency is constrained by the throughput of transactions. Therefore, it needs to explore how to simultaneously meet the requirements of high efficiency and privacy preservation.
- Storage overhead on the blockchain: To build trust among domains, each domain server should store a copy of the blockchain ledger and maintain a key-value state database (e.g., LevelDB) locally for the recorded transactions [19]. Hence, with the number of the on-chain public keys built from multiple factors and domains growing, the domain server is required to have decent disk and RAM capacities to accept the increasing overhead. The research [4] utilized the off-chain storage to reduce the data written on the blockchain. And, the blockchain data structure was optimized by adopting the RSA accumulator in work [20] and the Merkle Patricia Tree in work [21]. These studies all reduce the overhead by changing the conventional storage structure of the blockchain. Is it possible to address the on-chain storage issue from the aspect of the application (authentication) protocol layer?

To tackle the aforementioned problems, an efficient and privacy-preserving multi-factor authentication protocol using blockchain for cross-domain IIoT is proposed in this paper. To the best of our knowledge, it is the first work that combines blockchain and multi-factor authentication in a cross-domain IIoT environment. Our protocol can resist the potential loss of factor attack and greatly reduce the on-chain storage overhead. Besides, our work is also a novel attempt to both achieve high efficiency and privacy preserving in blockchain-based cross-domain authentication. Our contributions are discussed as follows:

(1) We design a hardware-assisted multi-factor key derivation method via physically unclonable functions (PUFs), which both prevents the system from the potential loss of factor attack and guarantees the security of multiple factors at server-side.

(2) We develop a novel cross-domain trust-building method by leveraging the on-chain dynamic accumulator to accumulate derived keys for IIoT devices with multiple factors. The blockchain only stores each domain's accumulator, needless of directly recording a large number of device keys, thereby greatly reducing the on-chain storage overhead.

(3) We propose integrating the on-chain accumulator into cross-domain device authentication to efficiently verify the unlinkable identities of devices from different IIoT domains. As a result, the requirements of high efficiency and privacy preservation could be satisfied simultaneously.

(4) A formal security proof is given based on BAN Logic, and the discussion of security features and functionalities is included in security analysis. A proof-of-concept prototype of our protocol was implemented to evaluate the performance.

The remaining part of this paper proceeds as follows. The related work is reviewed in Section II, and the preliminaries and system model are discussed in Sections III and IV. Section V describes our protocol in detail. The security analysis is given in Section VI, and the performance is evaluated in Section VII. Finally, we conclude our work in Section VIII.

II. Related Work

In this section, we review the multi-factor authentication protocols and blockchain-based cross-domain security mechanisms. And, the unresolved common problems and specific contributions of relevant studies are summarized in Table I.

A. Multi-factor Authentication with Factors Protection

In this part, we review multi-factor authentication protocols with factors protection, which authenticates multiple factors (e.g., password, biometrics, hardware) simultaneously at server-side to bring higher security levels. To protect the multi-factor database in the server, Li *et al.* [6] developed a multi-factor harden service via the oblivious pseudo random function to prevent the attack to the stored factors. And, the low-interactivity authentication was also designed based on smooth protective hash function to mitigate the latency. Similarly, in work [5], Zhang *et al.* proposed a parallel model to take in multiple factors for authentication. These factors

TABLE I
SUMMARY OF UNRESOLVED COMMON PROBLEMS AND SPECIFIC CONTRIBUTIONS OF EXISTING PROTOCOLS

Category of Protocols	Unresolved common problems	Protocols	Specific contributions
Multi-factor authentication with factors protection	1. Potential loss of factor attack 2. Trust among different domains	[5]	Their work improves the single-factor authentication and addresses the arisen issues of security and efficiency.
		[6]	It contributes to providing a low-interactivity secure multi-factor protocol in the standard model.
		[22]	Privacy-preserving remote multi-factor authentication is constructed by leveraging Pedersen commitment.
Blockchain-based cross-domain authentication	3. Contradiction between efficiency and privacy preservation 4. On-chain storage overhead from protocol layer	[11]	A dynamic key management framework is developed to transmit secret keys in the form of blockchain transactions.
		[12]	Smart contract is used in their work to manage public keys and system parameters to efficiently support cross-domain authentication.
		[14]	The blockchain is combined with a key derivation method to realize an effective certificate management.
		[4]	The identity-based cryptography is combined with the blockchain to ensure cross-domain device communications.
		[21]	Their work utilizes the Merkle Patricia Tree to improve blockchain's data structure and reduce the storage overhead.
Cross-domain authentication of combined usage	Address all above-mentioned issues	Our work	We proposed combining blockchain with multi-factor authentication to secure cross-domain device communications and collaborations.

will be transformed into public elements to realize the secure storage. Liu *et al.* [23] designed a secure remote multi-factor scheme, which leverages the chaotic map to provide shorter key size and reduce the communication overhead. The work [22] developed a privacy-preserving ZKPoK protocol for multiple factors. The Pedersen commitment was constructed to authenticate user passwords and biometrics at the same time.

In general, there exist two common problems impeding the integration of multi-factor authentication with cross-domain IIoT. The first problem is how to build trust among different domains. Second, although the security of multiple factors at server-side is ensure in these studies, the device-side would still be threatened by the loss of factor attack.

B. Blockchain-based Cross-domain Security Mechanisms

State-of-the-art review papers [24]–[27] have focused on the combination of blockchain and IoT. The concept of blockchain of Tings (BCoT) was proposed by Dai *et al.* in work [24]. They introduced the convergence of these two technologies and the related industrial applications. Moreover, several major challenges and potential solutions were all summarized. Ferrag *et al.* did a remarkable survey [25], which provides a classification of threat models for blockchain-based IoT (BIoT), and reviews the researches for BIoT security and privacy to show future challenges. Besides, they concluded the evaluation techniques for consensus algorithms and security analysis in survey [27]. This work also gave a guidance on evaluating the performance of blockchain-based security and privacy solutions for IoT. In addition, Kai *et al.* [26] performed a review of security challenges and potential research directions with respect to the deployment of smart contracts in IoT. Inspired by these surveys, the blockchain have been leveraged to build trust among different domains. A large number of blockchain-based security mechanisms have recently been proposed to improve security for cross-domain systems.

Ali *et al.* [9] proposed a blockchain-based framework called xDBAuth for cross-domain IIoT users and devices. In their work, a hierarchy of local and global smart contract was developed to perform the permission delegation and access control, which resists illegal delegations and preserves the user privacy at the same time. This master-slave hierarchical structure has also been used to construct cross-domain trust access mechanisms [28] for power terminals, and improve the efficiency and data credibility for cross-domain IoT authentication [15]. In addition, Wang *et al.* [29] leveraged undirected graph to build authentication relationships for IIoT devices. The digital signature and accumulator were combined to achieve the signature transitivity among different domains.

For access control, IoT Passport [30] was constructed as a trust framework to support collaborations among cross-platform devices. The blockchain will store the signatures of collaborative device operations. The signature will be used to realize the authorization and build the incentive mechanism. The work [31] proposed a blockchain-based hierarchical access control for user privacy-oriented scenarios. The multi-blockchain architecture was utilized to meet the requirements of low-latency and high-scalability.

The identity-based cryptography was also combined with the blockchain for cross-domain IoT in work [4], [32], [33]. Jia *et al.* [32] used the blockchain to replace the traditional trusted certificate authority and supported the identity-based self-authentication. To achieve the unlinkable cross-domain device authentication [4], the blockchain network is responsible for transmitting the one-time identity-based public keys to different IIoT domains. Moreover, Chen *et al.* [33] developed a decentralized identity management to avoid single point failure. And, the consensus algorithms was used to transmit the identity information to different domains without disclosing the user privacy.

In a typical cross-domain area of transportation, a conditional privacy-preserving protocol was developed by Lin *et al.* [14]. They proposed using a key derivation method that periodically updates public key pairs to reduce the key storage overhead. Moreover, a modified signature scheme was designed to support batch verification of transactions to improve efficiency. Similarly, in research [11], the transactions will be signed by the sender, and the information encrypted by the public key of the destination server will be transmitted through the blockchain network. Moreover, an optimized transaction collection algorithm was introduced for better performance. Kang *et al.* [34] proposed combining a reputation-based data sharing scheme with the blockchain to realize the high-quality authorized data sharing and secure data storage in vehicular edge computing. The reputation of vehicles will be managed by a three-weight subject logic model.

For heterogeneous wireless networks, Li *et al.* [12] proposed a smart contract-based cross-domain authentication by recording the public keys of the access point in the blockchain. The on-chain public keys efficiently verified users' certificates. Cheng *et al.* [16] designed a mutual authentication scheme for collaborative edge computing. The decentralization, the anonymity, the mobility of devices are ensured in their work. The work [13] solved the problem of "incomplete cross-domain" based on the blockchain. The participants from different domains enables to adopt completely different settings.

All these blockchain-based cross-domain mechanisms have not considered to reduce the on-chain storage overhead from the aspect of the protocol layer. Besides, the efficiency and unlinkability also have not been simultaneously achieved in these studies.

Overall, it is promising to leverage the combination of blockchain and multi-factor authentication in our work to secure cross-domain device communications and collaborations. However, the above-mentioned unresolved problems existing in this combined usage still remain to be addressed.

III. PRELIMINARIES

A. A Hardware-assisted Multi-factor Key derivation Method

PUFs could be regarded as hardware fingerprints and physical roots, which are derived from the manufacturing variations of integrated circuits. PUFs usually act as a secure one-way function $\{0,1\}^n \rightarrow \{0,1\}^m$. A n-bit challenge is taken in to output an unpredictable and unclonable m-bit response. We propose combing PUFs with the key derivation method BIP32 to support the device authentication and key protection.

Specifically, the physically secure PUF response fisrt combines with different random numbers, to generate unique PUF seeds for different working modes in IIoT applications. The PUF seed will encode multiple factors into random numbers. Then, random numbers are transformed into private/public key pairs using BIP32 based on elliptic curve cryptography. The server-side only stores public keys, thus, ensuring the security of multi-factor database. Even if factors are revealed, the private key could not be recovered correctly without the PUF seed. This is also the core that our protocol could resist the loss of factor attack. Moreover, the security threat of key leakage [14] existing in BIP32 is solved in our method by inserting a variable and recoverable secret c_i into the generation process of child private key $sk_{i,j}$. Our key derivation shown in Fig. 1 is detailed as follows:

1) Parent secret key derivation: The parent secret key derivation algorithm is *(k_i, c_i)=DerivePsk($R, r_i, factors$)*, where R, r_i, i, $factors$ denote the PUF response, the random number, the working mode, and factor list. The PUF seed $K_{(root,i)}$ is first calculated by $H_1(R||r_i)$. Then, the input factors $f_{t\in(0,1,2...n)}$ and the working mode i are encoded by the PUF seed into random numbers f_t^* , which are further transformed into parent secret keys (k_i, c_i).

