Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT

Meng Shen, *Member, IEEE*, Huisen Liu, Liehuang Zhu, *Member, IEEE*, Ke Xu, *Senior Member, IEEE*, Hongbo Yu, Xiaojiang Du, *Fellow, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*— Industrial Internet of Things (IIoT) is considered as one of the most promising revolutionary technologies to prompt smart manufacturing and increase productivity. With manufacturing being more complicated and sophisticated, an entire manufacturing process usually involves several different administrative IoT domains (e.g., factories). Devices from different domains collaborate on the same task, which raises great security and privacy concerns about device-to-device communications. Existing authentication approaches may result in heavy key management overhead or rely on a trusted third party. Thus, security and privacy issues during communication remain unsolved but imperative. In this paper, we present an efficient blockchain-assisted secure device authentication mechanism BASA for cross-domain IIoT. Specifically, consortium blockchain is introduced to construct trust among different domains. Identity-based signature (IBS) is exploited during the authentication process. To preserve the privacy of devices, we design an identity management mechanism, which can realize that devices being authenticated remain anonymous. Besides, session keys between two parties are negotiated, which can secure the subsequent communications. Extensive experiments have been conducted to show the effectiveness and efficiency of the proposed mechanism.**

***Index Terms*— Industrial Internet of Things (IIoT), secure cross-domain authentication, key agreement, consortium blockchain, identity-based cryptography.**

I. Introduction

WITH the proposal of Industry 4.0 [1] and other similar concepts [2] mentioned frequently in recent years, Industrial Internet of Things (IIoT) is deemed as one of the crucial enabling technologies [3] to put these concepts into practice. It not only connects devices but also links them to the Internet, providing diverse services [4]–[6] for manufacturing.These Internet services are usually provided with privacy-preserving under cloud-based environment [7], [8]. Inter-connectivity makes it possible for devices to work collaboratively to significantly improve efficiency and productivity with the assistance of Internet services.

It has become a trend that devices inside an administrative domain (e.g., factory) using IIoT technologies connect to automate manufacturing tasks, which can significantly improve productivity and reduce management cost. However, it is hard to have a complete product manufactured in a standalone domain as manufacturing is getting more sophisticated. The entire production process is more likely to be spanned across several domains that have a cooperation relationship. In such a scenario, devices located in different domains need to communicate with each other to exchange information for better collaboration.

Although devices in different domains can be easily connected via widely-used networking infrastructures, establishing secure communication among them is a non-trivial task because quite a few issues on trust and security remain unsolved. Domains do not necessarily trust each other as one is usually reluctant to make its sensitive data acquirable by others. For instance, a factory administrator will not allow its devices accessed by any device outside its administrative domain without being authenticated. In fact, most existing authentication mechanisms are built on well-known Public Key Infrastructure (PKI) systems, where a trusted third party named certificate authority (CA) is involved to provide the root of trust for all the PKI certificates. Certificates are used to authenticate the identities of individuals, devices, and other entities, but it may also introduce heavy management cost. Besides, CAs are vulnerable to potential attacks and prone to operational errors [9].

Identity-based cryptography (IBC) [10] is a type of public-key cryptography in which a publicly known string representing the identity of an entity is used as the public key. In IBC systems, a trusted party named Key Generation Center (KGC) is responsible for creating the private key based on the identity of an entity. This mechanism is widely used in a closed domain, where an administrator has full control of the devices in the domain. IBC cannot be directly used for cross-domain device authentication, as one domain lacks

Manuscript received July 16, 2019; revised January 14, 2020; accepted February 26, 2020. Date of publication March 16, 2020; date of current version May 7, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803405, in part by the National Natural Science Foundation of China under Grant 61972039, Grant 61932016, and Grant 61872041, in part by the Beijing Natural Science Foundation under Grant 4192050, in part by China National Funds for Distinguished Young Scientists under Grant 61825204, in part by the Beijing Outstanding Young Scientist Program under Grant BJJWZYJH01201910003011, and in part by the Beijing National Research Center for Information Science and Technology (BNRist) under Grant BNR2019RC01011. *(Corresponding author: Liehuang Zhu.)*

Meng Shen, Huisen Liu, and Liehuang Zhu are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenmeng@bit.edu.cn; liuhuisen@bit.edu.cn; liehuangz@bit.edu.cn).

Ke Xu and Hongbo Yu are with the Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China, and also with the Department of Computer Science, Tsinghua University, Beijing 100084, China (e-mail: xuke@mail.tsinghua.edu.cn; yuhongbo@mail.tsinghua.edu.cn).

Xiaojiang Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj@ieee.org).

Mohsen Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.2980916

of control of devices in another domain due to the peering relationship among domains.

Consortium blockchain is a kind of distributed ledger maintained by several cooperated peer nodes. It is a permissioned blockchain [11] whose network nodes have to be verified before joining the network. The structure of such participated nodes is similar to business partnership. These nodes do not fully trust each other, but they are regulated under certain contracts and work collaboratively. Blockchain has been used as the supporting technology in multi-party solutions [12], [13]. Thus, consortium blockchain can be exploited to construct trust among different domains, where each domain has a representative node responsible for maintaining the global ledger.

The combined usage of consortium blockchain and IBC can be a challenging task:

- *Revocation of Identity*. IBC uses the identity of an entity as its public key, which makes revocation of public key burdensome and inflexible under the compromise of the corresponding private key. If a user uses his email as the public key, it is unrealistic to forbid his further use of email account once the private key being compromised.
- *Identity Privacy-preserving*. The identity may disclose the privacy of an entity. For instance, an adversary can easily know, by intercepting packets sent from or received by an entity, which type of services the entity accesses or with whom it communicates.
- *Storage Limitation*. Blockchain techniques may introduce time latency as new transactions should be validated and verified before being written in the ledger. Storage limitation is another practical problem as block size in the blockchain is restricted to a certain size. These constraints result in a bottleneck in throughput.

To tackle the aforementioned challenges, in this paper, we propose a Blockchain-Assisted Secure Authentication mechanism (BASA) for cross-domain IIoT. With the ingenious design, the public key can be easily invoked if needed. It also enables a device to be authenticated by other devices in a different administrative domain without exposing its identity information. On this basis, the session key is negotiated for the following information exchange securely.

The main contributions of this paper are as follows:

- We propose an efficient and secure blockchain-assisted authentication mechanism (BASA), which supports the authentication of devices located in different IIoT domains. Considering the storage limitation, we design off-blockchain storage to reduce the data to be written on the blockchain, which eliminates the throughput bottleneck.
- We propose an identity management method to remedy the drawback of IBC to revoke the public key of an entity when the corresponding privacy is compromised. Meanwhile, based on this design, devices can be anonymously authenticated (i.e., without exposing its real identity) by those in a different administrative domain.
- We propose a key agreement mechanism to negotiate session keys between a pair of devices. With the negotiated session keys, devices can securely exchange information to work collaboratively.
- We conduct security analysis to demonstrate the security and privacy guarantees provided by BASA. We also conduct simulation-based experiments to evaluate its performance in terms of computation overhead, communication overhead, write latency, etc. Experimental results show the effectiveness and efficiency of BASA.

The remainder of the paper is organized as follows. Section II reviews the existing authentication mechanisms and Section III states the problem. Section IV gives the overview of the proposed solution, which is followed by the design details in Section V. Extensive experiments are conducted to demonstrate the effectiveness of BASA in Section VI and some discussions are provided in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

There already exist lots of authentication mechanisms for IoT applications. From the perspective of the key being exploited, authentication mechanisms can be classified into two categories: symmetric key based mechanisms and public key based mechanisms, which are summarized in Table I.

A. Symmetric Key Based Mechanisms

Symmetric key based mechanisms are known for their fast operations of the cryptographic primitives. However, it is inevitable to have a secret key pre-distribution phase, which is usually implemented by a Key Distribution Center (KDC), e.g., Kerberos [14], to reduce the risks inherent in the key exchange. Message Authentication Code (MAC) is the most commonly used technique in symmetric key based authentication mechanisms. Bellare *et al.* [15] propose two related schemes, the Nested construction (NMAC) and the Hash-based MAC (HMAC), which are proven to be secure as long as there is an underlying hash function. HMAC can be utilized with any iterative cryptographic hash function [16]. The problem is that these mechanisms lack of scalability to deploy large-scale devices. Thus, symmetric key based mechanisms are more suitable for relatively small-scale IoT applications.

