# Exploiting Blockchain and Secure Access Control Scheme to Enhance Privacy-Preserving of IoT Publish-Subscribe System

**Hongliang Tian**
Northeast Electric Power University

**Xiaonan Ge** ( ✉ gexiaonan1995@163.com )
Northeast Electric Power University https://orcid.org/0000-0003-3280-1155

**Jiayue Wang**
Northeast Electric Power University

**Chenxi Li**
Northeast Electric Power University

**Research Article**

**Keywords:** Access control, consortium, blockchain, confidentiality, data privacy, FHE, IoT, ledger, publish-subscribe system, private blockchain.

**Posted Date:** June 23rd, 2021

**DOI:** https://doi.org/10.21203/rs.3.rs-165857/v1

# Exploiting Blockchain and Secure Access Control Scheme to Enhance Privacy-Preserving of IoT Publish-Subscribe System

Hongliang Tian, Xiaonan Ge, Jiayue Wang, and Chenxi Li

**Abstract**

With the dramatically increasing deployment of intelligent devices, the Internet of Things (IoT) has attracted more attention and developed rapidly. It effectively collects and shares data from the surrounding environment to achieve better IoT services. For data sharing, the publish-subscribe (PS) paradigm provides a loosely-coupled and scalable communication model. However, due to the loosely-coupled nature, it is vulnerable to many attacks, resulting in some security threats to the IoT system, but it cannot provide the basic security mechanisms such as authentication and confidentiality to ensure the data security. Thus, in order to protect the system security and users' privacy, this paper presents a secure blockchain based privacy-preserving access control scheme for PS system, which adopt the fully homomorphic encryption (FHE) to ensure the confidentiality of the publishing events, and leverage the ledger to store the large volume of data events and access cross-domain information. Finally, we analyze the correctness and security of our scheme, moreover, we deploy our proposed prototype system on two computers, and evaluate its performance. The experimental results show that our PS system can efficiently achieve the equilibrium between the system cost and the security requirement.

**Keywords** Access control · consortium · blockchain · confidentiality · data privacy · FHE · IoT · ledger · publish-subscribe system · private blockchain.

## 1 Introduction

With the rapid development of Internet of Things (IoT) in recent years, IoT devices deployed in application scenarios such as smart grid, smart city and smart home have increased sharply [1]-[3]. It was estimated by Ericsson that there will be over 24.9 billion IoT devices connected to the Internet by 2025 [4]. These interconnected mass terminal devices store and forward data to better realize system functions. As an attractive communication paradigm, publish-subscribe (PS) system can be used to build distributed data sharing across the Internet by separating the sender from the receiver. However, due to the loose coupling between publishers and subscribers, it brings a challenge to provide security mechanisms such as authentication and confidentiality among each domain of the IoT [5]. In addition, it also puts forward more and higher security requirements for the PS system in the increasingly complex application environment of large-scale IoT. Thus, how to ensure the data is only delivered to eligible subscribers who are interested, and protect the confidentiality of the published events and the privacy of sensitive information is of great significance in the process [6],[7].

In the traditional IoT PS system, access control technology can protect the confidentiality, integrity and availability of PS service and user data [6]. However, the traditional access control schemes cannot be used directly to provide fine-grained and scalable requirements for publish-subscribe systems [8], and the original publish-subscribe model relies on a trusted third-party broker such as MQTT [9], LooCI [10], NesC [11], where data from all devices flows to subscribers through a central broker. Such a centralized architecture makes the PS model have the following disadvantages:

- The centralized architecture is vulnerable to a single point of failure. Since the broker is a centralized server, which coordinates the communication between the publishers and subscribers, if the server fails or is attacked by a malicious adversary, it may cause a large amount of sensitive information be compromised, thus threatening the privacy of the users and even making the whole system down.
- The semi-trusted broker may be immoral, it may lead to unauthorized access, abuse, and tampering with data.
- Since centralized servers rely on computationally greedy encryption algorithms, this is not suitable for computing resources-constrained IoT devices.

Therefore, a novel decentralized PS model needs to be designed to address these issues. Due to the advantages of decentralization, anonymity and non-tampering of records of blockchain [12], [13], it can provide reliable subscription record storage, subscription content forwarding and subscription information verification for the PS system. The application of blockchain in the PS system has the following benefits:

- Decentralization: The published encrypted data and the subscription records are stored in blocks in the distributed ledger, and the consistency of network records is maintained through the consensus mechanism. Due to the decentralized nature of blockchain, it can increase the fault tolerance and anti-aggression of the system, thus avoiding the impact of a single point of failure.

- Anonymity: All subscription contents are stored in the blockchain in an encrypted way, and the subscriber can access the data through its public key address. However, malicious users can only link to the public key address through hash pointer, but do not know the real identity of the users.
- Non-tampering: The subscription information is added to the blockchain after consensus verification, and then it will be recorded by all nodes together, and related to each other through cryptography, so tampering the data is very difficult and expensive.

In order to solve the mentioned challenges in the PS system, this paper designs a novel blockchain-based PS model, and on this basis, proposes an access control mechanism based on the fully homomorphic encryption (FHE) algorithm [14] to protect the privacy of data sharing among multiple domains in the IoT. The proposed model mainly includes four entities: publishers, subscribers, broker based on private blockchain, and consortium blockchain, where publisher is responsible for publishing specific encrypted data, and subscriber receives related content by subscribing to the interested topics. Each broker based on private blockchain is composed of multiple distributed and decentralized gateway devices, and it only serves a subset of IoT devices to match user needs, delivers subscription content, and stores the subscription records, whereas the consortium blockchain connects private blockchain to facilitate cross-domain data sharing.

It is noteworthy that with the dramatically increasing of mobile services and applications, the broker needs to be equipped with more computing and storage capacity, but IoT devices are usually resource-limited sensors with low computational power, and they cannot bear the resource consumption caused by complex verification calculation of blockchain, so we mitigate this problem by using edge computing. Edge computing utilizes nearby edge servers to bring real-time computations and communications [15], [16]. As one way to process data at the network edge, it greatly expands the capacity and feasibility of terminal devices. In our model, we make full use of the private blockchain that has been formed through the gateway in [17], and then use the edge servers to create the consortium blockchain and perform FHE. By this way, it can provide publishers and subscribers with effective privacy protection. Our contributions are as follows:

- We propose a blockchain based PS model for data sharing among multiple domains of IoT. This model eliminates the disadvantages of traditional PS model based on centralized broker, and can make full use of consortium blockchain to carry out cross-domain subscription services in the large-scale IoT.
- We combine edge computing to provide computing power for data validation and all cryptographic computations, and make it possible to deploy blockchain in the resource-constrained IoT. In addition, the cryptographic accumulator is used to quickly verify whether the subscription information on the one private blockchain is valid or not, which reduces the cost and latency of cross-domain data sharing.
- We use FHE with IND-CPA security to realize the attribute-based access control mechanism, so that the edge servers can perform arbitrary calculation of ciphertext without decryption. In this way, while ensuring the confidentiality and privacy of the subscription information, and realizing the fine-grained access control of user data.