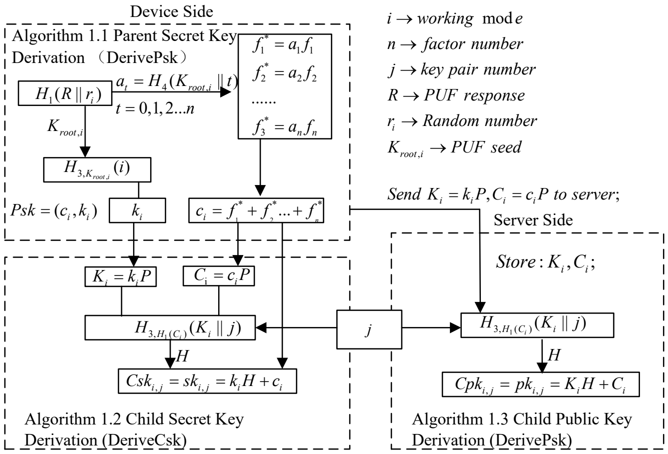

Device Side
Algorithm 1.1 Parent Secret Key Derivation （DerivePsk）
$H_1(R\|r_i)$
$a_t = H_4(K_{root,i}\|t)$
$t=0,1,2...n$
$f_1^* = a_1 f_1$
$f_2^* = a_2 f_2$
......
$f_3^* = a_n f_n$
$K_{root,i}$
$H_{3,K_{root,i}}(i)$
$Psk=(c_i,k_i)$
k_i
$c_i = f_1^* + f_2^* ... + f_n^*$
$i \rightarrow$ working mode
$n \rightarrow$ factor number
$j \rightarrow$ key pair number
$R \rightarrow$ PUF response
$r_i \rightarrow$ Random number
$K_{root,i} \rightarrow$ PUF seed
Send $K_i = k_i P, C_i = c_i P$ to server;
Server Side
Store: K_i, C_i;
$K_i = k_i P$
$C_i = c_i P$
$H_{3,H_1(C_i)}(K_i\|j)$
j
$H_{3,H_1(C_i)}(K_i\|j)$
H
$Csk_{i,j} = sk_{i,j} = k_i H + c_i$
H
$Cpk_{i,j} = pk_{i,j} = K_i H + C_i$
Algorithm 1.2 Child Secret Key Derivation (DeriveCsk)
Algorithm 1.3 Child Public Key Derivation (DerivePsk)

Fig. 1. A hardware-assisted multi-factor key derivation method.

$Remark$: Each IIoT device could choose different working modes to perform the automatic production tasks or human-machine interaction.

2) Child private key derivation: The child private key derivation algorithm is defined as *$Csk_{i,j}$=DeriveCsk(k_i, c_i, j)*. The elliptic curve cryptography transforms elements c_i and k_i to C_i and K_i by scalar multiplying the curve generator P. Then, the child private key $sk_{i,j}$ is computed as $k_i H_{3,H_1(C_i)}(K_i||j) + c_i$, where j is the number of the child private/public key pair.

3) Child public key derivation: The child public key derivation algorithm is denoted as *$Cpk_{i,j}$=DeriveCpk(K_i, C_i, j)*. The child public key is calculated as $Cpk_{i,j} = pk_{i,j} = K_i H + C_i$, where $H = H_{3,H_1(C_i)}(K_i||j)$.

B. Dynamic Accumulators

The accumulator is featured by accumulating a set S of values $(x_1, x_2, x_3..., x_n)$ into an element Δ. This is also the core of our protocol that reduces the storage overhead on the blockchain. The initialization of the accumulator refers to the method in [35]. A Type 3 bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, r, e)$ is first selected for the accumulator. $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are the cyclic groups of prime order r, and e denotes the bilinear mapping function: $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. Then, $y \in \mathbb{Z}_r^*$ is selected to compute $\hat{Y} = g_2^y, Y = g_1^y$, where g_1, g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. And, the element Δ is computed as $\tilde{g}^{\prod_1^n (y+x_i)}$ with $\tilde{g} \in \mathbb{G}_1$ and the value set S.

For each value x_i, there exists a witness $W_i = \Delta^{1/(y+x_i)}$. If the value x_i is included in the accumulator, the equation $e(\Delta, g_2) = e(W_i, \hat{Y} g_2^{x_i})$ will hold. Our protocol leverages this novel feature to verify the unlinkable identities of cross-domain IIoT devices. Furthermore, the dynamic accumulator supports the revoke or update of accumulated values.

This paper also constructs a zero-knowledge proof of knowledge method to verify whether x_i is included in the accumulator Δ. First, the random number $rw \in \mathbb{Z}_r$ is chosen by the prover and combined with the value x_i of the prover to generate three elements $X = (\hat{Y})^{rw}$, $K = H_4(rw||x_i)x_i$, $R = g_{2a}^{H_4(rw||x_i)}$. Then, the verifier only needs to verify the equation $e(W_i, Xg_{2a}^{K}) ==?e(\Delta, R)$ with W_i, X, K, needless of acquiring the actual value of x_i.

C. Blockchain and Smart Contract

The blockchain emerges as a distributed and decentralized ledger, the data of which is tamper-proofing, highly available, and synchronized among the peer nodes in the form of transactions [36]. The peer node will keep a copy of the digital ledger and build the recorded transactions into a state key-value database by invoking the smart contract. In this paper, the key is the blockchain address of each domain, and the value is the domain information. The domain information $(H_1(DID), PP_a, version, \Delta, PK_{sa})$ includes the hash value of the domain identity $H_1(DID)$, the public parameters PP_a, the accumulator Δ, the version, and the public key of the domain server PK_{sa}. By this way, the blockchain could effectively build trust among different domains.

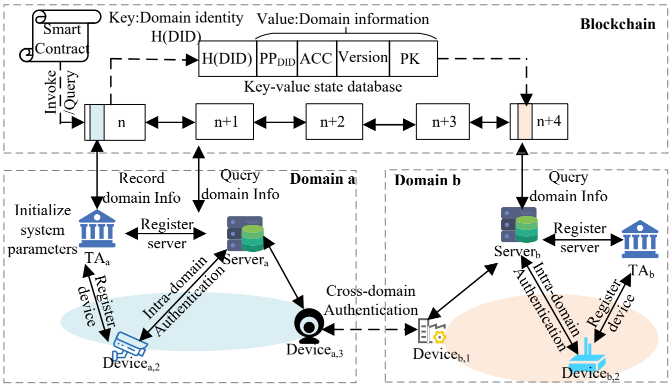


Fig. 2. System model.

IV. System Model

A. System Model

The system model is composed of the blockchain and multiple domains. As is presented in Fig. 2, each domain contains the trusted authority (TA), the server, and IIoT devices.

(1) Trusted authority (TA): The TA is trusted by all the entities in each domain, and registers IIoT devices and the domain server. Moreover, the TA also manages the on-chain domain information by invoking the smart contract.

(2) IIoT device: IIoT devices are deployed in one designated IIoT domain to finish the production tasks or interact with the users. In this paper, IIoT devices are not lightweight and will provide multiple factors as well as afford the elliptic curve cryptography. They could communicate with the intra-domain server, or connect with the entities in other domains to make cross-domain collaborations.

(3) Server: The server of each domain has decent capabilities to provide various services, such as data collection, real-time access, and data analysis. The server could query the blockchain ledger to acquire other domains' information.

(4) Blockchain: The blockchain acts as a public ledger to manage the domain information. The TA and the server of each domain all join the blockchain to register, update, query, or revoke the domain information by invoking or querying the smart contract. Hence, there are two basic requirements for the blockchain: (1) the platform should be robust and secure; (2) the smart contract function should be supported.

Based on these requirements, it is really convenient and efficient to transplant current blockchain platforms into our system. We only need to deploy the TA and server of each domain as peer nodes of the chosen platform (e.g., Hyperledger Fabric, Ethereum 2.0). And, the smart contract should be properly installed in all these nodes to maintain the public ledger of the domain information.

B. Threat Model

The capabilities of the adversary in our system are defined as follows:

(1) The public channel is assumed to be fully controlled by the adversary, as presented in the Dolev-Yao (DY) model [37]. The adversary could intercept, replay, eavesdrop, and modify the messages transmitted through the public channel.

(2) IIoT devices would be at the risk of physical and cloning attacks, e.g., power analysis. Hence, the adversary may access the secrets and keys stored in IIoT devices.

(3) The server is modeled as a semi-honest entity, which performs the protocol honestly but tries to acquire sensitive information, such as factors. However, the long-term secret keys of the server will not be accessed by the adversary as assumed in [38]. The TA of each domain is always secure and trustworthy.

(4) The blockchain is able to reliably manage the public ledger. The recorded transactions are tamper-proofing and always available. The adversary can query the ledger, but cannot propose transactions or corrupt the blockchain system.

The adversary with aforementioned abilities is most likely to perform the following known attacks:

(1) Impersonation attack: The adversary would try to impersonate the server, or forge IIoT devices with derived factors. Also, they may become the man-in-the-middle attacker and modify transmitted messages to pass the verification.

(2) Replay attack: The adversary could intercept transmitted messages and replay them later to perform this attack.

(3) Physical attack and loss of factors attack: The adversary would capture IIoT devices to perform physical attacks and derive secret keys or devices factors.

(4) Desynchronization attack: The adversary may intercept the communication channel to desynchronize the update of key materials or identity information.

C. Security Assumptions

The security assumptions of our proposed protocol are discussed as follows.

The PUF circuit and the blockchain are all assumed to be secure in our protocol. Furthermore, the underlying security assumptions and difficult problems are defined as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3176192, IEEE Internet of Things Journal

Definition 1 (Elliptic Curve Discrete Logarithm Problem): Given any P, $sP \in \mathbb{G}$, the probabilistic polynomial time (PPT) adversary cannot get s with a non-negligible probability.

Definition 2 (Elliptic Curve Computational Diffie-Hellman Problem): Given any aP, $bP \in \mathbb{G}$ to compute abP. The PPT adversary cannot solve the ECDH problem with a non-negligible probability.