Public key based mechanisms possess good scalability characteristics. It relies on the fact that the keys are created pairwise, and encrypted data by private key (public key) can only be decrypted by the corresponding public key (private key). Public keys can be transmitted in an insecure channel while the private key is kept secretly on the owner's side.

B. Certificate-Based Mechanisms

Porambage *et al.* [17] propose a two-phase implicit certificate-based authentication mechanism for wireless sensor networks. The cryptographic credentials are stored in edge nodes, which exposes the mechanism to cloning attacks. The authors in [18] propose a key management protocol for mobile and IIoT systems, which provides node authentication and key negotiation. Elliptic Curve Qu-Vanstone (ECQV) implicit certificate and the Elliptic-Curve Diffie–Hellman (ECDH) technique are exploited in that mechanism. In [19], the authors

TABLE I
SUMMARY OF TYPICAL AUTHENTICATION MECHANISMS

Mechanisms	Techniques	Pros	Cons
Symmetric key based	HMAC and NMAC [15]	Computation-efficient; Easy implementation	Key pre-distribution; Message-oriented authentication
	HMAC [16]	Computation-efficient; Easy implementation	Key pre-distribution; Message-oriented authentication
	Kerberos [14]	Interoperability; Communication-efficient	KDC-dependent; DOS attack
Certificate based	ECC [17]	Computation-efficient	CA-dependent
	ECQV and ECDH [18]	Communication-efficient	CA-dependent; Lack of forward security
	ECDSA and ECDH [19]	Computation-efficient	CA-dependent; Smart gateway-dependent
	DTLS and CoAP [20]	Interoperability; Lightweight	CA-dependent
Identity based	IBE, IBS and EDH [21]	Certificate-free; Forward security	Private key escrow; Public key revocation-hard
	IBE and IBS [22]	Certificate-free	Private key escrow; Public key revocation-hard

develop an authentication and authorization architecture for IoT-based healthcare relying on the certificate-based DTLS handshake protocol and smart e-health gateways are needed.

There are also many other certificate-based mechanisms [20], [23], [24], which afford a lot to maintain the PKI and implicitly put trust in CAs. However, CAs are vulnerable to potential attacks and prone to operational errors. Failures of CAs have been observed all around the world [9].

C. Identity-Based Mechanisms

With the development of identity-based cryptography (IBC), researchers try to apply it into authentication uses. An identity-based mutual device authentication scheme is developed in [21] for power line communication (PLC). The complexity of deploying and managing authentication credentials is reduced as no public key certificates are utilized. Li *et al.* [22] propose an identity-based authentication for cloud computing, which is valuated to be more efficient than SSL Authentication Protocol. However, since the authenticated parties are cloud server and device user, the mutual authentication of peer devices is not considered.

Most of these authentication mechanisms focus on the authentication for the single-domain IoT application. An authentication mechanism presented in [25] considers the cross-domain scenario. However, several security gaps of the work have been found in [26], in which improvements are proposed to satisfy the security properties of the application scenario. However, certificates are inevitable to be exploited.

D. The Novelty of the Paper

In this paper, we propose a blockchain-assisted secure device authentication mechanism for cross-domain industrial IoT. We employ and extend the IBS techniques for cross-domain authentication without introducing any trusted third party. In such a condition, public key certificates are no longer needed, which reduces the heavy work of digital certificate issuing, maintaining and revoking. Besides, we design a flexible identity management mechanism, which can efficiently revoke the identities (public keys) of IIoT devices and preserve the privacy of IIoT devices.

III. PROBLEM STATEMENT

In this section, we first depict an application scenario with cross-domain authentication requirements, where devices in different IIoT administrative domains work collaboratively. Then, we discuss the security threats which may occur and identify the design goals.

A. Application Scenarios

As sensing and actuating technologies are rapidly developed, IIoT is paving the way to connect more devices not only in a single manufacturing domain but also in more relevant domains. Fig. 1 depicts a simplified application scenario where two manufacturing domains are involved. These two factories may be operated by two business partners.

Within each factory, IIoT devices equipped with sensors, processors, actuators and other components are deployed in the production line. They can perceive the surrounding environment such as temperature, humidity, or sense the status of products being manufactured. Further, they make decisions based on collected data to timely change their behaviors for reducing losses or optimizing product manufacturing. Two factories in Fig. 1 participate in the manufacturing process of the same productions. During the manufacturing process, devices in such two factories need extensive communications. Thus, devices have to be authenticated before permitting others to access their data.

B. Security Threats

The scenario described in Fig. 1 seems to be promising, as traditional single domains can be more open and interconnected, which can significantly reduce the management cost and heavily improve productivity. However, several security and privacy problems should be seriously considered.

As IIoT devices are connected to the Internet, they are exposed to many cyber attacks. When the sensitive sensing data is exchanged over the Internet, an attacker may eavesdrop the packet using sniffing tools, which is known as eavesdropping attack [27]. Besides, impersonation attack [28] is based on the intercepted messages. Attackers may attempt to impersonate as a legitimate user to deceive IIoT devices for retrieving sensitive sensing data [29], [30]. Also, an attacker may perform man-in-the-middle attack by intercepting messages transmitted from the sender and forging different messages to the genuine receiver. In such a condition, the attack impersonates both the user and the server, which leads to heavy data leakage. Many other attacks [31] also exist as threats to security, privacy, integrity, and availability in IIoT services.

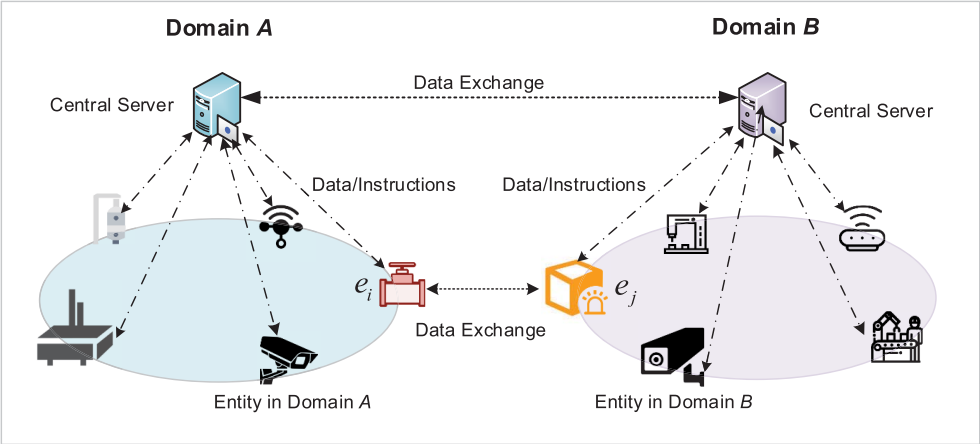


Fig. 1. A simplified cross-domain IIoT scenario with two administrative domains. These two domains usually have no subordinate relationship.

Besides, there is no administrative authority at a higher level that can bridge trust among domains. Traditionally, a trusted third party is introduced, such as CAs. However, CAs are easily compromised and prone to operational errors. Failures of CAs have been found all around the world [9]. Worse still, CAs may corrupt with some malicious attackers due to the lure of huge benefits.

Furthermore, although devices in one domain cooperate with those in other domains, these entities are very likely not to expose themselves to others as they are not in the same administrated scope and may responsible for other functionalities whose maximum privacy is needed. This identity information may affect the process of manufacturing and further damage the benefit of that factory.

C. Design Goals

1) Cross-Domain Authentication: Devices are grouped in different domains, devices in a separate domain are supervised by the central server resides in that domain. They have to authenticate each other under a situation where no trust exists among them. To construct trust between devices located in different administrative domains, the participated devices need to authenticate each other which crosses administrated domains without a trusted third party involved.

2) Identity Privacy-Preservation: Since devices may be involved in more than one task, for the secrecy concern, it is better to impede devices from privacy leakage. Thus, the identity attribute is expected to be preserved when devices are authenticated.