The rest of this paper is organized as follows. Section II introduces some related work and briefly analyzes the pros and cons of various solutions. Section III reviews the preliminaries used in this paper. In section IV, we present a blockchain-based privacy-preserving PS model. Section V analyzes the performance and security of our scheme by deploying it on two computers. Finally, we summarize the paper with a further research discussion.

## 2 Preliminaries

In this section, we review some of the relevant theoretical basis of this study and briefly introduce and analyze the related background technologies, which mainly include the concepts of publish-subscribe system, attribute-based authorization, blockchain, fully homomorphic encryption, and edge computing.

### 2.1 Publish-Subscribe System

Publish-subscribe system can be seen as a way of data-centric message distribution [27]. During the distribution of a message, the publisher can publish the message without specifying the identity of the user, and the subscriber also does not need to know the identity of the data owner to use message. In such a middleware solution, a message is represented as an event that can be detected in the application. As Fig.1 shows, the PS model relies on three elements: publisher, subscriber, and the broker.

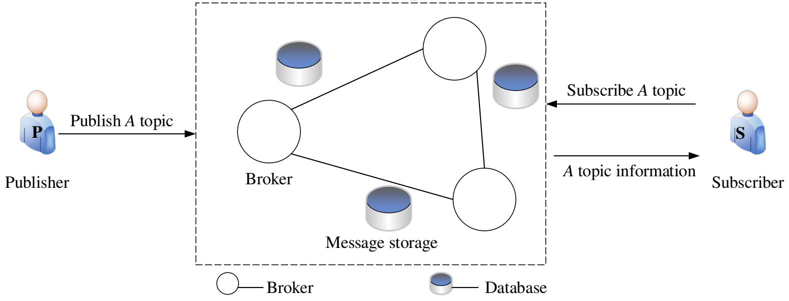

**Fig. 1** Publish-subscribe system architecture

In the model, a publisher is an actor who generates any content and publishes it to the specified topic; Subscriber is a user of events who subscribes the interested topics, and subscriber gets the published event when a publisher creates a publication for its subscription request; The broker is responsible for receiving the published events and notifying subscribers of the interested topics.

### 2.2 Attribute-based Authorization [28]

An attribute $A$ is defined as $A$= ($st$, value), meaning that the attribute $st$ with value. A user has one attribute $A$ that can be represented by conjunctive formula $A_1 \wedge A_2 \wedge\cdots\wedge A_t$. For a given system event topic $tp$, authorization policy restricts access to event data with a $tp$ topic by using a user's specific attribute value.

*Definition 1*: The expression for an authorization policy is $\Lambda_{tp} = (A_{11} \wedge A_{12} \wedge\cdots\wedge A_{1t}) \vee\cdots\vee (A_{s1} \wedge A_{s2} \wedge\cdots\wedge A_{st})$, which means that when a subscriber has at least a set of attributes from attribute concatenation $A_{11} \wedge A_{12} \wedge\cdots\wedge A_{1t}$ to $A_{s1} \wedge A_{s2} \wedge\cdots\wedge A_{st}$, the subscriber can access the data with topic $tp$.

For a subscriber whose attribute expression is $\omega = \left(A_{11}' \wedge A_{12}' \wedge\cdots\wedge A_{1t}'\right) \vee\cdots\vee \left(A_{h1}' \wedge A_{h2}' \wedge\cdots\wedge A_{ht}'\right)$, he/she has $h$ group connection attributes. As long as one of the $h$ group conjunctive attributes appears in $\Lambda_{tp}$, then $\omega$ is defined to satisfy $\Lambda_{st}$.

### 2.3 Blockchain and Edge Computing

Since Satoshi Nakamoto [12] published the Bitcoin White paper in 2008, the blockchain, as the underlying technology of Bitcoin has quickly attracted a lot of attention due to its characteristics such as decentralization, no tampering, public verification and anonymity. The blockchain works as a distributed database that records all transactions that have occurred in the peer-to-peer (P2P) network. As shown in Fig.2, the blockchain is a series of blocks connected one by one by hash. Blocks are added to the longest main blockchain by consistency protocol among most nodes in the network. Each block contains two parts: block header and block body, where all transactions involved in the block body, the block header consists of the link pointers of the previous block header, a Merkle root of all transactions and a timestamp. Hyperledger Fabric [29-30] is a consortium blockchain based on distributed ledger. Unlike public or private blockchain, it executes the verification of transactions by a set of pre-selected nodes in the consortium blockchain, and the nodes can change dynamically, so the consortium blockchain is more suitable for the scenario that supports node scalability.

Due to the limited computing capacity and available energy consumption of IoT terminal device, it has become the key bottleneck restricting the application of blockchain in IoT, but edge computing can help mitigate this problem. Edge computing transfers data processing from the remote cloud center to the edge of the network, and the computation and data storage can be dispersed to the edge of the Internet near the endpoint of things, sensors and users. It brings real-time computation and communication by leveraging nearby edge servers.

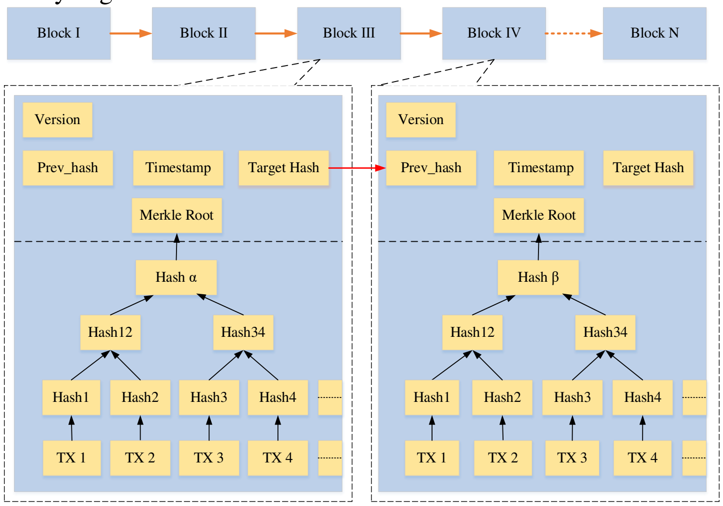

**Fig. 2** Blockchain structure

### 2.4 Fully Homomorphic Encryption [14]

Let $q$ be prime, $\mathbb{Z}_q$ be the integer field of modulo $q$, and $n$ be an integer. For the given plaintext $\upsilon \in \mathbb{Z}_q$ and the key $K$ generated by the parameters $q$ and $n$, there are encryption function $\mathrm{Enc}(K,\upsilon) = (c_1, c_2, \cdots, c_n)$ and decryption function $\mathrm{Dec}(K,(c_1, c_2, \cdots, c_n)) = \upsilon$, where ciphertext $(c_1, c_2, \cdots, c_n)$ is an $n$-dimensional vector. Public key $PK$ generated by key $K$ can be used to encrypt $\upsilon$, then:

$$\begin{gathered}\mathrm{Enc}\left(PK,\upsilon\right)=\left(c_1,c_2,\cdots,c_n\right)\\ \mathrm{Dec}\left(K,\left(c_1,c_2,\cdots,c_n\right)\right)=\upsilon\end{gathered} \tag{1}$$

Let $C=\left(c_1,c_2,\cdots c_n\right)$ and $C'=\left(c_1',c_2',\cdots,c_n'\right)$. When $\mathrm{Dec}\left(K,C\right)=\upsilon$ and $\mathrm{Dec}\left(K,C'\right)=\upsilon'$ exist in the decryption function, the FHE algorithm satisfies the following additional homomorphism properties:

$$\begin{gathered}\mathrm{Dec}\left(K,C\oplus C'\right)=\upsilon+\upsilon'\left(\mathrm{mod}\,q\right)\\ \mathrm{Dec}\left(K,d\ \square\ C\right)=d*\upsilon\left(\mathrm{mod}\,q\right)\end{gathered} \tag{2}$$

Where $\oplus$ is vector addition, and $\square$ is scalar multiplication of vectors.

The homomorphic operation of multiplication also requires the public evaluation key $\mathrm{PEK}_{ij}\left(1\le i\le n,\ 1\le j\le n\right)$, which is generated by *K*. For $\upsilon*\upsilon'$ obtained from ciphertext *C* and $C'$, it can be expressed as:

$$\left(\left(c_1*c_1'\right)\square\ \mathrm{PEK}_{11}\right)\oplus\cdots\oplus\left(\left(c_i*c_j'\right)\square\ \mathrm{PEK}_{ij}\right)\oplus\cdots\oplus\left(\left(c_n*c_n'\right)\square\ \mathrm{PEK}_{nn}\right) \tag{3}$$

For a given publisher's secret key $sk_p$ and subscriber's public key $pk_s$, the ciphertext encrypted with $sk_p$ can be converted to the ciphertext encrypted with subscriber's secret key $sk_s$. The key exchange process is as follows:

Let $\mathrm{KeySwitch}\left(pk_s,sk_p\right)$ be the generating function of exchange key *KS*, then $KS=\left\{KS_1,KS_2,\cdots,KS_n\right\}$, where any $KS_i$ is an *n*-dimensional vector. Suppose there is $\mathrm{Decrypt}\left(sk_p,\left(c_1,c_2,\cdots,c_n\right)\right)=\upsilon$, then the re-encryption of ciphertext *C* with exchange key *KS* can be expressed as $\mathrm{ReEnc}\left(KS,C\right)=\left(c_1\square KS_1\right)\oplus\left(c_2\square KS_2\right)\oplus\cdots\oplus\left(c_n\square KS_n\right)$, let $C'=\mathrm{ReEnc}\left(KS,C\right)$ ,then $\mathrm{Dec}\left(K_s,C'\right)=\upsilon$ .

## 3 Related Work

In recent years, most of the research on PS system has focused on effective event routing, event filtering, and composite event detection, and little has been done to address privacy issues. Here we briefly summarize some relevant work in recent years, and find that it can be divided into two categories: 1) PS system based on traditional broker server; 2) PS system based on P2P (peer to peer) network. This section mainly analyzes the current research status of privacy-preserving PS system.

### 3.1 Based on Traditional Broker Servers

Duan et al. [18] proposed a comprehensive access control framework CACF to guarantee the data confidentiality and service privacy of the publish-subscribe model in different domains. It uses fully homomorphic encryption to encrypt data and bi-directional privacy-preserving policy to match access policies and subscription policy. We can see from the performance analysis result that the CACF scheme can provide confidentiality and privacy-preserving with acceptable latency. But the centralized message-oriented Java Message Service (JMS) broker can cause a single point of failure.

AKPS [19] is a privacy-preserving attribute-keyword based data publish-subscribe scheme. This scheme uses attribute-based encryption with decryption outsourcing to encrypt the published data. While realizing the publisher's own control of data access, it transfers the main decryption overhead from subscribers to the cloud server. And subscribers who search by keyword can choose to receive the data according to their own interests. However, the publisher has only one identity, that is, it cannot receive the information as a subscriber.

In [20], Wang et al. proposed a privacy protection scheme for a content-based publish/subscribe system with differential privacy in a fog computing environment. It used U-Apriori algorithm to extract the collection of the first K frequent items from uncertain data sets, and then applied the exponential and Laplace mechanism to ensure differential privacy. Brokers mine the first K item sets to eventually match the appropriate publishers and subscribers. This method reduces the cost of user computation and storage, but the complex attribute matching method increases the delay of matching time, and increases with the number of users.

In order to provide basic security mechanisms for fog computing based publish-subscribe system in IoT, Diro et al. [21] proposed a secure lightweight publish-subscribe protocol based on Elliptic Curve Cryptography (ECC). It reduces the overhead of computations, storage, and communications in traditional security protocols such as SSL/TSL.

Borcea et al. [22] introduced PICADOR, a topic-based publish-subscribe system designed using proxy re-encryption. This system provides end-to-end encrypted information distribution service, and it ensures the information confidentiality between publishers and subscribers without sharing encryption and decryption keys. The system not only reduces the communication cost but also reduces the vulnerability of internal attack. However, re-encryption also brings a heavy computing burden to proxy server.

### 3.2 Based on P2P Network

Zhao et al. [23] built a fair and secure publish-subscribe system (SPS) based on blockchain. In SPS, in order to realize fair data exchange, publishers publish a topic on the blockchain and subscribers subscribe the interested topic by deposit. At the same time, the publisher and subscriber use hybrid encryption to ensure data confidentiality, and take advantage of the pseudo-anonymity of bitcoin system to ensure the identity privacy of both parties. However, because this scheme cannot provide fine-grained access control, it cannot provide users with more accurate and efficient services according to their own features.