Definition 3 (q-Strong Diffie-Hellman Assumption): Given the cyclic group $\mathbb{G}$ with the prime order p, generator g of group $\mathbb{G}$, $a \in \mathbb{Z}_p^*$, and $q > 0$, for any PPT algorithm $\mathcal{A}$, the following function is negligible.

$$\Pr[(x, g^{1/(a+x)}) \leftarrow \mathcal{A}(g, g^a, g^{a^2}, ..., g^{a^q}) \wedge x \in \mathbb{Z}_p^*] \leq \varepsilon$$

V. Our Proposed Protocol

This section describes the detailed discussion of our proposed protocol. This protocol consists of four main phases as shown in Fig. 3, including **Registration** (R.1∼R.5), **Intra-domain Authentication** (A.1∼A.3), **Cross-domain Authentication** (C.1∼C.8) authentication, and **Key Negotiation**.

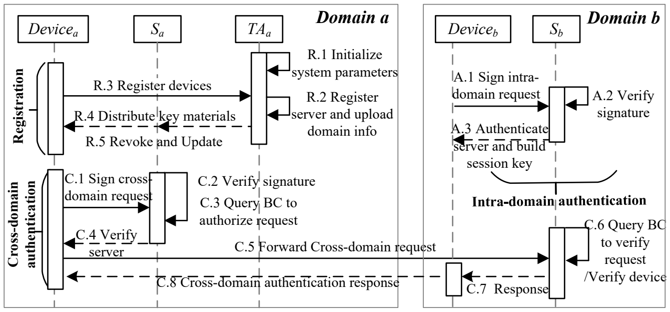
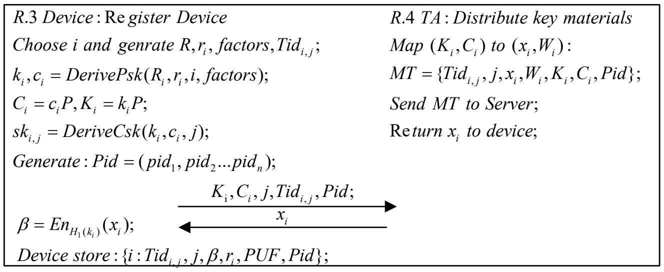

Domain a
Device$_a$
S_a
TA_a
R.1 Initialize system parameters
R.2 Register server and upload domain info
R.3 Register devices
R.4 Distribute key materials
R.5 Revoke and Update
Registration
Cross-domain authentication
C.1 Sign cross-domain request
C.2 Verify signature
C.3 Query BC to authorize request
C.4 Verify server
C.5 Forward Cross-domain request
C.8 Cross-domain authentication response
Domain b
Device$_b$
S_b
A.1 Sign intra-domain request
A.2 Verify signature
A.3 Authenticate server and build session key
Intra-domain authentication
C.6 Query BC to verify request /Verify device
C.7 Response

Fig. 3. Flow chart of our protocol.



```
$R.3\ Device: \mathrm{Re}gister\ Device$
$Choose\ i\ and\ genrate\ R, r_i, factors, Tid_{i,j};$
$k_i, c_i = DerivePsk(R_i, r_i, i, factors);$
$C_i = c_i P, K_i = k_i P;$
$sk_{i,j} = DeriveCsk(k_i, c_i, j);$
$Generate: Pid = (pid_1, pid_2 ... pid_n);$
                    $K_i, C_i, j, Tid_{i,j}, Pid;$ →
                    ← $x_i$
$\beta = En_{H_1(k_i)}(x_i);$
$Device\ store: \{i: Tid_{i,j}, j, \beta, r_i, PUF, Pid\};$

$R.4\ TA: Distribute\ key\ materials$
$Map\ (K_i, C_i)\ to\ (x_i, W_i):$
$MT = \{Tid_{i,j}, j, x_i, W_i, K_i, C_i, Pid\};$
$Send\ MT\ to\ Server;$
$\mathrm{Re}turn\ x_i\ to\ device;$
```

Fig. 4. Registration.

A. Registration Phase

In this phase, the parameters of each domain are first initialized. Then, the trusted authority registers the server and devices. At last, we discuss the revoke and update processes.

1) Initialize system parameters (R.1): The TA of each domain first initializes the dynamic accumulator to acquire a tuple $(\mathbb{G}_{a1}, \mathbb{G}_{a2}, \mathbb{G}_{aT}, r, e, y, Y, \hat{Y})$ as mentioned in section III-B . Then, another elliptic curve cyclic group $\mathbb{G}_a$ of order r is chosen with the generator P_a. Besides, four hash functions are also defined as: $H_1 : \{0,1\}^* \to \{0,1\}^{l_1}$, $H_2 : \{0,1\}^* \to \{0,1\}^{l_2}$, $H_3 : \{0,1\}^* \times \{0,1\}^k \to \mathbb{Z}_r^*$, $H_4 : \{0,1\}^* \to \mathbb{Z}_r^*$, where l_1 and l_2 are the length of the hash functions, and k denotes the length of the hash key. Last, the public parameters $PP_a = (\mathbb{G}_{a1}, \mathbb{G}_{a2}, \mathbb{G}_{aT}, \mathbb{G}_a, P_a, g_{a1}, g_{a2}, r, e, y, Y, \hat{Y}, H_{i \in \{1,2,3,4\}})$ and the domain identity DID_a are made public.

2) Register the server and upload domain information (R.2): The TA would distribute the public/private key pair (sk_{sa}, PK_{sa}) for the server. Afterward, the TA invokes the smart contract to record the domain information $(H_1(DID_a), PP_a, version, \Delta_a, PK_{sa})$ into the blockchain ledger. The entities of each domain will update the key-value state database for newly recorded transactions.

3) Register devices (R.3): The devices will choose a working mode i and an identity $Tid_{i,j}$ for this mode. The PUF response R is derived to generate the PUF seed $K_{(root,i)}$ with the chosen random number r_i. Then, multiple factors will be input in this mode and transformed into the parent secret keys *(k_i, c_i)=DerivePsk($R, r_i, factors$)*. Even if factors are revealed, the attacker still cannot derive the correct parent secret key (k_i, c_i) without the PUF response. Next, the elements K_i, C_i will be further generated as well as a set of pseudo identities $Pid = (pid_1, pid_2, ..., pid_n)$. The device sends the $(K_i, C_i, j, Tid_{i,j}, Pid)$ to the TA.

4) Distribute key materials (R.4): Once receiving the registration request from IIoT device, the TA maps the elements K_i, C_i to a selected value/witness pair x_i/W_i, which is already included in the accumulator. Then, the mapping table $MT = \{Tid_{i,j} : j, x_i, W_i, K_i, C_i, Pid\}$ will be built. Afterward, the mapping table is distributed to the server, and the value x_i is returned to the device.

The IIoT device specially protects the accumulator value x_i by encrypting it into the element $\beta = En_{H_1(k_i)}(x_i)$ and stores $\{i : Tid_{i,j}, j, \beta, r_i, PUF, Pid\}$ locally. The server stores the mapping table MT and also uses its long-term secret key to protect the element x_i as the device-side does.

5) Revoke and update device values (R.5): If the revocation is required, the TA will remove x_j from the accumulator Δ by computing $\Delta_{new} = \Delta^{(1/y+x_j)}$ and invoke the smart contract to upload the new version of the domain information into the blockchain. Moreover, the server will delete the revoked information from mapping table and update each (x_i, W_i) pair by calculating $W_i^{new} = (W_i/\Delta_{new})^{1/(\mathrm{x}_j - x_i)}$.

If the device value x_j needs to be updated, the TA and server will first revoke the old value x_j as mentioned above, and then allocate an unused new (x_i^{new}, W_i^{new}) pair for the child public key pair. Finally, the new value x_i^{new} and (x_i^{new}, W_i^{new}) will be respectively distributed to the device and server.

B. Intra-domain Authentication

In this phase, IIoT devices establish mutual intra-domain authentication with the intra-domain server.

1) Sign intra-domain request (A.1): The $device_a$ working under the module i first reads the $Tid_{i,j}, j, \beta, r_i$ from the storage. Then, the parent private keys (k_i, c_i) are generated to derive the jth child secret key $sk_{i,j}$ and obtain the accumulator value $x_i = De_{H_1(k_i)}(\beta)$. Next, random numbers $n_1, N_1 = n_1 P$ are selected to compute $D_1 = n_1 PK_{sa}$ and perform the schnnor signature [39] $s = sk_i H_4(x_i || N_1) + n_1$. Besides, the timestamp TS_1 and the hash value Z_1 are inserted to resist the replay attack and ensure the integrity of the

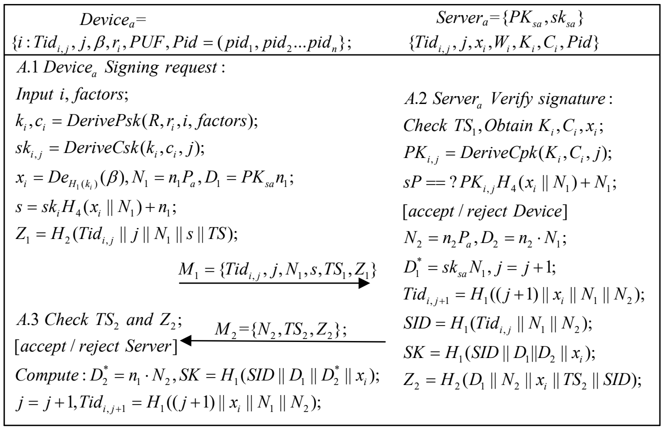

$Device_a$=
$\{i: Tid_{i,j}, j, \beta, r_i, PUF, Pid = (pid_1, pid_2 ... pid_n\};$
$Server_a = \{PK_{sa}, sk_{sa}\}$
$\{Tid_{i,j}, j, x_i, W_i, K_i, C_i, Pid\}$
$A.1\ Device_a\ Signing\ request:$
$Input\ i, factors;$
$k_i, c_i = DerivePsk(R, r_i, i, factors);$
$sk_{i,j} = DeriveCsk(k_i, c_i, j);$
$x_i = De_{H_1(k_i)}(\beta), N_1 = n_1 P_a, D_1 = PK_{sa} n_1;$
$s = sk_i H_4(x_i \| N_1) + n_1;$
$Z_1 = H_2(Tid_{i,j} \| j \| N_1 \| s \| TS);$
$M_1 = \{Tid_{i,j}, j, N_1, s, TS_1, Z_1\}$
$A.2\ Server_a\ Verify\ signature:$
$Check\ TS_1, Obtain\ K_i, C_i, x_i;$
$PK_{i,j} = DeriveCpk(K_i, C_i, j);$
$sP == ?\, PK_{i,j} H_4(x_i \| N_1) + N_1;$
$[accept / reject\ Device]$
$N_2 = n_2 P_a, D_2 = n_2 \cdot N_1;$
$D_1^* = sk_{sa} N_1, j = j+1;$
$Tid_{i,j+1} = H_1((j+1) \| x_i \| N_1 \| N_2);$
$SID = H_1(Tid_{i,j} \| N_1 \| N_2);$
$SK = H_1(SID \| D_1 \| D_2 \| x_i);$
$Z_2 = H_2(D_1 \| N_2 \| x_i \| TS_2 \| SID);$
$A.3\ Check\ TS_2\ and\ Z_2;$
$M_2 = \{N_2, TS_2, Z_2\};$
$[accept / reject\ Server]$
$Compute: D_2^* = n_1 \cdot N_2, SK = H_1(SID \| D_1 \| D_2^* \| x_i);$
$j = j+1, Tid_{i,j+1} = H_1((j+1) \| x_i \| N_1 \| N_2);$

Fig. 5. Intra-domain authentication scheme.

transmitted messages. We will omit the description of TS_i and Z_i in the following discussions.