3) Key Negotiation: Most information will be exchanged via insecure channels such as the Internet, which are exposed to many malicious adversaries equipped with powerful computational and storage resources. However, transmitting data is not desired to eavesdrop. To this end, session keys should be successfully negotiated before data transmitting.

IV. The Proposed BASA

In this section, we introduce the proposed solution of authentication and key agreement for cross-domain IIoT devices and describe the main constructions. More design details will be given in the next section.

A. Architecture Overview

Blockchain constructs trust among different administrative domains. The cross-domain authentication mechanism is running on top of the blockchain. Specifically, consortium blockchain is exploited underneath. The underlying blockchain is used to provide a consensus service on the state binding domain to its domain-specific information, which is indispensable to devices in other domains for authentication purposes. Blockchain in this context is more likely to be a common and authentic platform for domain-specific information sharing.

1) Identity-Based Signature (IBS) and Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) Key Exchange Techniques Are Used During the Authentication and Key Agreement Process: Specifically, IBS is exploited to the authentication of devices. In IBS-based systems, when a device as a claimant requests to be authenticated by another device as a verifier, the verifier has to verify the validity of the signature generated by the claimant using the public key of the claimant based on some indispensable parameters where the domain claimant resides.

2) Hierarchical Design of the Mechanism: Several roles exist in the proposed mechanism, which includes IIoT devices, Key Generation Center (KGC), Blockhain Agent Server (BAS), and Authentication Agent Server (AAS). We group them into different layers according to their functionalities, which are illustrated in Fig. 2. IIoT devices and KGC are included on the entity layer as they are the least roles in IBC systems. BAS and AAS are two task-specific server introduced for agent missions, whose details are explained in Section IV-C. Besides, two more layers are introduced, which include the blockchain layer and storage layer. The blockchain layer can be treated as a common secure channel for domain-specific information sharing. Blockchain only stores the least information, i.e., domain identifier and its binding values consisted of a uniform resource identifier (URI) and a hash value computed upon the real domain-specific data. URI points to the actual storage file located on the Internet, where real domain-specific data are stored.

B. Entity Layer

The entity layer consists of the most components in an IBC cryptographic system, including IIoT devices and KGC.

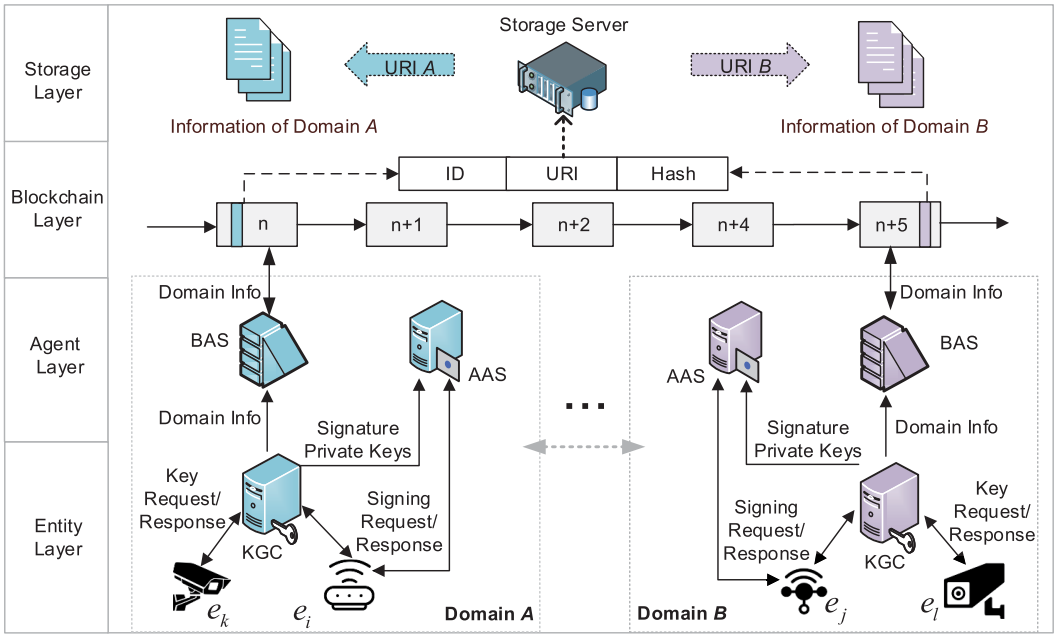


Fig. 2. Layered architecture of the proposed cross-domain authentication mechanism.

1) Key Generation Center (KGC): KGC is unique in an administrative domain that is responsible for the management of private keys of IIoT devices in that domain. Specifically, KGC generates a private key for an IIoT device based on its identity string submitted along with a request. The generated private key is then sent back to the requesting device. Besides, KGC cooperates with BAS and AAS to finish the cross-domain authentication process. It is similar to the central server mentioned in the application scenario in Section III-A

2) IIoT Devices: IIoT devices are manufacturing facilities that have sensing, processing and executing capabilities. They are responsible for a specific manufacturing task or more. These IIoT devices have to send a request to their KGC for generating their private key once they want to be authenticated.

C. Agent Layer

The agent layer contains blockchain and authentication two agent servers.

1) Blockchain Agent Server (BAS): Every KGC in an administrative domain needs to build a node to maintain the global ledger of a consortium blockchain. The consortium blockchain node encapsulates the domain-specific information into transactions and writes them into blocks. This domain-specific information will be acquired by other domains for authentication. It would be overloaded if the ledger maintaining work is afforded by KGC. Thus, it is better to split the consortium blockchain node into a separate server. The node server only receives domain-specific information from KGC and writes it into the blockchain. There is no doubt that such a node server can be treated as an agent of KGC. In addition to KGC, BAS also cooperates with AAS to complete cross-domain authentication.

2) Authentication Agent Server (AAS): The IBS technique is used to realize the authentication purpose. The authentication process can be divided into two key operations, i.e., signature generation and verification. However, these two operations are computation-consumptive. Besides, in IBC based application scenarios, devices put their trust in KGC, who should be the central server in an administrative domain and has full control of them. KGC owns the private keys of all devices of its domain. Therefore, AAS is introduced to run signature generation and verification operations on behalf of the requesting devices. These two operations can be finished under the coordination of KGC and BAS.

The functionalities that BAS and AAS servers provide can also be substituted by KGC. For the sake of clearness and eliminating the workload of KGC, they are split from KGC as a separate server.

D. Blockchain Layer

The blockchain layer here represents the consortium blockchain used underneath. It is a global distributed ledger composed of blocks encapsulating numbers of transactions, which carry domain-specific information related to different administrative domains. This information is shared by each KGC in a domain and will be used when cross-domain authentication happens. The global distributed ledger is maintained by a set of preselected nodes, each representing a KGC of an administrative domain.

1) Domain-Specific Information Formation: In fact, the domain-specific information may contain quite a lot bytes. Considering the transaction latency and throughput of blockchain, it is better to write the minimal information as little as possible. The domain-specific information written to the global ledger formats is shown in Fig. 3.

ID_{domain}*:* A unique identifier distinguishing a domain from others.

2) Uniform Resource Identifier (URI): URI is a universal naming and routing method to locate a piece of resource on the Internet, e.g., URI can be a uniform resource locator (URL).

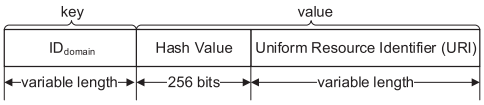


Fig. 3. Data fields indicating domain-specific information encapsulated into transactions.

TABLE II
EXPLANATION OF DOMAIN SYSTEM PARAMETERS

Field	Description
cid	curve identifier indicating the type of elliptic curve
q	parameter of elliptic curve base field
a	parameter of elliptic curve equation
b	parameter of elliptic curve equation
β	the twisted curve parameter
N	order of curve
cf	cofactor related to N
k	embedding degree of the curve
P_1	generator of additive group G_1
P_2	generator of additive group G_2
eid	bilinear pairing identifier
Ψ	homomorphism from G_2 to G_1
hid	signature private key generating function identifier

URI here points to a file hosted on a cloud service, where the details of domain-specific information are stored.