In [24], Lv et al. propose a privacy-preserving publish/subscribe model by using the blockchain technique, which ensures the system confidentiality by employing public key encryption with equality test (PKEwET), and they solved the single point of failure and the anonymity of the participants by using the Ethereum.

Tariq et al. [25] proposed a new approach to provide authentication and confidentiality in broker-less content-based publish/subscribe system. Credentials are assigned to publishers and subscribers by adapting the pairing-based cryptography mechanisms. Because the private keys and ciphertext assigned to publishers and subscribers are marked with credentials, a particular subscriber can decrypt an event only if the credentials associated with the event match the private key. However, Tariq et al. do not consider the anonymity of subscriber.

In [26], the authors contributed Trinity, a novel distributed publish-subscribe broker with blockchain-based immutability. It distributes the published data to all brokers in the network, and stores the distributed data in an immutable ledger by using the blockchain technology. In this way, it can guarantee persistence, ordering, and immutability across trust boundaries, but the Trinity framework increases the end-to-end delay while consuming bandwidth and computation resources.

## 4 BPAC System Model

In this section, we mainly explain how the proposed blockchain-based IoT publish-subscribe system works.

### 4.1 Security Model

In our work, we assume the certificate authority (CA) that creates the public/private keys for the publisher or subscriber and assigns public parameters to the system is honest, that is, the CA follows the rules to perform computations. And the publisher who can correctly and truly publish the encrypted data is legal. All published events are stored in the global ledger maintained by the edge devices, and all data validation and publish-subscribe services processing are performed by the edge devices to reduce the workload of an IoT device. It is worth emphasizing that the storage and protection of the published events is only performed by blockchain, without intervention of any other entity. Therefore, the security of our scheme is mainly guaranteed by blockchain. In our scheme, publishers and subscribers within the domain directly interact with each other through private blockchain, and the cross-domain users connect private blockchain through consortium blockchain for temporary cross-domain information interaction. In the actual collaborative IoT services, there may have a many-to-many relationship among multiple publishers and subscribers. Here we just take one publisher and one subscriber to discuss the access control procedure in our framework. The system model is shown in Fig.1.

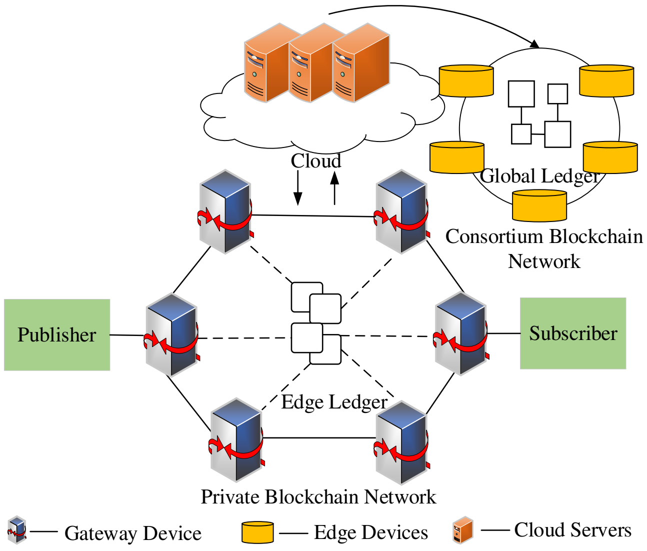

**Fig. 3** System Model

### 4.2 Blockchain-based Security Publish-subscribe System

Our proposed secure PS scheme is based on FHE scheme proposed by Brakerski et al. [14]. Here we assume a publisher $P$ has a key pair $(PK_p, SK_p)$ ,and a subscriber $S$ has a key pair $(PK_s, SK_s)$ . The specific dynamics data flow is shown in Fig.2. The access control procedure mainly contains the following phases: Setup, Publish, Subscribe, Match and Receive.

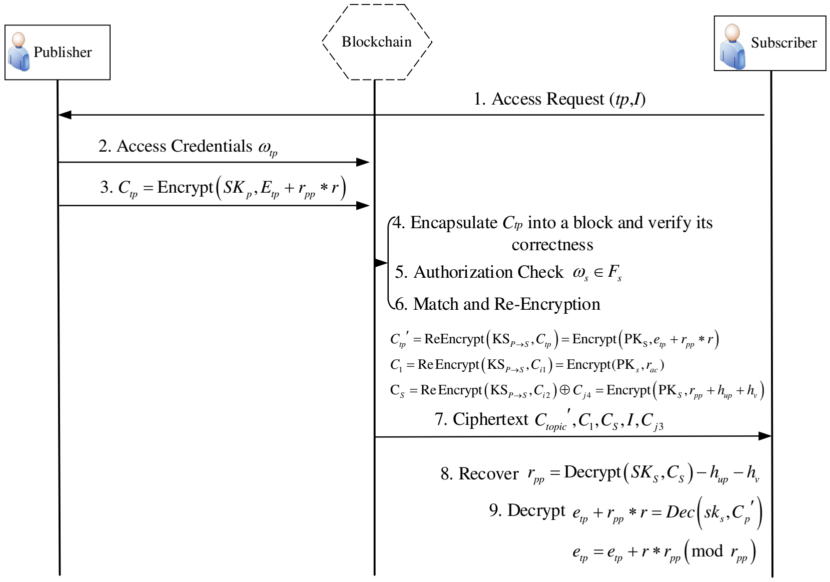
Publisher
Blockchain
Subscriber
1. Access Request (*tp*,*I*)
2. Access Credentials $\omega_{tp}$
3. $C_{tp} = \text{Encrypt}\left(SK_p, E_{tp} + r_{pp} * r\right)$
4. Encapsulate $C_{tp}$ into a block and verify its correctness
5. Authorization Check $\omega_s \in F_s$
6. Match and Re-Encryption
$C_{tp}{}' = \text{ReEncrypt}\left(\text{KS}_{P\to S}, C_{tp}\right) = \text{Encrypt}\left(\text{PK}_S, e_{tp} + r_{pp} * r\right)$
$C_1 = \text{Re Encrypt}\left(\text{KS}_{P\to S}, C_{i1}\right) = \text{Encrypt}(\text{PK}_s, r_{ac})$
$\text{C}_S = \text{Re Encrypt}\left(\text{KS}_{P\to S}, C_{i2}\right) \oplus C_{j4} = \text{Encrypt}\left(\text{PK}_S, r_{pp} + h_{up} + h_v\right)$
7. Ciphertext $C_{topic}{}', C_1, C_S, I, C_{j3}$
8. Recover $r_{pp} = \text{Decrypt}\left(SK_S, C_S\right) - h_{up} - h_v$
9. Decrypt $e_{tp} + r_{pp} * r = Dec\left(sk_s, C_p{}'\right)$
$e_{tp} = e_{tp} + r * r_{pp}\left(\text{mod } r_{pp}\right)$