The message $M_1 = \{Tid_{i,j}, j, N_1, s, TS_1, Z_1\}$ is sent to the $server_a$.

2) Verify signature (A.2): The $server_a$ retrieves the elements (K_i, C_i, j) from the storage according to the $Tid_{i,j}$, and further derives the child public key $PK_{i,j}$=*DeriveCpk*(K_i, C_i, j) to verify the signature s. If the signature is valid, the server accepts the device. Then, $Tid_{i,j+1} = H_1(j+1||x_i||N_1||N_2)$ are calculated with N_1, N_2 to update the one-time $Tid_{i,j}$ and the key pair number. If necessary, the session identity and the session key could be computed as $SID = H_1(Tid_{i,j}||N_1||N_2)$ and $H_1(SID||D_1||D_2||x_i)$, where the element $D_2 = n_2 N_1$.

The $server_a$ returns the message $M_2 = \{N_2, TS_2, Z_2\}$ to $device_a$, which contains the secret element $D_1 = N_1 sk_{sa}$.

3) Authenticate server and build session key (A.3): After the timestamp TS_2 and the hash value Z_2 are verified, the $device_a$ accepts the $server_a$. Then, $Tid_{i,j}$ and j are updated as aforementioned, and the session identity and session key are computed to secure the following communications.

C. Cross-domain Authentication

In this phase, IIoT device first authenticates with the intra-domain server to get the authorization of the cross-domain request. Then, devices encode the accumulator value into the request, and further send the request to the cross-domain server. The server queries the accumulator from the blockchain ledger and verifies whether the value of the requestor is included in the accumulator. If so, the unilateral cross-domain authentication succeeds.

1) Sign cross-domain request (C.1): This step is almost the same as step A.1. The only difference is that the identity ID_b of the target $device_b$ in domain b is encrypted into ID_b^*, and inserted into the message $M_3 = \{Tid_{i,j}, j, N_3, s, TS_3, ID_b^*, Z_3\}$.

2) Verify signature and authorize request (C.2∼C3): In step C.2, the $server_a$ verifies the signature as mentioned in step A.2, and obtains $ID_b = De_{H_1(D_3^*||x_i)}(ID_b^*)$.

In step C.3, the $server_a$ queries the blockchain ledger to acquire the information $\{PP_b, ACC_b, version, PK_{sb}\}$ of domain b. Nonce $N_{4,b}$ is selected to compute the secret element $D_{4,b} = n_4 PK_{sb}$, which is used to encrypt four elements ID_b and ID_a, the witness W_i, and the domain information $version$ into W_i^*. At last, the $server_a$ authorizes the cross-domain request, and sends the message $M_4 = \{N_{4,b}, TS_4, W_i^*, Z_4\}$ back to the $device_a$

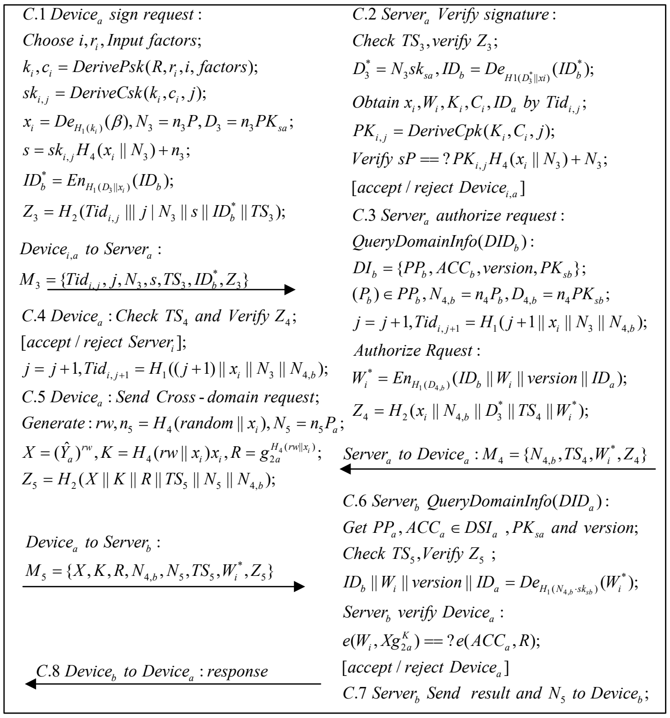

$C.1\ Device_a\ sign\ request:$
$Choose\ i, r_i, Input\ factors;$
$k_i, c_i = DerivePsk(R, r_i, i, factors);$
$sk_{i,j} = DeriveCsk(k_i, c_i, j);$
$x_i = De_{H_1(k_i)}(\beta), N_3 = n_3 P, D_3 = n_3 PK_{sa};$
$s = sk_{i,j} H_4(x_i \| N_3) + n_3;$
$ID_b^* = En_{H_1(D_3 \| x_i)}(ID_b);$
$Z_3 = H_2(Tid_{i,j} \||| j | N_3 \| s \| ID_b^* \| TS_3);$
$Device_{i,a}\ to\ Server_a:$
$M_3 = \{Tid_{i,j}, j, N_3, s, TS_3, ID_b^*, Z_3\}$
$C.2\ Server_a\ Verify\ signature:$
$Check\ TS_3, verify\ Z_3;$
$D_3^* = N_3 sk_{sa}, ID_b = De_{H1(D_3^* \| xi)}(ID_b^*);$
$Obtain\ x_i, W_i, K_i, C_i, ID_a\ by\ Tid_{i,j};$
$PK_{i,j} = DeriveCpk(K_i, C_i, j);$
$Verify\ sP == ?\, PK_{i,j} H_4(x_i \| N_3) + N_3;$
$[accept / reject\ Device_{i,a}]$
$C.3\ Server_a\ authorize\ request:$
$QueryDomainInfo(DID_b):$
$DI_b = \{PP_b, ACC_b, version, PK_{sb}\};$
$(P_b) \in PP_b, N_{4,b} = n_4 P_b, D_{4,b} = n_4 PK_{sb};$
$j = j+1, Tid_{i,j+1} = H_1(j+1 \| x_i \| N_3 \| N_{4,b});$
$Authorize\ Rquest:$
$W_i^* = En_{H_1(D_{4,b})}(ID_b \| W_i \| version \| ID_a);$
$Z_4 = H_2(x_i \| N_{4,b} \| D_3^* \| TS_4 \| W_i^*);$
$C.4\ Device_a: Check\ TS_4\ and\ Verify\ Z_4;$
$[accept / reject\ Server_i];$
$j = j+1, Tid_{i,j+1} = H_1((j+1) \| x_i \| N_3 \| N_{4,b});$
$C.5\ Device_a: Send\ Cross\text{-}domain\ request;$
$Generate: rw, n_5 = H_4(random \| x_i), N_5 = n_5 P_a;$
$X = (\hat{Y}_a)^{rw}, K = H_4(rw \| x_i) x_i, R = g_{2a}^{H_4(rw \| x_i)};$
$Z_5 = H_2(X \| K \| R \| TS_5 \| N_5 \| N_{4,b});$
$Server_a\ to\ Device_a: M_4 = \{N_{4,b}, TS_4, W_i^*, Z_4\}$
$Device_a\ to\ Server_b:$
$M_5 = \{X, K, R, N_{4,b}, N_5, TS_5, W_i^*, Z_5\}$
$C.6\ Server_b\ QueryDomainInfo(DID_a):$
$Get\ PP_a, ACC_a \in DSI_a\ , PK_{sa}\ and\ version;$
$Check\ TS_5, Verify\ Z_5\ ;$
$ID_b \| W_i \| version \| ID_a = De_{H_1(N_{4,b} \cdot sk_{sb})}(W_i^*);$
$Server_b\ verify\ Device_a:$
$e(W_i, Xg_{2a}^K) == ?\, e(ACC_a, R);$
$[accept / reject\ Device_a]$
$C.8\ Device_b\ to\ Device_a: response$
$C.7\ Server_b\ Send\ result\ and\ N_5\ to\ Device_b;$

Fig. 6. Cross-domain authentication scheme.

Remark: The identities ID_a and ID_b here are composed of the real identity of the IIoT device and the working mode i.

3) Verify server and forward cross-domain request (C.4∼C5): Upon receiving M_4, the $device_a$ verifies the timestamp and hash value to decide whether to accept the $server_a$ and updates the $Tid_{i,j}$ as mentioned in step A.3.

In step C.5, the ZKPoK method is constructed. The element $rw \in \mathbb{Z}_r^*$ is chosen and combined with the accumulator value x_i and the random number $random$ to generate $n_5 = H_4(random||x_i)$, $N_5 = n_5 P_a$, $x = (\hat{Y})^{rw}$, $K = H_4(rw||x_i)x_i$, $R = g_{2a}^{H_4(rw||x_i)}$.

Then, the $device_a$ continues to forwarding the cross-domain request $M_5 = \{X, K, R, N_{4,b}, N_5, TS_5, W_i^*, Z_5\}$ to $server_b$.

4) Verify the cross-domain request and send the response (C.6∼C.8): On receiving M_5, the $server_b$ first queries the blockchain ledger to obtain the information $(PP_a, version, ACC_a, PK_{sa})$ of domain a. Then, the $server_b$ decrypts W_i^* by calculating $De_{H_1(N_{4,b} \cdot sk_{sb})}(W_i^*)$ to obtain the witness W_i of x_i and the real identities ID_b, ID_a. Afterward, the $server_b$ checks whether $e(W_i, Xg_{2a}^K) ==?e(ACC_a, R)$. If this equation holds, it means that the value x_i of $device_a$ (under working mode i) is included in the accumulator. The $server_b$ authenticates the $device_a$ successfully.