3) Hash Value: Hash value is computed on the domain-specific information file. Since the file is stored off-blockchain and hosted on the cloud service governed by a third party, it could be potentially altered by a cyber adversary or cloud service provider. The hash value on the blockchain is used to verify the authenticity of the real domain-specific information of a domain.

E. Storage Layer

Real domain-specific information is stored off-chain, which includes domain name, domain master public key, domain system parameters and a public key list of entities in that domain. Domain system parameters are illustrated in Table II. The public key of an entity is the pseudo-identity, which is discussed in Section V-A. It is needed when an entity is authenticated by others.

The real domain-specific information is stored in a single file (e.g., a *JSON* file) hosted in a cloud service, e.g., AliCloud, Microsoft Azure. To protect data from being modified by malicious adversaries, the whole file is hashed and the hash value is further written into the blockchain. Therefore, the authenticity of the data can be easily verified through the newest hash value maintained on the blockchain compared with the recomputed one upon the actual file.

V. DESIGN DETAILS

In this section, we describe the design details of the proposed authentication mechanism.

A. Identity Management Mechanism

In IBC based systems, identity acts as the public key. Two kinds of identities (public keys) of a single entity are introduced, which includes *non-anonymous identity* and *pseudo-anonymous identity*. They are exploited in different phases for different purposes. The lifetime of these two kinds of identities is decided by *identity policy* of the domain.

1) Identity Policy: The identity policy is the setting of time windows for the two kinds of identities. These time windows can be dynamically regulated to fit the actual application scenario. The identity policy is maintained and controlled only by KGC. KGC generates a signature private key for an entity based on the concatenation of identity and expire-time, which is set as the current time plus the corresponding time window.

2) Non-Anonymous Identity: When a device is deployed, it gets a unique non-anonymous identity. Non-anonymous identities are likely distributed by KGC in a regular rule for a better management. For revocation purposes, during the deployment time, KGC generates a signature private key based on the non-anonymous identity concatenated by an expire-time and stores them in the deployed device. Usually, the non-anonymous identity expire-time is long-lasting. Non-anonymous identity is used for devices to be authenticated by KGC when updating these two kinds of identities.

3) Pseudo-Anonymous Identity: A pseudo-anonymous identity is generated by the entity itself. An entity randomly generates a number and concatenates to the non-anonymous identity. Then, the combined messages are hashed into a length-fixed digest. Such a length-fixed digest is exploited as the pseudo-anonymous identity, which is similar to the Bitcoin address [32]. When an entity applies the signature private key of its pseudo-anonymous identity, it signs the pseudo-anonymous identity using its signature private key of non-anonymous identity and send a request to KGC attached with its pseudo-anonymous identity and the signature on it. KGC verifies the signature and generates a signature private key for the requester based on the concatenation of pseudo-anonymous identity and an expire-time. Usually, pseudo-anonymous identity expire-time is short-dated.

B. Authentication Mechanism

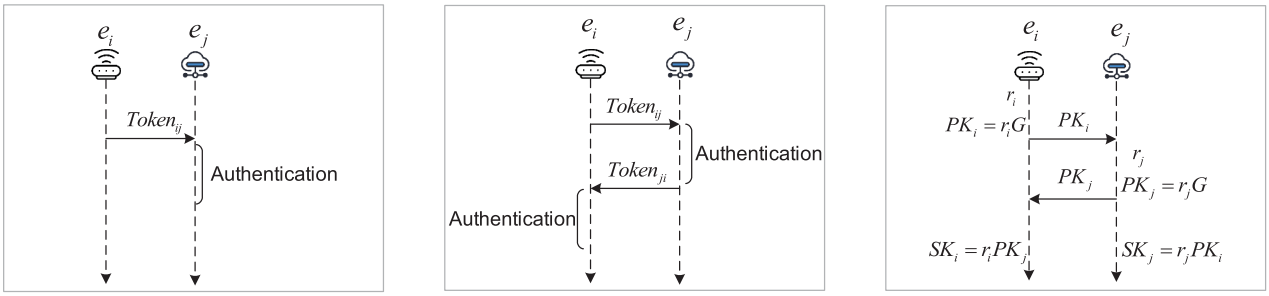
In the proposed authentication mechanism, the IBS technique is exploited. An entity to be authenticated needs to prove its claimed identity by showing its knowledge of its corresponding signature private key.

1) Unilateral Authentication: Only one pass is needed in unilateral authentication, where only one of the two communicating entities is authenticated by others. A simplified authentication mechanism is shown in Fig. 4a.

In the unilateral authentication mechanism, the authentication process is initiated by the *claimant* e_i and is authenticated by the *verifier* e_j. The form of $Token_{ij}$ is:

$$Token_{ij} = N_i||ID_i||Text||s_{sk_i}(N_i||ID_i||Text),$$

where $s_{sk_i}(X)$ means the signing on the message X using the signature private key sk_i of claimant e_i. N_i is a non-repeating random number, used to prevent valid authentication information from being accepted at a later time. $Text$ is not a necessary data field for authentication but it can be added for other purposes.



(a) Unilateral authentication model (b) Mutual authentication model (c) ECDHE key exchange model

Fig. 4. Authentication and key agreement model.

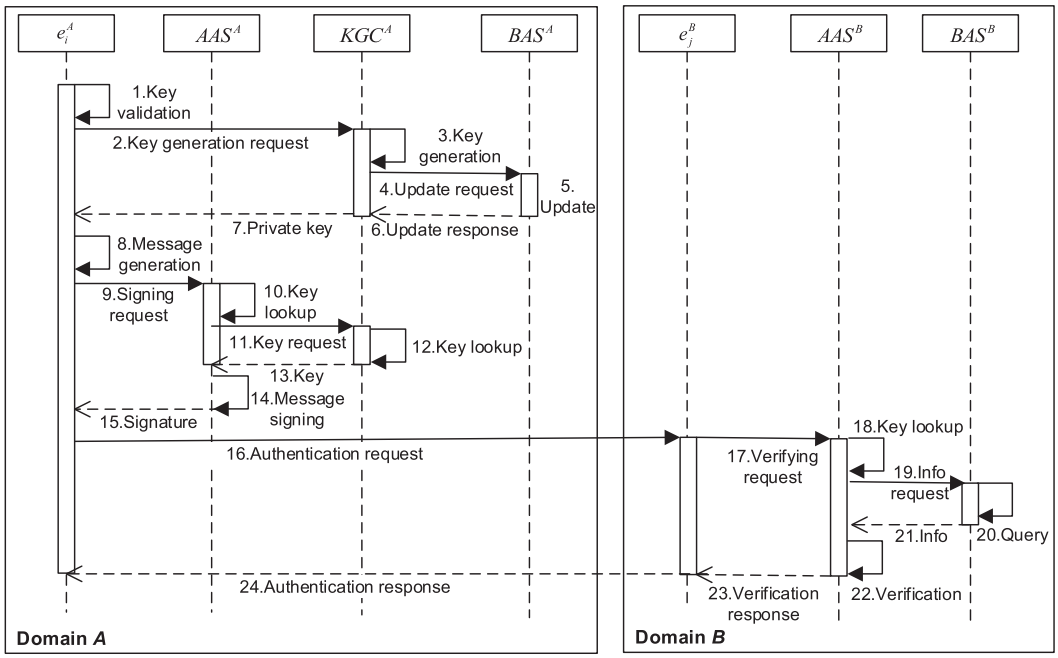


Fig. 5. Overview of cross-domain authentication process. Entity e_i^A in Domain A is authenticated by enity e_j^B in Domain B under the coordination of KGC, AAS and BAS in each domain.

Claimant e_i initializes the authentication process by sending $Token_{ij}$ to verifier e_j. Upon receiving $Token_{ij}$, verifier e_j first ensures that it posses a valid public key of claimant e_i. Then, verifier e_j verifies $Token_{ij}$ by generating signature on the unsigned message and further comparing with the received signature in $Token_{ij}$.

2) Mutual Authentication: If the communicating two entities can be mutually authenticated by each other, one more inverse pass is included, as shown in Fig. 4b. The form of $Token_{ji}$ is:

$$Token_{ji} = N_j||ID_j||Text||s_{sk_j}(N_j||ID_j||Text),$$

After e_i is authenticated by e_j, the two parties exchange their roles, which means e_i becomes the *verifier* and e_j becomes the *claimant*. e_j initializes another around authentication process, and sends $Token_{ji}$ to verifier e_i. Upon receiving $Token_{ji}$, verifier e_i first ensures that it posses a valid public key of e_j. Verifier e_i then verifies $Token_{ji}$ by generating signature on the unsigned message and further comparing with the received signature in $Token_{ji}$.