**Fig. 4** Interactive Time Sequence in Our Scheme

- Setup

The setup algorithm takes the security parameter $\lambda$, a number of levels $L$ and $b \in \{0,1\}$ as input to generate the system parameter *Params* = ($q$, $d$, $n$, $N$, $\chi$). This algorithm is run by CA and only CA knows the value of *Params*, where let $\mu=\mu(\lambda, L, b)$, whose modulus is prime $q$, and $d=d(\lambda, \mu, b)$, $n=n(\lambda, \mu, b)$, $N=N(\lambda, \mu, b)$ and $\chi=\chi(\lambda, \mu, b)$. Finally, generate the key pair *PK* and *SK*:

$$\begin{gathered} SecretKeyGen\left(params\right) \to SK \\ PublicKeyGen\left(params\right) \to PK \end{gathered} \tag{4}$$

Where the key pair of publisher and subscriber are respectively $\left(PK_p, SK_p\right)$ and $\left(PK_S, SK_S\right)$.

- Publish

The publisher randomly selects random number $r_{pp}, r_{up}, r_{ac}$ and hash function $h$ in advance, where $r_{pp}$ is greater than the number of topics in the publishing event $e_{tp}$, then generates $h_{up} = h\left(\text{A}_{i1} \,||\, \text{A}_{i2} \,||\cdots||\, \text{A}_{im} \,||\, r_{up}\right)$, and encrypts event $e_{tp}$ with topic $tp$ and policy $\Lambda_{tp} = \left(\text{A}_{11}\Lambda\text{A}_{12}\Lambda\cdots\Lambda\text{A}_{1t}\right)\text{V}\left(\text{A}_{s1}\Lambda\text{A}_{s2}\Lambda\cdots\text{A}_{st}\right)$ as $C_{tp}$ through edge servers. For each set of attribute conjunction formula $\text{A}_{i1} \;\Lambda\; \text{A}_{i2} \;\Lambda\cdots\Lambda\; \text{A}_{im} \left(1 \le i \le n\right)$, the publisher generates $F_s$ through the attribute filter function $F\left(\text{A}_{i1} \;\Lambda\cdots\Lambda\; \text{A}_{im}\right)$, and uses the edge servers to convert it into access credentials:

$$\omega_{topic} = \begin{pmatrix} \text{KS}_{P\to S}, \left\{\left(C_{11}, C_{12}, F_1\right), \left(C_{21}, C_{22}, F_2\right), \cdots, \left(C_{s1}, C_{s2}, F_s\right)\right\} \\ \left\{\left(C_{13}, C_{14}\right), \left(C_{23}, C_{24}\right), \cdots, \left(C_{h3}, C_{h4}\right)\right\} \end{pmatrix} \tag{5}$$

and finally publishes $F_s$ and $C_{tp}$ on private blockchain. The encryption process for publishing events is as follows:

$$\begin{aligned} &C_{i1} = \text{Encrypt}\left(\text{SK}_P, r_{up}\right) \\ &C_{i2} = \text{Encrypt}\left(\text{SK}_P, h_{up} + r_{pp} - r_{ac}\left(h\left(A_{i1}\right) + h\left(A_{i2}\right) + \cdots + h\left(A_{im}\right)\right)\right) \\ &C_{j3} = \text{Encrypt}(PK_S, r_S) \\ &\text{C}_{j4} = \text{Encrypt}\left(PK_S, h_v + r_{ac}\left(h\left(A_{j1}{}'\right) + h\left(A_{j2}{}'\right) + \cdots + h\left(A_{jm}{}'\right)\right)\right) \end{aligned} \tag{6}$$

When the private blockchain receives the encrypted event $\text{C}_{\text{tp}}$, the edge servers packaged it into a block and stored in the edge ledger after being authenticated by the whole network.

- Subscribe

First, the subscriber *S* with property expression $\omega_S = \left(\text{A}_{11}{}'\Lambda\text{A}_{12}{}'\Lambda\cdots\text{A}_{1t}{}'\right)\text{V}\cdots\text{V}\left(\text{A}_{h1}{}'\Lambda\text{A}_{h2}{}'\Lambda\cdots\Lambda\text{A}_{ht}{}'\right)$ subscribes to an interested topic through edge ledger, then subscriber encrypts its property index value *j* to $I = \text{Encrypt}\left(PK_s, j\right)$, and finally sends it to the private blockchain broker.

- Match and Key Switching

When the publisher receives a subscription request from the subscriber, it first checks whether subscriber's attribute conjunction $\omega_s$ satisfies $\omega_s \in F_s$. If satisfying the condition, the subscriber is certified as a valid user and his subscription request is allowed.

Then the publisher will re-encrypt the ciphertext $C_{tp}, C_{i1}, C_{i2}$ through edge servers to $C_{tp}', C_1, C_s$. The conversion process is as follows:

$$\begin{aligned}
&C_{tp}' = \mathrm{ReEncrypt}\left(\mathrm{KS}_{P\to S}, C_{tp}\right) = \mathrm{Encrypt}\left(\mathrm{PK}_S, e_{tp} + r_{pp} * r\right) \\
&C_1 = \mathrm{ReEncrypt}\left(\mathrm{KS}_{P\to S}, C_{i1}\right) = \mathrm{Encrypt}(\mathrm{PK}_s, r_{ac}) \\
&\mathrm{C}_S = \mathrm{ReEncrypt}\left(\mathrm{KS}_{P\to S}, C_{i2}\right) \oplus C_{j4} = \mathrm{Encrypt}\left(\mathrm{PK}_S, r_{pp} + h_{up} + h_v\right)
\end{aligned} \tag{7}$$

Finally, the publisher authorizes the subscriber $S$ to access $C_{tp}', C_1, C_s, I$ and $C_{j3}$ from the edge ledger.

If subscriber $S$ cannot meet the access control policy, the edge servers simply refuse the subscriber's access requests.