In step C.7, the $server_b$ would locate the $device_b$ and sends the cross-domain authentication result as well as the element N_5 to $device_b$.

In step C.8, the $device_b$ will finally return the response to the $device_a$.

D. Mutual Cross-domain authentication and key negotiation

1) Mutual Cross-domain authentication: The intra-domain protocol is mutual authentication. And, the cross-domain protocol discussed in section V-C is unilateral authentication, where the $device_a$ is authenticated by $device_b$. When the mutual cross-domain authentication is needed, the $device_b$ only needs repeating the steps C.1∼C.8. After that, the authenticity of $device_b$ would be verified by the $device_a$.

2) Key negotiation: Both sides of the communicators will establish the session key and the session identity to protect the public channel.

For the cross-domain request, we insert a long-term secret x_i into the Ephemeral Elliptic Curve based Diffie-Hellman (ECDHE) based key exchange method. The communicators ($device_a$ and $device_b$) will retain the random number $N_{5a} = n_{5a}P$,$N_{5b} = n_{5b}P$ respectively. Then, the session key $SK = H_1(n_{5a}N_{5b}) = H_1(n_{5b}N_{5a})$ as well as the session identity $SID = H_1(N_{5a}||N_{5b})$ could be computed.

For the intra-domain authentication, the way of negotiating the session identity $SID = H_1(Tid_{i,j}||N_1||N_2||D_1)$and the session key $H_1(SID||D_1||D_2||x_i)$ has already been explain.

VI. Security analysis

In this section, we first leverage BAN Logic to perform the formal security proof for our intra-domain and cross-domain authentication protocols. Then, the security features and functionalities are discussed.

A. Formal Security Proof of BAN Logic

The goal of the authentication is to ensure that communicators trust each other and communications are only carried out among the authorized entities. Besides, the adversary cannot perform malicious impersonation or access sensitive information. Due to the increasing complexity of protocols, it becomes more difficult to formally prove the security of the authentication. However, the formal proof method BAN Logic proposed by Burrows, Abadi, and Needham (BAN) [40] has recently been used to express the belief and the reasoning for the security of authentication [41], [42]. The review article [27] also regarded BAN Logic as a formal security proof technique for blockchain-based protocols. Hence, this paper leverages BAN Logic to give the formal security proof.

To better understand the formal proof of the BAN logic, we first give the related notations and definitions.

(1) $P| \equiv M$. The entity P believes the message M.

(2) $P \lhd M$. The entity P sees the message M.

(3) $P| \sim M$. The entity P sent the message M.

(4) $P| \Rightarrow X$. The entity P fully controls the message X.

(5) $\#(M)$. The message M is fresh.

(6) $\{M\}_K$. The meessage M is encrypted by K.

(7) $< M >_Y$. The message M is sent combined with Y.

(8) $P \xleftrightarrow{SK} Q$. Entities P and Q share the secret SK.

(9) $P| \equiv \xrightarrow{K_Q} Q$. P believes Q's public key.

Then, the rules applied to the proof procedure are explained.

$R.1$ Message-meaning rule: If P believes K is shared between P and Q, and P receives the M encrypted by K, P will believe M was sent by Q.

$$\frac{P| \equiv P \xleftrightarrow{SK} Q, P \lhd M_K}{P| \equiv Q| \sim M}$$

We extend the message-meaning rule to the following message-signature rule and accumulator rule :

$R.2$ Message-signature rule: If M is signed by the private key of Q using the secure signature scheme and P believes the public key of Q, P will believe M was sent by Q.

$$\frac{P| \equiv \xrightarrow{K_Q} Q, P \lhd \{M\}_{K_Q^{-1}}}{P| \equiv Q| \sim M}$$

$R.3$ Accumulator rule: If P believes the accumulator ACC and the witness W_i of x_i, and P receives the message M that contains Q's value x_i, P will believe M was sent by Q.

$$\frac{P| \equiv \xrightarrow{ACC,W_i} Q, Q \lhd < M >_{x_i}}{P| \equiv Q| \sim M}$$

$R.4$ Nonce verification rule: It checks the freshness of the message M.

$$\frac{P| \equiv \#(M), P| \equiv Q| \sim M}{P| \equiv Q| \equiv M}$$

$R.5$ Jurisdiction rule: If P believes Q trusts and fully controls the message X, P believes X.

$$\frac{P| \equiv |Q| \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X}$$

$R.6$ Belief rule: If P believes Q trusts the message (X, Y), P also believes Q trust X.

$$\frac{P| \equiv Q| \equiv (X, Y)}{P| \equiv Q| \equiv X}$$

B. Formal proof of Intra-domain authentication

We analyze our protocol according to the proof steps discussed in work [42].

B.1 Goal of authentication:

Our proposed intra-domain authentication should meet the following goals. P is the IIoT device and Q is the server.

(1) $P| \equiv (P \xleftrightarrow{SK} Q)$ $(Goal\ 1)$;

(2) $Q| \equiv (P \xleftrightarrow{SK} Q)$ $(Goal\ 2)$;

B.2 Idealization process:

The process of intra-domain authentication is idealized as :

(1) The message M_1:

$P \longrightarrow Q : (Tid_{i,j}, j, N_1, s, TS_1, Z_1, \{x_i, D_1\}_{(PK_{sa})^{-1}})$.

(2) The message M_2:

$Q \longrightarrow P : (Z_2, < N_1, N_2, TS_2 >_{(D_1,x_i)})$.

(3) The message M_c:

$P \longrightarrow Q : (< Tid_{i,j}, N1, N2 >_{(D_1,x_i)})$. This message idealizes the upcoming communication process.

B.3 Assumption process:

The initial assumptions are made as follows:

S_1: $Q| \equiv P \xleftrightarrow{x_i} Q$; S_2: $P| \equiv P \xleftrightarrow{x_i} Q$;

S_3: $Q| \equiv \#(N_1, TS_1)$; $S_4 : P| \equiv \#(N_2, TS_2)$

S_5: $Q| \equiv \xrightarrow{PK_{i,j}} P$. This assumption shows that the server holds the child public key of the IIoT device.

S_6: $Q| \equiv P| \Longrightarrow SK$. S_7: $P| \equiv Q| \Longrightarrow SK$.

B.4 Proof process:

Now, we perform the BAN logic proof by applying the rules and assumptions.

(1) For message M_1, Q sees $(Tid_{i,j}, j, N_1, s, TS_1, Z_1, < x_i, D_1 >_{(PK_{sa})^{-1}})$. Considering the assumption S_5, the message-signature rule $(R.2)$ and the security of the schnorr signature proved in work [39], we get the step (2).

(2) $Q| \equiv P| \sim M_1$. By applying the nonce-verification rule to step (2), we get $Q| \equiv P| \equiv M_1$. The $server_a$ could use its private key sk_{sa} to obtain $D_1 = sk_{sa}N_1$. Then, we acquire the step (3) based on the belief rule $(R.6)$ and S_1.

(3) $Q| \equiv P| \equiv (D_1, N_1, x_i)$. For the message M_c, Q sees $< Tid_{i,j}, N1, N2 >_{(D_1,x_i)}$. Combining the proofs from step (3) with the message-meaning rule, we get the step (4).

(4) $Q| \equiv P| \sim M_c$. Combining the nonce-verification rule with the step (4), we acquire the step (5).

(5) $Q| \equiv P| \equiv (N_1, N_2, D_1, x_i)$. Considering the proofs from step (3), (5), $D_2 = n_1 N_2$, S_1, and the session key $SK = H_1(SID||D_1||D_2||x_i)$, we conclude that $Q| \equiv P| \equiv P \xleftrightarrow{SK} Q$. By applying the justification rule with the assumption S_6, we conclude that $Q| \equiv P \xleftrightarrow{SK} Q$ $(Goal\ 2)$.

(6) For the message M_2, D_1 can only be obtained by Q, since the ECDH problem cannot be broken. Considering the assumption S_2, we could utilize the message-meaning rule $(R.1)$ to obtain step (7).

(7) $P| \equiv Q| \sim M_2$. Integrating the S_4 into the nonce-verification rule, we get $P| \equiv Q| \equiv M_2$. Then, we apply the belief rule to obtain step (8).

(8) $P| \equiv Q| \equiv (N_1, N_2, D_1, D_2)$. From the S_2 and the step (8), and the session key $SK = H_1(SID||D_1||D_2||x_i)$, we conclude that $P| \equiv Q| \equiv P \xleftrightarrow{SK} Q$. By applying the justification rule to the assumption S_7, we conclude that $P| \equiv P \xleftrightarrow{SK} Q$ $(Goal\ 1)$.

C. Formal proof of Cross-domain authentication

C.1 Goal of authentication:

Our proposed cross-domain authentication should meet the following goals. P is the requestor $device_a$ and Q is the receiver $Server_b$. It is noted that $Server_b$ is able to forward the results securely to $device_b$. Thus, we replace the $device_b$ with the $server_b$ here for better understanding.

(1) $P| \equiv (P \xleftrightarrow{N_{5a}} Q)$ $(Goal\ 3)$;

(2) $Q| \equiv (P \xleftrightarrow{N_{5b}} Q)$ $(Goal\ 4)$;

The goal of the cross-domain authentication is that Q and P should authenticate each other and accept the random numbers N_{5a}, N_{5b}. The random numbers will be used to establish the session key $SK = n_{5a}N_{5b} = n_{5b}N_{5a}$ between $device_a$ and $device_b$ based on ECDHE.

C.2 Idealization process:

(1) The message M_{5a}:

$P \longrightarrow Q : (X, < K, R >_{x_i}, N_4, N_{5a}, < W_i^* >_{D_4})$

The $device_a$ sends the cross-domain request M_{5a} authorized by $server_a$ to $server_b$.

C.3 Assumption process:

S_1: $Q| \equiv \xrightarrow{ACC_i} P$. This assumption simulates that the $server_b$ could query the blockchain ledger to get the accumulator ACC_i of each domain.

S_2: $Q| \equiv \#(N_4, N_{5a}, TS_5)$; S_3: $Q| \equiv P| \Longrightarrow N_{5a}$

C.4 Proof process:

(1) For message M_{5a}, Q sees $(X, < K, R >_{x_i}, N_4, N_{5a}, < W_i >_{D_4})$. First, $D_4 = sk_{sb}N_4$ is calculated using the private key sk_{sb} of $server_b$ to verify the authenticity of the cross-domain request M_{5a} and acquire the real witness W_i of x_i. Then, the equation $e(W_i, Xg_{2a}^K) ==?e(ACC_i, R)$ is combined with S_1 to verify whether x_i is included in ACC_i. If so, we could use the accumulator rule $(R.3)$ to get step (2).