C. Cross-Domain Authentication Process

In the proposed authentication mechanism, entities are authenticated under the coordination of the three main components, i.e., KGC, AAS and BAS in each domain. The process of entity e_i^A in Domain A cross-authenticated by entity e_j^B in Domain B is illustrated in Fig. 5. Before starting the process, each domain is assumed to have been initialized, which means:

- System parameters in each domain have been instantiated, including the curve identifier cid, the parameters of the base field F_q of the elliptic curve, the parameters a and b of the elliptic curve equation, the prime N indicating the order of the curve and the cofactor cf relative to N, the embedding degree k of the curve $E(F_q)$ relative to N, the generator P_1 of the cyclic subgroup G_1 of $E(F_{q^{d_1}})$ of order N (where d_1 divides k), the generator P_2 of cyclic subgroup G_2 of $E(F_{q^{d_2}})$ of order N (where d_2 devides k), the bilinear pairing identifier eid of $e : G_1 \times G_2 \rightarrow G_T$ (of order N), and optionally the homomorphism Ψ from G_2 to G_1;

Algorithm 1 Identity-Based Signing Algorithm

Input: Signature master public key P_{pub-s} of domain, system parameters of domain, message M and e's signature private key sk_e.

Output: Signature (h, S).

1: Compute $g = e(P_1, P_{pub-s})$ in G_T;
2: Generate an random integer $r \in [1, N-1]$;
3: Compute $w = g^r$ in G_T, and convert the data type of w into a bit string;
4: Compute integer $h = H_2(M||w, N)$;
5: Compute integer $l = (r-h) \ mod \ N$; if $l = 0$, go to step 2);
6: Compute element $S = [l]sk_e$ in G_1;
7: Convert the data type of h and S to a byte string, output (h, S) as the signature on message M.

Algorithm 2 Identity-Based Verifying Algorithm

Input: Signature master public key P_{pub-s} of domain, system parameters of domain, message M', e's identity ID_e, and digital signature (h', S').

Output: Verification result: succeed or fail.

1: Convert the data type of h' to integer; if $h' \in [1, N-1]$ does not hold, the verification fails;
2: Convert the data type of S' to a point; if $S' \in G_1$ does not hold, the verification fails;
3: Compute element $g = e(P_1, P_{pub-s})$ in G_T;
4: Compute element $t = g^{h'}$ in G^T;
5: Compute integer $h_1 = H_1(ID_e||hid, N)$;
6: Compute element $P = [h_1]P_2 + P_{pub-s}$ in G_2;
7: Compute element $u = e(S', P)$ in G_T;
8: Compute element $w' = u \cdot t$ in G_T, converts the data type of w' into a bit string;
9: Compute integer $h_2 = H_2(M'||w', N)$. if $h_2 = h'$ holds, the verification succeed. Otherwise, the verification fails;

- Signature master key has been generated. The KGC randomly generates $ks \in [1, N-1]$ as the signature master private key and $P_{pub-s} = [ks]P_2$ in G_2 is computed as the signature master public key. Thus, (ks, P_{pub-s}) is the signature master key pair;
- Signature private key generating function has been selected and is identified by hid.

The authentication process is initialized by entity e_i^A of domain *A*. Entity e_i^A first validates its pseudo-anonymous identity $ID_{e_i^A}$. If $ID_{e_i^A}$ expires, e_i^A computes an new pseudo-anonymous identity $ID_{e_i^A}$. Then, e_i^A signs (using Algorithm 1) on the created pseudo-anonymous identity and applys for its signature private key from KGC^A by sending a request attached with the new created $ID_{e_i^A}$ and the signature on it. Upon receiving the request and the attached data, KGC^A verifies (using Algorithm 2) the signature. If passed, KGC^A generates signature private key $sk_{e_i^A}$ for e_i^A based on signature master private key ks^A and $ID_{e_i^A}$: KGC^A first computes $t_1 = H_1(ID_{e_i^A}||hid^A, N_A) + ks^A$ over finite field F_N^A. Then it computes $t_2 = ks^A \cdot t_1^{-1}$ and $sk_{e_i^A} = [t_2]P_1^A$. After that, KGC^A sends a request to BAS^A for updating its domain-specific information attached. BAS^A updates the content of domain *A*'s specific information file, and further writes a new record into blockchain by invoking the pre-deployed chaincode. When BAS^A receives successfully written messages from the chaincode, it notifies KGC^A for the success of information updating. KGC^A then sends back the generated $sk_{e_i^A}$ to e_i^A.

e_i^A generates a message: $M = N_{e_i^A}||ID_{e_i^A}$, where $N_{e_i^A}$ is a random number. To sign on message M, e_i^A sends a signing request to AAS^A. AAS^A looks up $sk_{e_i^A}$ and validates it. If it does not exist or has expired, AAS^A sends a request to KGC^A for applying $sk_{e_i^A}$. KGC^A looks up in its local database for $sk_{e_i^A}$ and sends $sk_{e_i^A}$ back to AAS^A. AAS^A generates signature (h, S) on message M and sends (h, S) back to e_i^A. To be authenticated, e_i^A sends an authentication request to e_j^B attached with M and (h, S).

Upon receiving message M' and signature (h', S'), e_j^B sends a verifying request to AAS^B attached with the received M' and (h', S'). AAS^B looks up in its local database for e_i^A's identity (public key). If not found or $ID_{e_i^A}$ has expired, AAS^B sends a request to BAS^B for the newest domain *A*'s specific information. BAS^B queries the newest record of domain *A* from the blockchain by invoking query chaincode and further gets the newest domain *A*'s specific information indicated by the URI field in the record. BAS^B sends the newest domain *A*'s specific information back to AAS^B, who then verifies the received signature (h', S') based on domain *A*'s specific information using Algorithm 2. AAS^B sends the verification result back to e_j^B. Based on the verification result received, e_j^B sends the authentication response back to e_i^A.

D. Key Agreement Mechanism

The key agreement mechanism proposed is intertwined with the authentication process. We use the Ephemeral Elliptic Curve based Diffie-Hellman (ECDHE) key exchange technique in the key negotiation mechanism, which can provide perfect forward security (PFS). Entities using ECDHE can compute the same session key value by sharing their public key while keeping the private key on their own. The private keys, which are vitally important to derive the session key, are never transmitted on the Internet and are further discarded when the target session key is achieved. Thus, no one can compute the same session key except for exchanged entities.

A typical ECHDE key exchange process is illustrated in Fig. 4c. The two entities share the same ecliptic curve parameters, which are the curve equation $E(Z_p)$, the curve order N, and the curve generator G. These parameters can also be pre-shared through the blockchain. e_i first randomly generates a number $r_i \in Z_p$ as its private key and computes its public key $PK_i = r_i \cdot G$. Then, e_i sends its public key PK_i to e_j. Upon receiving the public key of e_i, e_j randomly picks a number $r_j \in Z_p$ and computes its public key. Afterwards, e_j sends its public key $PK_j = r_j \cdot G$ to e_i. At this moment, e_i and e_j both have the complete data to independently compute the target session key. e_i computes the shared secret key $SK_i = r_i \cdot PK_j = r_i \cdot r_j \cdot G = SK$. e_j computes the same

secret key $SK_j = r_j \cdot PK_i = r_j \cdot r_i \cdot G = r_i \cdot r_j \cdot G = SK$. Based on the computed secret key, e_i and e_j derive the target session key $sk = H(SK||PK_i||PK_j)$.

ECDHE is vulnerable to the man-in-the-middle attack as an entity cannot determine whether the received public key is owned by the valid entity or not. To defend against man-in-the-middle attacks during the public key exchange process, the exchanged public keys are embedded in the message transmitted during the authentication process. Specifically, a shared public key is put in the $Text$ field of $Token_{ij}$ and $Token_{ji}$. Thus, the public key of each entity is signed using its authentication private key, which can be verified by the received one.