- Receive

After subscriber $S$ receives $C_{tp}', C_1, C_s, I$ and $C_{j3}$, it first decrypts $I$ to obtain index $j$, thus obtaining the authorization attribute conjunction $\omega_j = A_{j1}' \Lambda\, A_{j2}' \Lambda \cdots \Lambda\, A_{jm}'$. Then decrypts $C_{j3}$ and $C_1$ to get the random values $r_s$ and $r_{ac}$. Next the subscriber uses hash function $h$ to restore $r_{pp}$:

$$\begin{aligned}
&h_{up} = h\left(\mathrm{A}_{j1}' \,\|\, \mathrm{A}_{j2}' \,\|\cdots\|\, \mathrm{A}_{jm}' \,\|\, r_{up}\right) \\
&h_v = h\left(\mathrm{A}_{j1}' \,\|\, \mathrm{A}_{j2}' \,\|\cdots\|\, \mathrm{A}_{jm}' \,\|\, r_S\right) \\
&r_{pp} = \mathrm{Decrypt}\left(SK_S, C_S\right) - h_{up} - h_v
\end{aligned} \tag{8}$$

Finally, the subscriber decrypts the ciphertext $C_{tp}'$ and gets $e_{tp} + r_{pp} * r$, then performs modular operation on $r_{pp}$ to recover the event $e_{tp}$.

### 4.3 Efficient Cross-domain Access and Authentication

For cross-domain PS system, because there is no direct connection among edge ledgers, and no copies of other ledgers are kept. Therefore, after obtaining the authorization information, the subscriber needs to verify whether the authorization information block belonging to another edge ledger is valid. Here we assume that one subscriber of edge ledger $EL_1$ needs to access the publishing events in $EL_2$ through the global ledger $GL$ and verifies its validity. After obtaining the authorization information block, the verification process is shown in Fig.3, and the specific process is as follows:

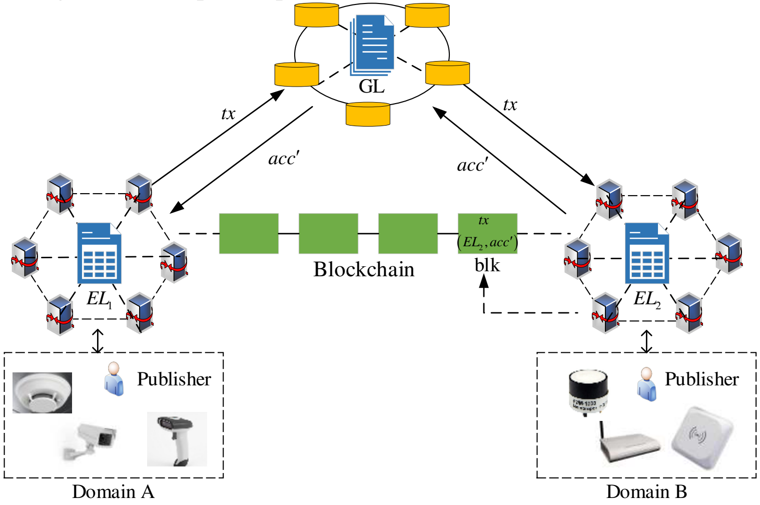

**Fig. 5** Cross-domain Data Verification

1) *EL*$_2$ processes the new authorization information block *tx*.

- *EL*$_1$ initiates a verification request for information block *tx* to the global ledger *GL*. *GL* forwards it to *EL*$_2$ and *EL*$_2$ initializes the value *acc* of the accumulator after receiving the verification request.
- *EL*$_2$ packs *tx* into a new block *blk*, and updates the accumulator value to $acc'$.
- All nodes *el*$_{2j}$ in *EL*$_2$ run the consensus protocol to add *blk*, and update accumulator value $acc'$ to the blockchain.

2) *EL*$_2$ updates its status to *GL*.

- *EL*$_2$ only updates the accumulator value to *GL* after a certain number of new blocks are created.

- *GL* checks whether *EL*2 has achieved consensus on $acc'$, if it passes the check, then the latest state of $\left(EL_2, acc'\right)$ is included in the new block.

3) *EL*1 checks the validity of *tx*

- *EL*1 obtains the current accumulator value of *EL*2 from *GL*.
- *EL*1 requests *EL*2 to provide evidence that block *blk* contains the authorization information block *tx*.
- *EL*2 responses to *EL*1's request, and provides a proof that *blk* is included in the edge ledger *EL*2.

*EL*1 verifies the evidence. After verification, it can utilize the information in *tx*.

## 5 Security and performance analysis

In this section, we first theoretically analyze the security of the proposed scheme and illustrate the correctness of our scheme, where our scheme only aims to resist collusion attack and spoofing attacks. Then we implement the prototype system to evaluate its performance.

### 5.1 Security Analysis

**Confidentiality** For our proposed publish-subscribe scheme, the security of data sharing is based on the security of blockchain and FHE algorithm. Among them, since the FHE is IND-CPA secure, that is to say, an adversary first gets a properly generated *pk*, then specifies message $m_0, m_1 \in R_M$ ( $R_M$ is a message ring) and finally gets $\mathrm{Enc}_{pk}\left(m_b\right)$ for a random number *b*, it cannot guess the value of *b* with probability $> {}^{1}\!/_{2} + \varepsilon(\lambda)$, where $\varepsilon$ is a negligible function in the security parameter $\lambda$. In other words, for a given ciphertext, an adversary is not able to know any useful information about the corresponding plaintext, that is, it is secure against chosen-plaintext attack. And we adopted FHE algorithm to set up a credible PS system for IoT, which can separate data processing rights and data ownership, so as to prevent data privacy leakage while using edge servers computing power. In addition, blockchain lies on the hardness of preventing sibyl attacks and DDoS attacks. In the large-scale IoT environments, with more IoT devices connected to the blockchain network, the more gateway nodes in the network increases, and the more security will be improved, so it is difficult for an attacker to launch a DDoS attacks in the blockchain network. This is because if you want to launch 51% attacks in the blockchain network, you need a lot of computing power to control the nodes that are distributed everywhere, an adversary is not powerful enough to take over the majority of the nodes, Therefore, the scheme can guarantee the confidentiality of the message.