(2) $Q| \equiv P| \sim M_{5a}$. By applying the nonce-verification rule to step (2) and the S_2, we get $Q| \equiv P| \equiv M_{5a}$. Then, the step (3) is acquired based on the belief rule.

(3) $Q| \equiv P| \equiv (P \xleftrightarrow{N_{5a}} Q)$. Combining the justification rule with S_3, we get the step (4).

(4) $Q| \equiv P \xleftrightarrow{N_{5a}} Q$ $(Goal\ 3)$. The $server_b$ would send the result of the cross-domain authentication and $N_{5a}$to $device_b$.

Similarly, if the mutual authentication is performed, we get $P| \equiv P \xleftrightarrow{N_{5b}} Q$ $(Goal\ 4)$. The goals of the cross-domain authentication would be satisfied.

That is to say, $device_a$ and $device_b$ in different domains could authenticate each other. Moreover, the random numbers N_{5a} and N_{5b} will be used to negotiate the session key $SK = H_1(n_{5b}N_{5a}) = H_1(n_{5a}N_{5b})$ to protect the cross-domain public channel.

D. Discussion of Security Features and Functionalities

We first discuss the security features of our protocol.

(1) Intra-domain mutual authentication between IIoT device and server:

a) The IIoT device is authenticated by server by verifying the signature s. Assume that an adversary successfully forges a valid message M_1 to pass the server's authentication, and we can get $sP = PK_{i,j}H_4(x_i||N_1) + N_1$. Although the adversary does not know the secret key $sk_{i,j}$, the forking lemma could be invoked to continue the attack. The same random tape is chosen to input the same randomness. But the different output from the hash oracle is chosen to compute the valid login message. Therefore, the simulator could calculate $(s - s^*)(H_4(x_i||N_1) - H_4(x_i||N_1)^*)^{-1}$ as a solution to a given instance $(P, sk_{i,j}P)$, which contradicts the hardness of ECDL problem. Therefore, the adversary cannot forge the IIoT device to pass the authentication.

b) The server is authenticated by the hash value Z_2 . Assume that an attacker forges a valid message $M_2 = \{N_2, TS_2, Z_2 = H_2(D_1||*)\}$ to pass device's verification and the private key of the server is not revealed. It means the attacker has queried the hash oracle and computed $D_1^* = n_1 PK_s = sk_s n_1 P$. D_i could be the solution to solve the given instance $(P, sk_s P, n_1 P)$, which contradicts the hardness of ECDH problem. Thus, no attacker could impersonate the server to pass the verification.

(2) Cross-domain mutual authentication between IIoT devices: The IIoT device is authenticated by the on-chain accumulator ACC_i. The $server_b$ will first decrypt W_i^* to obtain witness W_i. Then, the accumulator ACC_i is queried from the blockchain to verify whether the value x_i of the $device_a$ is included in ACC_i through our $ZKPoK$ method. If so, the $device_a$ is authenticated by the $server_b$ as well as $device_b$.

As x_i is securely protected at device-side and also hidden in X, R, it is difficult for the adversary to access the correct x_i. The adversary can only forge a new valid value/witness pair to pass the verification, which contradicts the q-SDH assumption. Therefore, the unilateral cross-domain authentication is ensured. Similarly, the $device_b$ also could perform the mirror operations to establish the cross-domain mutual authentication.

(3) Anonymity and unlinkability: The identities of the IIoT devices are anonymous in our protocols. IIoT devices only transmit one-time pseudo identities $Tid_{i,j}$, or encode real identities into ID_b^* and W_i^*. As no real identity of IIoT devices will be revealed, the anonymity could be ensured.

To support unlinkability, only the domain information will be acquired from the blockchain ledger, and no device public keys will be queried during the authentication. Moreover, the pseudo identity $Tid_{i,j}$ is unlinkable and periodically updated. It is hard for the adversary to link two messages to the same requestor device. In general, our protocol can both reach the unlinkability and anonymity, thus, preserving the privacy.

(4) Security of multiple factors: The multiple factors are transformed into the parent secret keys k_i and c_i, and then stored in the server in the form of $K_i = k_iP$ and $C_i = c_iP$. The adversary cannot break the ECDL problem to retrieve parent secret keys and device factors. Therefore, the protection of the multiple factors at server-side could be ensured.

(5) Strong forward secrecy: The session keys of our protocols are computed as $SK = H_1(SID||D_1||D_2||x_i)$ and $SK = H_1(n_{5a}N_{5b})$. Assume that history messages as well as the secret keys (sk_{sa}, sk_{sb}, x_i) are disclosed. Without the correct random numbers, it is still difficult for the adversary to compromise the elements $D_2 = n_1N_2$ and $n_5 = H_4(random||x_i)$ and calculate the right session key. Thus, our protocol can reach strong forward secrecy.

Then, we present the security functionalities to show how our protocol resists various potential known attacks.

(1) Resistance to impersonation attack: If the attack wants to impersonate an authenticated IIoT device or sever, it should break our mutual authentication protocols. However, the security of our mutual authentication protocols has all been proved. The impersonation attack will not succeed.

(2) Resistance to replay attack: The random number mechanism is applied to our protocol. As the random numbers N_1 to N_5 are different for each session, the adversary cannot replay valid authenticated requests to pass the verification. Hence, our protocol resists the replay attack.

(3) Resistance to physical attack and loss of factors attack: The PUF circuit is assumed to be secure, and the adversary cannot clone or predict the right PUF response R to compute the PUF seed $H_1(R||r_i)$. If factors of the corrupted IIoT device are leaked, without the PUF seed, the adversary still cannot correctly encode these factors to calculate the right c_i. As a result, our protocol could prevent the loss of factors attack.

(4) Resistance to desynchronization attack: This attack will intercept the channel and make only one side of the communicators update the $Tid_{i,j}$. However, the pseudo identity list Pid is used to resist this kind of attack as mentioned in [41]. If the desynchronization happens, any of the unused pid_i could replace the $Tid_{i,j}$ and establish the synchronization again.

VII. Performance Evaluation

We conducted the experiment to evaluate the performance of our protocol. The results of six critical metrics, including efficiency, computation and communication overhead, on-chain storage overhead, the performance of smart contract, and the comparison of functionalities, are discussed in this section.

A. Experiment Settings

1) Entities in IIoT domains: We simulated two IIoT domains in our experiment. Each domain contains an IIoT device, a server, a trusted authority. Servers and trusted authorities in two domains were deployed on a laptop with Intel core i5-10210U @1.6GHz and 16GB Memory. The operations of IIoT devices were executed on two Raspberry Pis 3B+.

2) Blockchain network: The blockchain network and the smart contract were implemented by open-source project Hyperledger Fabric. Blockchain clients realized by Java-SDK were installed on both servers and trusted authorities. They could invoke or query the smart contract to manage the blockchain ledger.

3) Multiple factors in IIoT device: We implemented three types of factors in IIoT devices, including the PUF key, the biometric key, and the serial number. The DRAM PUF introduced in our previous work [43] provided the 128-bit PUF response in our experiment. Moreover, the biometric feature of the user was derived using the widespread face feature extraction project Face_Recognition. Then, a fuzzy extractor was realized by BCH error correction code, to transform the biometric feature and the PUF response into the 128-bit PUF key and biometric key respectively. Last, the serial number was simply loaded from the device storage.

4) Algorithms: Our proposed protocols were implemented on the application layer based on HTTP protocol, which can be conveniently ported to CoAP in IIoT applications. The cryptography algorithms were realized by the Java JPBC 2.0.0 library and BouncyCastle 1.60 library. It is noted that the Type-F pairing and the $secp256r1$ elliptic curve were chosen in our implementation.

The average costs of all the cryptography operations used in our protocols are shown in Table II, where H is the general hash function, SM_{r1}, SM_1, SM_2 denote the scalar multiplications in $G_{r1}/G_1/G_2$, Exp_1, Exp_2, Exp_T are the exponentiations $G_1/G_2/G_T$, PM_2 and PM_{r1} are the point additions in G_2/G_{r1}, M_T is the multiplication in G_T, and BP denotes the bilinear pairing.

TABLE II
AVERAGE COSTS FOR CRYPTOGRAPHY OPERATIONS(UNIT: MILLISECOND)

Notations	Device	Server	Notations	Device	Server
H	1.02	0.01	Exp_1	51.62	5.52
SM_{r1}	27.45	1.53	Exp_2	105.86	8.92
SM_1	56.89	5.87	Exp_T	478.13	31.24
SM_2	114.92	9.26	PM_2	0.38	0.03
M_T	4.13	0.14	BP	2168.47	132.94
$PAr1$	0.10	0.01			

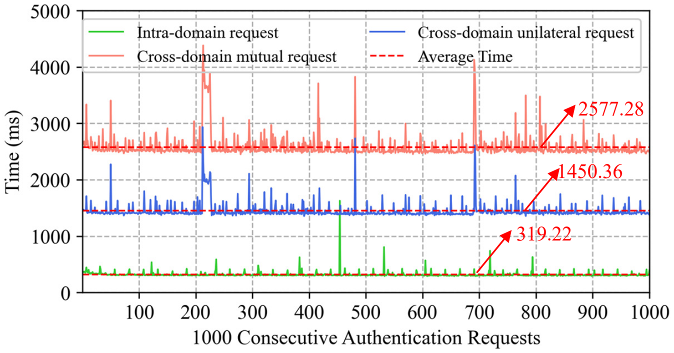


Fig. 7. Experiment results of consecutive authentication request. All the devices take the PUF key and the serial number as factors, and only the requestor ($device_a$) of the cross-domain authentication takes an extra biometric key.

B. Efficiency

We proposed 1000 distinct consecutive intra-domain and cross-domain mutual authentication requests. The actual time cost of each request is recorded in Fig. 7 with the average time marked in red. As shown in Fig. 7, the intra-domain request only takes 319.22 ms, and the unilateral cross-domain request takes 1450.36 ms, which already includes the time of deriving the biometric feature at the requestor side ($device_a$). And, it costs about 2577.28 ms to establish the mutual cross-domain authentication, where the requestor and the receiver ($device_a$ and $device_b$) authenticate each other and establish the session key. Moreover, Fig. 7 also shows that the real time of each request remains stable and varies around a red dashed line, which depicts the average value of executing time.

In summary, the performance of our protocols is reliable and efficient for IIoT applications.