E. Security Analysis

We realize entity authentication using the IBS technique specified in Chinese SM9 [33], which is a national standard of China published by State Cryptography Administration. The security of Chinese SM9-IBS has been analyzed by Cheng [33] and is considered to be secure enough for commercial use. The domain-specific information for cross-domain authentication is shared through consortium blockchain, which is a tamper-resistant ledger under the support of consensus protocol and underlying cryptographic primitives. The key agreement mechanism exploits ECDHE, which provides PFS for communication. ECDHE is based on the elliptic curve discrete logarithm problem (ECDLP), which has been proved unsolvable by a polynomial-time algorithm at present and the computational hardness has been evaluated in [34]. Combined with IBS, ECDHE can defend against man-in-the-middle attacks. Besides, the anonymity of the IIoT device's pseudo-identity is similar to a Bitcoin address, which challenges computer forensics [32]. The privacy security of the pseudo-identity is more depending on the pseudo-identity generating hash function. The *SHA-256* hash function has been used in this paper, which has never been compromised yet.

VI. Performance Evaluation

In this section, we conduct experiments to evaluate the performance of BASA in terms of several critical metrics, including computational overhead, communication overhead, write latency, and query latency.

A. Experimental Settings

We simulate two administrative domains in the experiments. Each domain contains necessary entities, which include a KGC, an AAS, a BAS and an IIoT device. The experimental network topology is shown in Fig. 6. All machines are inter-connected in a local network. The operations of KGC, AAS in each domain are executed in a single desktop with AMD Ryzen 5 2600X CPU @3.6GHz and 16.00 GB memory. The operations of BAS are executed in a virtual machine where 4GB memory is set, hosted on the desktop using VMware Workstation 15 Pro. The virtual machine's network connection is configured in the $Bridged$ mode to be connected directly to the physical network, which stays in the same

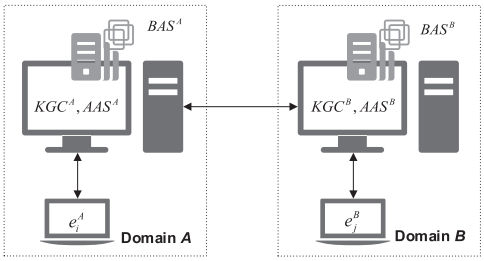


Fig. 6. An illustration of network topology in experiments.

local area network of the host machine. The operations of IIoT devices are executed in a laptop with Intel(R) Core(TM) i5-4200 CPU @1.7GHz and 4GB memory, whose computing power is similar to the widely used smartphone. Windows 10 Pro 64-bit operating system and JDK 11.02 are installed on each machine.

All the proposed layers are implemented on the application protocol layer of the protocol stack. We implement the mechanism based on the HTTP protocol, which means exchanges among different entities are encapsulated as payload in HTTP packets. It should be noted that the proposed mechanism can be easily ported to CoAP under the IoT architecture, as CoAP is similar to HTTP. More specifically, we implement the methods based on the Java JPBC 2.0.0 library, in which Type-F Pairing is utilized to realize the R-Ate pairing. The simulated two administrative domains use most of the same parameters except for the master key pair. The 256-bit Barreto-Naehrig curve, F_p-256BN, is set as $E(F_q) : y^2 = x^3 + 5$. t is an integer where $p(t) = 36t^4 + 36t^3 + 24^2 + 6t + 1$ and $N(t) = 36t^4 + 36t^3 + 18t^2 + 6t + 1$ are prime and $\#E(F_p) = N$. The embedding degree $k = 12$. p is approximately a 256-bit large prime. Let $G_1 = E(F_p)$ of order N, G_2 be the order-N subgroup of $E(F_{p^2})$, and G_T be the order-N subgroup of $E(F_{p^{12}})$. The standard $secp256r1$ elliptic curve is utilized during the ECDHE key exchange phase.

B. Computation Overhead

The computation cost is firstly evaluated through theoretical analysis on most time-consuming operations and then is evaluated through simulation experiments.

1) Theoretical Analysis: BASA is deployed in KGC, AAS, BAS and IIoT devices in each domain. They cooperate to achieve the authentication and key agreement goal. Each entity executes different cryptographic operations involved in the mechanism. We summarize the most time-consuming operations performed in BASA. To evaluate the computation overhead, we count the cryptographic operations including point addition in G_1/G_2, scalar multiplication in G_1/G_2/G_T/$secp256r1$, exponentiation in G_T and bilinear pairing, which are denoted by PA_1/PA_2, SM_1/SM_2/SM_T/SM_{r1}, Exp_T and BP respectively. The rest of operations, such as hash operation, integer addition and multiplication cost little time in our test, so they are not considered here. The numbers of time-consuming cryptographic operations are counted in Table III. It is noted that operations are not simply added up when combining authentication and

TABLE III
STATS ON TIME-CONSUMING CRYPTOGRAPHIC OPERATIONS OF BASA ON EACH ENTITY (AU:AUTHENTICATION, KN: KEY NEGOTIATION)

	AU	KN	AU + KN
e_i^A	SM_1+Exp_T	$2SM_{r1}$	SM_1+Exp_T+$2SM_{r1}$
KGC^A	SM_1+SM_2+SM_T+PA_2+Exp_T+BP	/	SM_1+SM_2+SM_T+PA_2+Exp_T+BP
AAS^A	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP
BAS^A	/	/	/
e_j^B	SM_1+Exp_T	$2SM_{r1}$	SM_1+Exp_T+$2SM_{r1}$
KGC^B	SM_1+SM_2+SM_T+PA_2+Exp_T+BP	/	SM_1+SM_2+SM_T+PA_2+Exp_T+BP
AAS^B	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP	SM_1+SM_2+SM_T+PA_2+$2Exp_T$+BP
BAS^B	/	/	/

TABLE IV
EXECUTION TIME OF BASA ON EACH ENTITY (UNIT: MILLISECOND, AU: AUTHENTICATION, KN: KEY NEGOTIATION)

	AU	KN	AU + KN
e_i^A	47.100	0.796	47.896
KGC^A	290.400	/	290.400
AAS^A	309.860	311.320	309.200
BAS^A	367.873	363.479	364.925
e_j^B	48.280	0.795	49.075
KGC^B	291.020	/	291.020
AAS^B	310.280	311.580	314.500
BAS^B	362.629	365.208	368.360

key negotiation. Two signing and verifying operations are combined into a single one as data for authentication and key negotiation can be put together for optimization. The only difference between them is that they sign on messages with different sizes.

2) Simulation Result: To record computational overhead in practice, we run BASA under the setting mentioned in Section VI-A. Table IV shows how entities in each domain undertake the computation burden, where e_i^A and e_j^B represent the IIoT devices. User (e_i^A) and server (e_j^B) only afford a small part of computation cost. KGC and AAS afford the heaviest tasks, i.e. bilinear pairing operations. It benefits from the fact that KGC intrinsically hosts signature private keys for its controlled IIoT devices. Thus, it is reasonable to set an agent for resource-constrained IIoT devices for signing or verifying tasks. The results indicate that smartphone-like devices can afford the computation burden of BASA. It is noted that the bilinear pairing $e(P_1, P_{pub-s})$ (line 1 in Algorithm 1 and line 3 in Algorithm 2) is pre-computed when the system is initialized and transmitted to IIoT devices. Under such a condition, the bilinear pairing $e(P_1, P_{pub-s})$ is only computed and stored as a constant value in IIoT devices, whose computation cost is not included in our test.

To further demonstrate the advantages of BASA in consideration of computation overhead, we compare it with the anonymous authentication and key agreement mechanism ES^3A [23], $CPAL$ [35] and $LCCH$ [36] under the same setting. Generally, after the authentication process, the session keys are immediately negotiated before sensitive data is transmitted. In the following experiments, we compare the cost time on the accumulation of authentication and key negotiation process.

We first compare computation overhead on the user-side, whose results are illustrated in Fig. 7a. As Fig. 7a shows, the computation cost of all the mechanisms grows linearly as the number of users increases since the computational overhead is approximate for each user. It is obvious that BASA outperforms the compared mechanisms. Fig. 7b shows the computational overhead on the server-side. Similar to the user-side, the execution time for BASA on the server-side is lower than the compared mechanisms.