**Resistance to collusion attack** For two collusive subscribers $S_1$ and $S_2$, they cannot successfully pass the inspection of the property filter function $F$ in the edge servers, because neither of them has the authentication attribute authorized by the access control policy. Even if the edge servers are malicious and also participate in the collusion attack, consequently make both pass the inspection and convert keys to generate $C_p'', C_1', C_s', I', C_{j3}'$ and $C_p''', C_1', C_s'', I'', C_{j3}''$ . However, $S_1$ and $S_2$ will only get the following ciphertext:

$$C_S' = \text{Encrypt}\left(\begin{array}{l} PK_{S_1}, r_{pp} + h_{up}' + h_v' + \\ r_{ac} * \left(\begin{array}{l} h\left(A_{k1}'\right) + h\left(A_{k2}'\right) + \cdots + h\left(A_{km}'\right) - \\ h\left(A_{i1} - A_{i2} - \cdots A_{im}\right) \end{array}\right) \end{array}\right) \tag{9}$$

$$C_S'' = \text{Encrypt}\left(\begin{array}{l} PK_{S_2}, r_{pp} + h_{up}'' + h_v'' + \\ r_{ac} * \left(\begin{array}{l} h\left(A_{q1}'\right) + h\left(A_{q2}'\right) + \cdots + h\left(A_{qm}'\right) - \\ h\left(A_{w1} - A_{w2} - \cdots - A_{wm}\right) \end{array}\right) \end{array}\right) \tag{10}$$

But since $S_1$ and $S_2$ do not know the values of $r_{ac}, A_k, A_\omega$, so $S_1$ and $S_2$ cannot recover $r_{pp}$ and the event $e_{tp}$ .

**Resistance to spoofing attacks** In our scheme, an edge server is placed in the same local network as the IoT devices, aiming to help the IoT devices perform certain kinds of computations. If the edge server is fake, it may fake the access credentials to recover event e, but it does not have any private keys of the subscribers to decrypt ciphertexts. At the same time, if an edge device tries to forge encrypted data while performing cryptographic computations, it will be detected and excluded by other nodes in the consortium blockchain. In addition, the consortium blockchain composed of edge devices has a certain fault-tolerant. Even if there are false malicious nodes in the network, as long as the number does not exceed 1/3 of the total number of nodes, it can guarantee the normal and stable operation of the system. So even if the edge devices are fake, as long as there are enough honest nodes in the network, our scheme is also available.

### 5.2 Correctness Analysis

*Theorem*: For the access control policy $\Gamma_{topic} = \left(A_{11}\Lambda A_{12}\Lambda\cdots\Lambda A_{1m}\right)V\cdots V\left(A_{n1}\Lambda A_{n2}\Lambda\cdots\Lambda A_{nm}\right)$ of an event *e* with a topic *tp,* and an attribute conjunction $\gamma = \left(A_{11}'\Lambda\ A_{12}'\Lambda\cdots\Lambda\ A_{nm}'\right)$ of a subscriber *S*, when $1 \le j \le m$ and $1 \le j \le k$ , and $A_{i1} = A_{j1}',\cdots,A_{im} = A_{jm}'$, then *S* can access all events of topic *tp.*

*Proof:* In our scheme, the edge servers generate $C_p', C_1, C_s, I$ and $C_{j3}$ for subscriber *S*, and *S* finally gets event *e* by decrypting it. When $e_{tp} + r_{pp} * r = e_{tp}\left(\bmod\ r_{pp}\right)$ , if $r_{pp} > e_{tp}$ , then theorem 1 is satisfied, so our scheme satisfies correctness.

We also compare our scheme with other related work from the aspects of confidentiality, data privacy, decentralization, fine-grained access, collusion resistance and anti-spoofing attack in Table 1, and the specific comparison results are described in Table 1.

**Table 1** The comparison with other schemes

| Scheme | Confidentiality | Decentralized | Privacy | Fine-grained access | Against collusion attack | Against spoofing attacks |
|---|---|---|---|---|---|---|
| Duan et al. [18] | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| Yang et al. [19] | ✓ | - | ✓ | ✓ | - | - |
| Wang et al. [20] | ✓ | - | ✓ | - | ✓ | - |
| Diro et al. [21] | ✓ | - | ✓ | - | - | - |
| Borcea et al. [22] | ✓ | - | ✓ | - | - | - |
| Zhao et al. [23] | ✓ | ✓ | ✓ | - | ✓ | ✓ |
| Lv et al. [24] | ✓ | ✓ | ✓ | - | - | ✓ |
| Tariq et al. [25] | ✓ | ✓ | - | - | - | - |
| Our scheme | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

As we can see from Table 1, all solutions are realized data events confidentiality, however, the proposed PS systems adopt centralized architecture in literature [18]-[22], in which all data are published to the subscriber by central broker, such a centralized architecture is vulnerable to the effects of a single point of failure, and the broker who is not fully trusted may leak or tamper with data, thus causing some insecure factors and posing a threat to the stable operation of the system. On the other hand, the data owner should have the right to determine who can use the data it provides, while in [20]-[25], there did not reflect the control of publishers over the authorization granularity for different information and subscribers. And subscribing services can be dishonest in practice, the subscribers may attempt to access unauthorized events by colluding with each other, but most of the other work did not consider this problem. On the contrary, our scheme can better solve the above problems.

### 5.3 Performance Analysis

In order to verify the availability and performance of our proposed BPAC mechanism, we deployed our prototype system on two computers, the publisher/subscriber and blockchain broker both ran on the configured with 8.0G of RAM, AMD 2.3GHz CPUs. Windows10_64 operating system, which the private blockchain is built on Ethereum. Furthermore, we use the Hyperledger Fabric deployed on the IBM Cloud platform for the consortium blockchain. Here we use system throughput and two types of time delay as the main performance evaluation criteria: 1) PS prototype system without using our proposed scheme; 2) Using the proposed blockchain-based secure PS system. Among them, the time overhead of the prototype system is from the time the subscriber initiates the subscription request until the subscriber successfully obtains the publishing service or data. Our scheme would consist the additional time spent in running BPAC. This paper evaluates the proposed scheme in terms of the different event sizes of a publish event, the number of different policies, and the number of attributes of a subscriber. Where the number of policies is 1,2,4,6,8, and the number of attribute values is 1,5,10,15,20. In addition, in order to better verify the efficiency of the proposed scheme, we compare our scheme with the CACF [18] scheme under the same test environment, which is a comprehensive access control framework using FHE scheme for publish/subscribe based IoT services communication. The specific experimental results are shown as follows. It is worth noting that all data were obtained after running 100 times.