TABLE III
STATS ON TIME-CONSUMING CRYPTOGRAPHY OPERATIONS (IDA: INTRA-DOMAIN AUTHENTICATION, CDA: UNILATERAL CROSS-DOMAIN AUTHENTICATION, KN:KEY NEGOTIATION)

Entity	IDA+KN	CDA
$device_a$	$5SM_{r1}$	$4SM_{r1}$+$2Exp_2$
$server_a$	$6SM_{r1}$+$2PA_{r1}$	$6SM_{r1}$+$2PA_{r1}$
$device_b$	$5SM_{r1}$	/
$server_b$	$6SM_{r1}$+$2PA_{r1}$	SM_{r1}+Exp_2+PM_2+$2BP$

TABLE IV
COMPARISON OF COMPUTATION OVERHEAD ON DEVICES DURING MUTUAL CDA AND KN

Schemes	Costs on Subject Device	Costs on Object Device
$BlockCAM$ [10]	$4SM_{r1}$+PA_{r1}	$4SM_{r1}$+PA_{r1}
$SCCA$ [12]	$3SM_{r1}$	$3SM_{r1}$
$BASDA$ [4]	$2SM_{r1}$+SM_1+Exp_T	$2SM_{r1}$+SM_1+Exp_T
Our	$6SM_{r1}$ +$2Exp_2$	$6SM_{r1}$ +$2Exp_2$

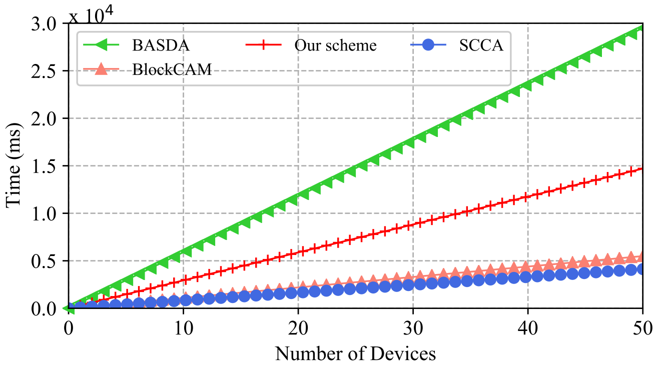


Fig. 8. Simulation result of the computation costs on IIoT devices during the mutual cross-domain authentication (CDA) and key negotiation (KN).

C. Computation Overhead

To evaluate the computation overhead, we first count the most time-consumption operations. Then, we presented the simulation results to compare our protocol with related studies.

1) Theoretical analysis: The most time-consumption operations executed by all entities are summarized in Table III. It takes the IIoT device $5SM_{r1}$ and the server $6SM_{r1}$+$2PA_{r1}$ to establish the intra-domain authentication and key negotiation. To process the unilateral cross-domain authentication request, the subject $device_a$ and the $server_a$ cost $4SM_{r1}$+$2Exp_2$ and $6SM_{r1}$+$2PA_{r1}$ respectively. And, it takes the $servr_b$ $3SM_{r1}$+Exp_2+PM_2+PA_{r1}+$2BP$ to verify the request and the object $device_b$ does nothing but to receive the result and the random number N_5. The unilateral cross-domain is a symmetrical process, thus, the overhead for mutual cross-domain request is easy to conclude. Since IIoT devices only take few SM_{r1} operations and the server helps to afford many heavy computation, the overhead would be acceptable for IIoT devices, which were defined in the system model.

2) Simulation results: We compare the computation overhead on IIoT devices of our protocol with other blockchain-based cross-domain authentication mechanisms BlockCAM [10], SCCA [12], and BASDA [4] . The server is assumed to have decent computation resources, thus, we omit the comparison of the costs on servers. As can been seen from Table IV, the subject device ($device_a$) and the object device ($device_b$) have the same costs during the symmetric cross-domain authentication process, and there is no heavy operation like bilinear pairing executed by devices. Moreover, the simulation results calculated at the same cryptography settings are presented in Fig. 8. It indicates that the computation cost of all the mechanisms during the mutual CDA and KN is liner to the number of the devices. It is also easy to find that our protocol outperforms the BASDA [4], and the mechanisms BlockCAM and SCCA have the similar computation overhead, which is lower than our protocol.

Combining the theoretical analysis with the simulation results, the computation overhead of our protocol is moderate and suitable for IIoT devices defined in our system model.

D. Communication Overhead

We evaluate the communication overhead in this part. The output of the H_2 is 256 bits, the signature s is 256 bits, the element X is 128 bits, the length of the element in $G/G_1/G_2$ is 256 bits, 256 bits, and 1024 bits. The real identity ID, TS, j are all 32 bits, and $Tid_{i,j}$ is 128 bits. According to these settings, the $device_a$ sends the 156 bytes cross-domain request M_3 to $server_a$, and receives the 185 bytes authorized request M_4. Then, the $devcie_a$ encodes the value x_1 into the 709 bytes message M_5. The message M_5 will be forwarded to the cross-domain $server_b$, to establish the cross-domain authentication. The $device_b$ could perform mirror operations as $device_a$ to achieve the mutual authentication and key negotiation.

The cross-domain requests in our protocol and BASDA [4] all need to be authorized by the domain server. However, the devices in other two mechanisms BlockCAM [10] and SCCA [12] can directly send the cross-domain requests without asking for the authorizations, thus bringing less communication overhead. Our protocol affords more communication costs than the most relevant work BASDA [4], which in all takes 768 bytes. The difference gap is mainly caused by the fact that the device in our protocol should insert the elements X, K, R, W_i^* into the cross-domain request message M_5, which are used to construct the ZKPoK method for the accumulator.

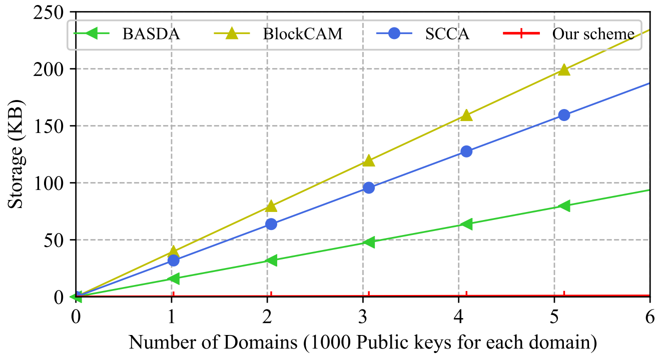


Fig. 9. Storage overhead in the blockchain.

E. Storage Overhead on the Blockchain

To demonstrate the advantage of our protocol in consideration of the storage overhead on the blockchain, we did the comparison by setting 1000 public keys for each domain. BlockCAM [10], SCCA [12], BASDA [4] will take 39.06 KB, 31.25 KB, and 15.63 KB on-chain storage resources for each domain respectively, and our protocol only occupies 0.14 KB for each domain. As is shown in Fig. 9, our protocol outperforms the other three mechanisms obviously with the growth of the number of domains. The advantage of our protocol could be explained by using our proposed trust-building method. This novel method stores the accumulator in the blockchain instead of a great number of the public keys or certificates. In general, our work indeed reduces the on-chain storage overhead greatly from the aspect of the application protocol layer.

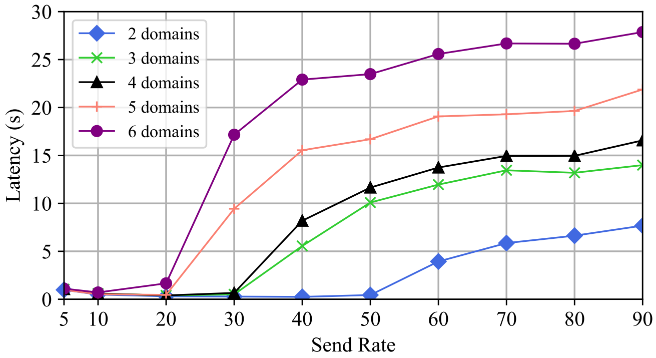


Fig. 10. Experiment results of invoking the smart contract in different blockchain networks.

F. Performance Evaluation of Smart Contract

In our protocol, the authentication phase only queries the smart contract. And, the smart contract will only be invoked during the registration and revocation processes. Hence, we implemented blockchain networks with different domains to evaluate the performance of smart contract in this part.

To initialize the blockchain, parameters $BatchTimeout$ and $BatchSize$ are configured as 2 s and 10. We also set the absolute maximum bytes of one block as 99 MB, and the preferred maximum size as 512 KB. There are no transaction fees in our consortium blockchain, and these basic settings are almost the same as work [4]. The only difference is that $BatchTimeout$ in their work was set as 0.05 s. Besides, we simulated five kinds of networks, which consist of different number of IIoT domains. Each domain contains two blockchain nodes that respectively act as the server and trusted authority.

To evaluate the performance, we recorded the latency of querying and invoking the smart contract. The server only queries the smart contract to retrieve data from the local copy of the blockchain ledger, which is also built as a key-value state database. Hence, the concurrent queries will not bring latency problems. The query latency always keeps an efficient value to be around 19.59 ms in our protocol.

The open-source project Hyperledger Caliper was leveraged to record the invoke latency and simulate concurrent transactions proposed at different send rates. The latency is recoded from the time point that the smart contract is invoked to the time point that the data is recorded into the blockchain.

As is shown in Fig. 10, the invoke latency reaches the bottom when the send rate is 10 transactions per second (TPS), which right equals to the parameter $BatchSize$. If the send rate is less than 20 TPS, the latency is low and will not exceed

TABLE V
SUMMARY OF SECURITY FEATURES AND FUNCTIONALITY

Security features and functionalities	BlockCAM [10]	SCCA [12]	BASDA [4]	Our protocol
Mutual authentication	Yes	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes	Yes
Unlinkability	No	No	Yes	Yes
Resistance to loss of factors attack	No	No	No	Yes
Resistance to replay attacks	Yes	Yes	Yes	Yes
Strong forward secrecy	No	Yes	Yes	Yes
[a]Efficiency	Yes	Yes	No	Yes
Low on-chain storage overhead	No	No	No	Yes

[a]This feature presents that the time of authentication process will not be restricted by the throughputs of the blockchain system.

the $BatchTimeout$ (2 seconds). However, the invoke latency will increase with more concurrent transactions needing to be processed. There appears an interesting phenomenon that a sharp increase point occurs for each kind of network with the increasing send rate. The latency increases sharply at different sharp points, where the send rate is 20, 30, and 50 TPS. It indicates that the more domains the network has, the more early the sharp point comes. This finding is as expected, since our instanced blockchain network with more domains is required to spend more time on verifying endorsed proposals and validating sorted transactions.