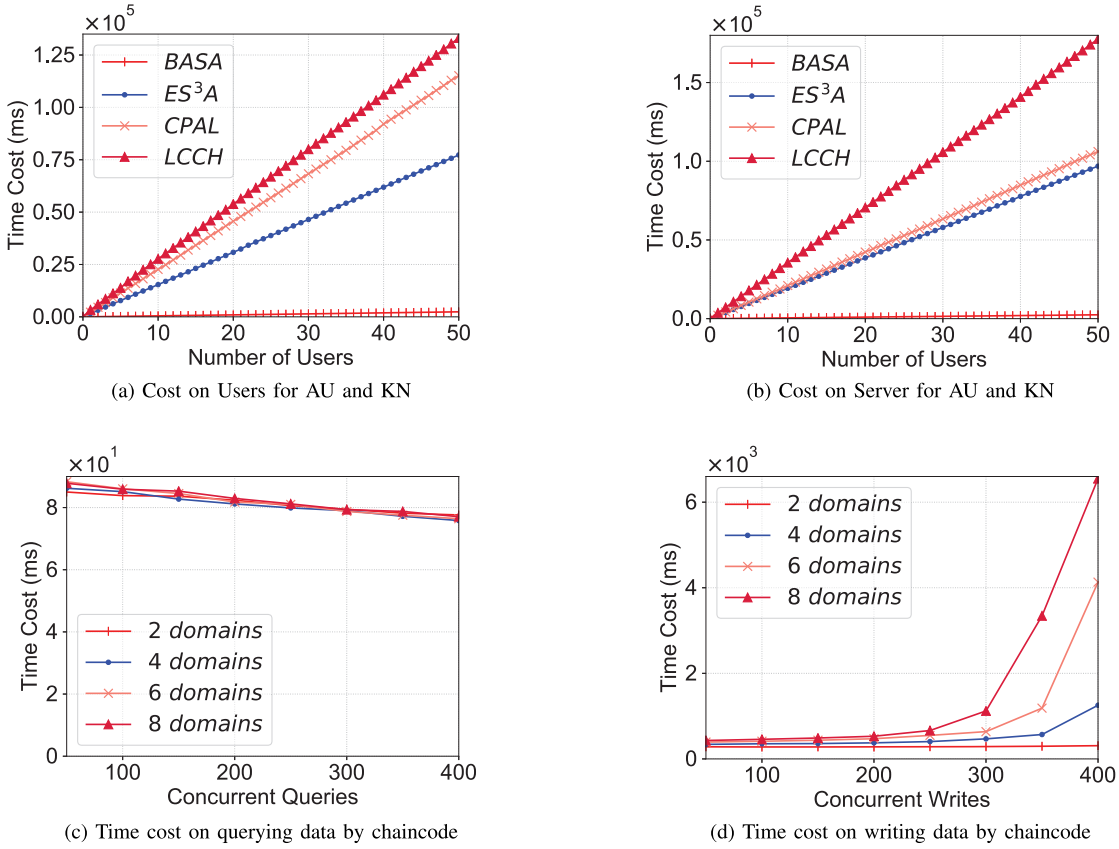
From Fig. 7a and Fig. 7b, we find that BASA costs approximate overhead on both user and server-side because it uses a digital signature-based technique to realize authentication and unilateral authentication is a symmetrical process. Thus, it is not surprising that the user and server-side consume almost the same computation power.

BASA performs superior to ES^3A [23], $CPAL$ [35] and $LCCH$ [36]. To explain the performance gap among the mechanisms, we further count the number of cryptographic operations contained in each mechanism, which are illustrated in Table V. Actually, bilinear pairing operation dominates the overall time cost. As Table V shows, the user and server-side in BASA afford no bilinear pairing operation. However, in ES^3A [23], $CPAL$ [35] and $LCCH$ [36], the user and server-side have to afford several blinear pairing operations.

C. Communication Overhead

In this section, we count the communication overhead undertaken by BASA. e_i^A sends its new generated 32-byte length pseudo-anonymous identity and the 96-byte length signature on it to KGC^A and gets its 64-byte length signature private key from KGC^A. e_i^A transmits the generated message with 96 bytes to AAS^A for signing and receives 96-byte signature. Upon receiving signature from AAS^A, e_i^A delivers the 192-byte length message and signature to e_j^B for authentication and key negotiation. To verify the received signature, e_j^B forwards the message and signature to AAS^B. To be authenticated by e_i^A and compute the same session key, e_j^B analogously performs the mirror operations of e_i^A. The communication bandwidth cost for authentication and key agreement achieves 1,536 bytes.

BASA affords more communication cost compared to ES^3A [23] and $CPAL$ [36], which are 1,336 bytes,



(a) Cost on Users for AU and KN

(b) Cost on Server for AU and KN

(c) Time cost on querying data by chaincode

(d) Time cost on writing data by chaincode

Fig. 7. Time consumption of BASA with varying parameters (AU:Authentication, KN: Key Negotiation).

TABLE V
COMPARISON OF TIME-CONSUMING CRYPTOGRAPHIC OPERATIONS IN AUTHENTICATION AND KEY NEGOTIATION

Mechanisms	User	Server
BASA	SM_1+Exp_T+$2SM_{r1}$	SM_1+Exp_T+$2SM_{r1}$
ES^3A [23]	$6Exp'_1$+$3Exp'_2$+$2Exp'_T$+$7BP'$	$10Exp'_1$+$5Exp'_2$+$4Exp'_T$+$8BP'$
$CPAL$ [35]	$18Exp''_1$+$7Exp''_T$+$7PM''_1$+$6PM''_T$+$7BP''$	$17Exp''_1$+$4Exp''_T$+$9PM''_1$+$5PM''_T$+$7BP''$
$LCCH$ [36]	$27Exp'''_1$+Exp'''_2+$9Exp'''_T$+$12PM'''_1$+$7PM'''_T$+$9BP'''$	$23Exp'''_1$+$11Exp'''_T$+$13PM'''_1$+$9PM'''_T$+$13BP'''$

[1] $*'$, $*''$ and $*'''$ indicate cryptographic operations needed in ES^3A [23], $CPAL$ [35] and LCCH [36] respectively.
[2] PM indicates point multiplication operation, and other symbols have similar meanings as defined in Section VI-B.

1,232 bytes, respectively. However, BASA is more communication-efficient than $LCCH$ [36] that costs 2016 bytes. IIoT devices in each domain delegate signing and verifying task to AAS when running BASA. During these processes, messages and signatures are forwarded between the IIoT device and AAS. It should be noted that 1,132 bytes are transmitted within an administrative domain, which is usually a local area network.

D. Write Latency and Query Latency

Consortium blockchain is introduced into BASA as a trustworthy distributed ledger for sharing domain-specific data. We record typically involved chaincode operation latency, which includes write latency and query latency. Write latency is measured as duration from the time point the write chaincode is invoked to the time point successful messages are returned. Query latency is measured as the duration from the time point the query chaincode is invoked to the time point query results are returned. Based on the *first-network* in *fabric-samples*, we extend the network from 2 domains to 8 domains. Besides, the *BatchSize* parameter value is set to 0.05s.

Fig.7c and Fig. 7d show time cost raised by chaincode. Specifically, latency happens when querying data from or writing data into the blockchain ledger. The simulated querying or writing operations are concurrent within 90m imitating a practical environment where a public key of a device is invoked. As Fig. 7c shows, time cost on querying data stays at a low level, which is about 75-90 ms. It is noted that the number of domains does not affect the query time because every time querying data from the blockchain ledger, chaincode retrieves data from the local copy of the ledger. As Fig. 7d shows, the time cost first stays at a low level no matter how many domains are included. However, as the concurrent writes

increase, time cost increases sharply. Interestingly, the more domains are included, the earlier the sharp increase point appears. It is reasonable that consensus time increases as the endorsing and validating nodes increase. It is noted that the write and query latency can be further reduced by exploiting some optimizing techniques.

VII. Discussion

BASA introduces blockchain to construct trust among untrusted domains instead of putting trust in a third party. Besides, by utilizing the consortium blockchain ledger and chaincode, we convert the public key invoking process into a blockchain ledger writing operation, which makes the public key invoking process easy in IBS systems.

There are also limitations in BASA. As mentioned in Section VI-C, the communication overhead increases as the data exchange happens frequently among the KGC, BAS and AAS servers. It is acceptable considering the significant computational overhead reduction. Besides, because of the use of blockchain, extra write and query latency of chaincode is needed. Blockchain ledger stores the currently valid public key of devices. Chaincode write operation means the invoking of the public key of devices or declaring of a new public key of devices. Chaincode query operation is the process of validating a public key, which is similar to the validation of digital certificates through the OSCP online query. The blockchain ledger is maintained by all the participant domains. Thus, the data stored in a blockchain ledger is more convincing than the OSCP source, which is only maintained by a single CA.

VIII. Conclusion

In this paper, we propose a blockchain-assisted secure authentication and key agreement mechanism BASA for cross-domain industrial IoT (IIoT). Specifically, consortium blockchain is introduced as a trusted platform for sharing domain-specific information. An identity in IBS systems can be easily revoked due to the flexible design of identity management mechanism. Further, entities in different administrative domains can authenticate each other without knowing the real identity, which can be used to protect the privacy of entities. Session key is negotiated by the key agreement mechanism, upon which the communication parties can transmit data in a secure channel. At last, the performance of BASA was evaluated to demonstrate the security and efficiency.

References

[1] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.

[2] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019.

[3] X. Du, M. Zhang, K. E. Nygard, S. Guizani, and H. H. Chen, "Self-healing sensor networks with distributed decision making," *Int. J. Sensor Netw.*, vol. 2, nos. 5–6, pp. 289–298, 2007.

[4] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 940–953, Apr. 2018.

[5] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[6] M. Shen, Y. Deng, L. Zhu, X. Du, and N. Guizani, "Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach," *IEEE Netw.*, vol. 33, no. 5, pp. 27–33, Sep. 2019.

[7] M. Shen, B. Ma, L. Zhu, X. Du, and K. Xu, "Secure phrase search for intelligent processing of encrypted data in cloud-based IoT," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1998–2008, Apr. 2019.

[8] X. Huang and X. Du, "Achieving big data privacy via hybrid cloud," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 512–517.

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

[10] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2001, pp. 213–229.

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

[12] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE J. Sel. Areas Commun.*, to be published.

[13] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Trans. Veh. Technol.*, to be published.

[14] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, *The Kerberos Network Authentication Service (V5)*, document RFC 1510, 2005.

[15] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 1996, pp. 1–15.

[16] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, document RFC 2104, 1997.

[17] 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)*, Apr. 2014, pp. 2728–2733.

[18] 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.

[19] 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, 2015.

[20] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight secure CoAP for the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3711–3720, Oct. 2013.

[21] J. Heo, C. S. Hong, M. S. Choi, S. Ho Ju, and Y. H. Lim, "Identity-based mutual device authentication schemes for PLC system," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Apr. 2008, pp. 47–51.

[22] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput.* Berlin, Germany: Springer, 2009, pp. 157–166.

[23] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 644–657, Mar. 2018.

[24] J. L. Hernandez-Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid, "Toward a lightweight authentication and authorization framework for smart objects," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 690–702, Apr. 2015.

[25] J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and access control in the Internet of Things," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2012, pp. 588–592.

[26] 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.

[27] X. Li, J. Xu, H.-N. Dai, Q. Zhao, C. F. Cheang, and Q. Wang, "On modeling eavesdropping attacks in wireless networks," *J. Comput. Sci.*, vol. 11, pp. 196–204, Nov. 2015.

[28] W.-C. Ku, "Impersonation attack on a dynamic ID-based remote user authentication scheme using smart cards," *IEICE Trans. Commun.*, vols. E88–B, no. 5, pp. 2165–2167, May 2005.

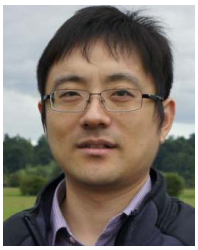
[29] X. Du, M. Shayman, and M. Rozenblit, "Implementation and performance analysis of SNMP on a TLS/TCP base," in *Proc. IEEE/IFIP Int. Symp. Integr. Netw. Manage. Integr. Netw. Manage. VII. Integr. Manage. Strategies New Millennium*, 2001, pp. 453–466.

[30] M. Shen, Y. Liu, L. Zhu, K. Xu, X. Du, and N. Guizani, "Optimizing feature selection for efficient encrypted traffic classification: A systematic approach," *IEEE Netw.*, to be published.
[31] M. J. Covington and R. Carskadden, "Threat implications of the Internet of Things," in *Proc. 5th Int. Conf. Cyber Conflict (CYCON)*, 2013, pp. 1–12.
[32] E. J. Imwinkelried and J. Luu, "The challenge of Bitcoin pseudo-anonymity to computer forensics," *Criminal Law Bull.*, 2016. Accessed: May 10, 2019. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2671921
[33] Z. Cheng, "The sm9 cryptographic schemes," in *Proc. IACR Cryptol. ePrint Arch.*, 2017, p. 117.
[34] M. Yasuda, T. Shimoyama, J. Kogure, and T. Izu, "Computational hardness of IFP and ECDLP," *Applicable Algebra Eng., Commun. Comput.*, vol. 27, no. 6, pp. 493–521, Dec. 2016.
[35] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, Feb. 2014.
[36] J. K. Liu, C.-K. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou, "Time-bound anonymous authentication for roaming networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 1, pp. 178–189, Jan. 2014.

Meng Shen (Member, IEEE) received the B.Eng. degree from Shandong University, Jinan, China, in 2009, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2014, all in computer science. He is currently an Associate Professor with the Beijing Institute of Technology, Beijing. His research interests include privacy protection for cloud and IoT, blockchain applications, and encrypted traffic classification. He received the Best Paper Runner-Up Award from the IEEE IPCCC 2014.

Huisen Liu received the B.Eng. degree in computer science from the Northwest Agriculture and Forestry University of China, China, in 2018. He is currently pursuing the master's degree with the Department of Computer Science, Beijing Institute of Technology. His research interests include blockchain applications and IoT security.

Liehuang Zhu (Member, IEEE) is currently a Professor with the Department of Computer Science, Beijing Institute of Technology. He was selected to the Program for New Century Excellent Talents in University by the Ministry of Education, China. His research interests include the Internet of Things, cloud computing security, and Internet and mobile security.



Ke Xu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He serves as a Full Professor with Tsinghua University. He has published more than 100 technical articles and holds 20 patents in the research areas of next generation Internet, P2P systems, the Internet of Things (IoT), and network virtualization and optimization. He is a member of ACM and has guest edited several special issues in IEEE and Springer journals.

Hongbo Yu received the Ph.D. degree from the School of Mathematics and System Science, Shandong University, Shandong, China. She currently serves as an Associate Professor with Tsinghua University, Beijing, China. Her research interests include cryptanalysis on hash functions and MACs, the design of dash functions and MACs, cryptanalysis on block ciphers, and the design of block cipher.

Xiaojiang Du (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland College Park in 2002 and 2003, respectively. He is a tenured Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, USA. He has authored over 400 journals and conference papers in his research areas as well as a book published by Springer. His research interests are wireless communications, wireless networks, security, and systems. He has been awarded more than U.S. $5 million research grants by the U.S. National Science Foundation (NSF), Army Research Office, Air Force, NASA, the State of Pennsylvania, and Amazon. He won the Best Paper Award from the IEEE GLOBECOM 2014 and the Best Poster Runner-Up Award from ACM MobiHoc 2014. He serves on the editorial boards of three international journals. He is a Life Member of the ACM.

Mohsen Guizani (Fellow, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering, and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Department of CSE, Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri-Kansas City, the University of Colorado Boulder, and Syracuse University. He is the author of nine books and more than 500 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grids. He is a Senior Member of the ACM. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, the chair, and the general chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and ad-hoc sensor networks. He was the Chair of the Wireless Technical Committee, IEEE Communications Society, and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is currently the Editor-in-Chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals, and the Founder and Editor-in-Chief of *Wireless Communications and Mobile Computing* journal (Wiley).