The throughput of our networks is restricted to be less than 50 TPS. However, it will not affect the efficiency of authentication, as there is no need to propose transactions during the authentication process in our protocol. Moreover, when deployed in industrial environments, the throughput of the blockchain network could reach 3500 TPS, even 20000 TPS [44]. In general, the query latency is satisfied to ensure efficiency. And, the performance of the invoke latency would not influence efficiency and could be further improved by optimizing the blockchain network as the work [44] did.

G. Comparison of Security Features and Functionality

In this part, we compare the security features and functionalities of our work with relevant protocols [4], [10], [12]. Table V summaries the similarities and differences to show the novel security properties that our work achieves.

As is shown in Table V, mutual authentication, anonymity, and resistance to replay attack are basic security requirements and all supported in state-of-the-art protocols. Besides, if secure channels need to be built as required in [4], [12] and our work, strong forward secrecy will also be provided to protect session keys. Protocol [4] is not regarded as efficiency, as new transactions need to be proposed during the authentication process. The unlinkability is only achieved in our solution and protocol [4]. However, only our work ensures efficiency and unlinkability at the same time. Furthermore, our work supports resistance to loss of factors attacks and low on-chain storage overhead, which are all not included in other protocols.

This comparison shows that only our protocol provides important properties of resistance to loss of factors attacks and low on-chain storage overhead. In addition, our solution is also a novel attempt to both achieve efficiency and unlinkability in blockchain-based cross-domain authentication.

VIII. Conclusion

We have developed an efficient and privacy-preserving multi-factor device authentication protocol using blockchain for cross-domain IIoT, to secure the cross-domain device collaborations. The formal security proof has been given by BAN Logic, and the security discussion shows that our protocol provides the protection of the multi-factor database as well as the resistance to the loss of factor attack. Moreover, the anonymity and unlinkability are also ensured to preserve the privacy. The performance evaluation shows that our intra-domain, unilateral cross-domain, and mutual cross-domain protocols are efficient and reliable to take 319.22 ms, 1450.36 ms, and 2577.28 ms respectively, and the on-chain storage overhead has been reduced to 0.14 KB for each domain with 1000 public keys. Finally, the performance of the smart contract is evaluated to show the scalability.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 11, pp. 4724–4734, 2018.

[2] T. Qiu, J. Chi, X. Zhou, Z. Ning, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–1, 2020.

[3] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[4] M. Shen, H. Liu, L. Zhu, K. Xu, and M. Guizani, "Blockchain-assisted secure device authentication for cross-domain industrial iot," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, 2020.

[5] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2017.

[6] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 844–859, 2021.

[7] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[8] J. Wang, L. Wu, K. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2019.

[9] G. Ali, N. Ahmad, Y. Cao, S. Khan, H. Cruickshank, E. A. Qazi, and A. Ali, "xdbauth: Blockchain based cross domain authentication and authorization framework for internet of things," *IEEE Access*, vol. 8, pp. 58 800–58 816, 2020.

[10] W. Wang, H. Ning, and L. Xin, “Blockcam: A blockchain-based cross-domain authentication model,” in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Conference Proceedings.

[11] L. Ao, H. Cruickshank, C. Yue, P. Asuquo, C. Ogah, and Z. Sun, “Blockchain-based dynamic key management for heterogeneous intelligent transportation systems,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[12] G. Li, Y. Wang, B. Zhang, and S. Lu, “Smart contract-based cross-domain authentication and key agreement system for heterogeneous wireless networks,” *Mobile Information Systems*, vol. 2020, no. 29, pp. 1–16, 2020.

[13] H. Zhang, X. Chen, X. Lan, H. Jin, and Q. Cao, “Btcas: A blockchain-based thoroughly cross-domain authentication scheme,” *Journal of Information Security and Applications*, vol. 55, p. 102538, 2020.

[14] C. Lin, D. He, X. Huang, N. Kumar, and K. Choo, “Bcppa: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–13, 2020.

[15] S. Guo, F. Wang, N. Zhang, F. Qi, and X. Qiu, “Master-slave chain based trusted cross-domain authentication mechanism in iot,” *Journal of Network and Computer Applications*, vol. 172, p. 102812, 2020.

[16] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, “A blockchain-based mutual authentication scheme for collaborative edge computing,” *IEEE Transactions on Computational Social Systems*, vol. PP, no. 99, pp. 1–13, 2021.

[17] “The pythia prf service,” in *USENIX 2015*, 2015.

[18] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[19] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, “A privacy-aware pufs-based multi-server authentication protocol in cloud-edge iot systems using blockchain,” *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[20] H. Chen and Y. Wang, “A lightweight scalable protocol for public blockchain,” *Journal of Computer Research and Development*, vol. 57, no. 7, pp. 1555–1567, 2020. [Online]. Available: ¡Go to ISI¿://CSCD:6759334

[21] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, “A blockchain-based privacy-preserving authentication scheme for vanets,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–10, 2019.

[22] H. Gunasinghe and E. Bertino, “Privbiomtauth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones,” *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2017.

[23] W. Liu, X. Wang, and W. Peng, “Secure remote multi-factor authentication scheme based on chaotic map zero-knowledge proof for crowdsourcing internet of things,” *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2019.

[24] H. N. Dai, Z. Zheng, and Y. Zhang, “Blockchain for internet of things: A survey,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[25] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derh Ab, and H. Janicke, “Blockchain technologies for the internet of things: Research issues and challenges,” *IEEE Internet of Things Journal*, vol. PP, no. 99, 2018.

[26] K. Peng, M. Li, H. Huang, C. Wang, and K. Choo, “Security challenges and opportunities for smart contracts in internet of things: A survey,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2021.

[27] M. A. Ferrag and S. Lei, “The performance evaluation of blockchain-based security and privacy systems for the internet of things: A tutorial,” *IEEE Internet of Things Journal*, vol. PP, no. 99, 2021.

[28] C. Wu, J. Lu, W. Li, H. Meng, and Y. Ren, “Master-slave blockchain based cross-domain trust access mechanism for upiot,” in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, Conference Proceedings.

[29] L. Wang, Y. Tian, and D. Zhang, “Toward cross-domain dynamic accumulator authentication based on blockchain in internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2858–2867, 2021.

[30] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, “Iot passport: A blockchain-based trust framework for collaborative internet-of-things,” in *the 24th ACM Symposium*, Conference Proceedings.

[31] M. Ma, G. Shi, and F. Li, “Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the iot scenario,” *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2019.

[32] 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.

[33] R. Chen, F. Shu, S. Huang, L. Huang, H. Liu, J. Liu, and K. Lei, “Bidm:a blockchain-enabled cross-domain identity management system,” *Journal of Communications and Information Networks*, vol. 6, no. 1, p. 15, 2021.

[34] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, “Blockchain for secure and efficient data sharing in vehicular edge computing and networks,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[35] N. Lan, “Accumulators from bilinear pairings and applications,” *Springer, Berlin, Heidelberg*, 2005.

[36] J. Wang, L. Wu, K. Choo, and D. He, “Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure,” *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2019.

[37] MEMBER, IEEE, D. Dolev, and A. C. Yao, “On the security of public key protocols,” *Information Theory IEEE Transactions on*, vol. 29, no. 2, pp. 198–208, 1981.

[38] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, “Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018.

[39] C. P. Schnorr, “Efficient identification and signatures for smart cards,” in *Advances in Cryptology — CRYPTO’ 89 Proceedings*, G. Brassard, Ed. Springer New York, Conference Proceedings, pp. 239–252.

[40] M. Burrows, M. Abadi, and R. M. Needham, “A logic of authentication,” *Acm Transactions on Computer Systems*, vol. 23, no. 5, pp. 1–13, 1989.

[41] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, “Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, p. 1, 2019.

[42] N. Lwamo, L. Zhu, C. Xu, K. Sharif, X. Liu, and C. Zhang, “Suaa: A secure user authentication scheme with anonymity for the single & multi-server environments,” *Information Sciences*, 2018.

[43] S. Chen, B. Li, and Y. Cao, “Intrinsic physical unclonable function (puf) sensors in commodity devices,” *Sensors*, vol. 19, no. 11, pp. 2428–, 2019.

[44] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, “Fastfabric: Scaling hyperledger fabric to 20000 transactions per second,” *International Journal of Network Management*, vol. 30, 2020.

Yan Zhang Yan Zhang was born in Changzhou, JiangSu province, China in 1993. He received the B.S degree and the M.S degree from Xidian University in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree with school of Cyber Science and Engineering, Southeast University.

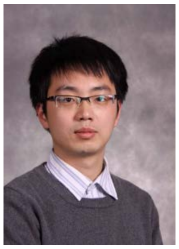
His main research is information security, including blockchain and Internet of Things.

Bing Li Bing LI was born in Nanjing, JiangSu province, China, in 1968. He received his B.S. degree in Electronics Science and Technology from Southeast University in 1991, and Ph.D. from Southeast University in 2004. He is currently Professor and tutor of doctoral students with the School of Microelectronics, the School of Cyber Science and Engineering, Southeast University, and the director of the joint research center for advanced cloud system of Southeast University.

His main research is the efficient and secure integrated circuits and system, including data compression, data encryption, Physical Unclonable Functions, the blockchain, Internet of Things, and the area of information security.

Jiaxin Wu was born in Wuxi, Jiang Su province, China, in 1993.She received her Master degree in Electronic and Communication Engineering from Northwest Normal University in 2018.Since 2018, she has been studying for her doctor degree in Electronics Science and Technology (Integrated Circuit Design) from Southeast University.

Her main research is Speech signal processing, Internet of Things, and the area of information security.



Bo Liu received the BEng degree from the Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004. He then received the MEng. and PhD. Degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively. He is currently a Senior Lecturer with the University of Technology Sydney, Australia.

His research interests include cybersecurity and privacy, location privacy and image privacy, privacy protection and machine learning.

Rui Chen received the B.Sc. and M.Sc. degrees in electronics engineering from Southeast University, Nanjing, China, in 2008 and 2011, respectively, and the Ph.D. degree in microelectronics from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2014.He is currently an associate professor with the School of Computer and Software Engineering, Nanjing Vocational University of Industry Technology, Nanjing, China.

His current research interests include reconfigurable architecture and cryptography hardware.

Jinke Chang received the Ph.D degree in medical science and engineering from University College London, London, U.K., in 2022. He is currently a research fellow with the Division of Surgery and Interventional Science, University College London.

His research interests include smart materials, wearable devices, implantable biomedical devices, acoustic communication and machine learning.