As shown in Fig.6a, with the publishing event sizes increases, the system delay gradually increases, that is the size of the data event is one of the main factors that affect PS system latencies. Among them, the delay of the prototype system is significantly lower than our proposed scheme, and the CACF scheme is slightly higher than the prototype system but significantly lower than our scheme. This is due to the fact that the consensus validation process in our scenario consumes part of time and increase with the event complexity. Figure 6b shows the average sustainable throughput in processing the publishing events per second using different event sizes. Node that the throughput results are based on the average system latencies with or without our BPAC mechanism. As shown in figure 6b, the system throughput decreases with the growth of data event sizes, that is to say, fewer the

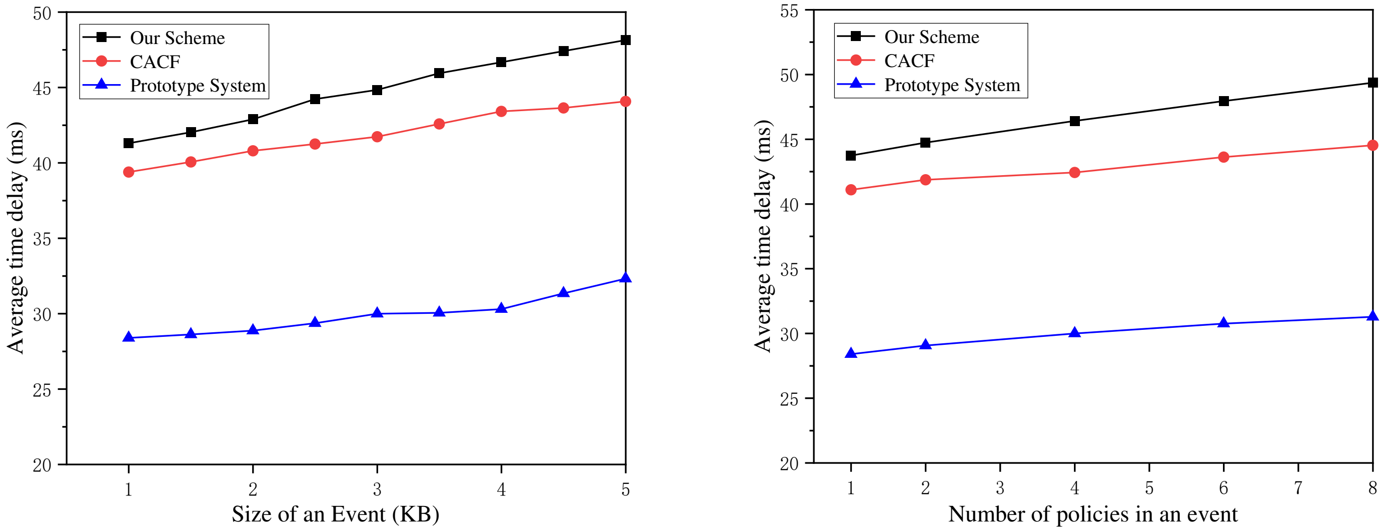

**Fig. 6** This is system delay and throughput with different event sizes: (a) Latency with different size of one event (KB); (b) Throughput for different event sizes in KB.

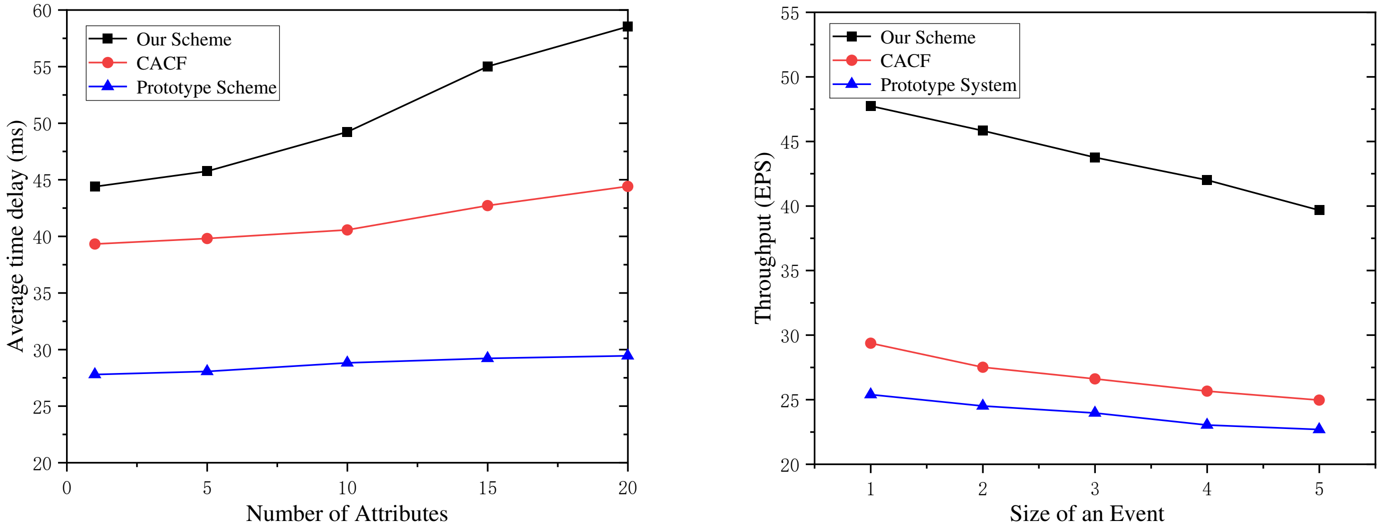

**Fig. 7** This is system delay with different number of attributes and policies: (a) Latency with different number of attributes on one subscriber; (b) Latency with different number of policies in one event.

publishing events per second can be sent from the publisher to subscriber. In addition, we can know from the above two figures that the moderate amount of event data can complete PS service with low latency and acceptable throughput.

Figure 7 shows the impact on the system time overhead from both publisher and subscriber factors, where we mainly consider how the number of policies in one publishing event and attributes in one subscriber affect PS system latencies. In figure 7a, an increase in the number of subscriber attributes will result in an increase in the system time latency. This is because an increase in the number of attributes directly lead to more time in the attribute filtering and access control policy enforcement phases. Among them, the CACF scheme is still slightly lower than the scheme we proposed, which is because the FHE algorithm used in our scheme increases the time overhead. As shown in figure 7b, with the increase of access control policies, the time delay of the system gradually increases, and the delay of our scheme is about 43~50ms. The time cost of the prototype system is significantly lower than ours, while CACF scheme is slightly higher than the prototype system but lower than our scheme. This is because our solution consumes part of the time and grows as the number of access control policies increases.

In Fig.8, in order to reflect the efficiency of cross-domain access operations, we test the throughput of our proposed PS system in different scenarios. All the experimental data is collected based on a minimum cross-domain access requirement that only involves one global ledger and two edge ledgers, and the average throughput in processing events per second is based on one KB event size. It is clear from the Fig.8 that the physical location of the nodes also affects the performance of the PS system.

As can be seen from the results discussed above, although our proposed BPAC mechanism increases the system time delay compared with the CACF scheme, the absolute value of the delay increment is not large, and the application of blockchain in PS system makes up for the lack of security and trust in the traditional scheme. We compromised the acceptable response time in exchange for higher reliability and solved the security problem in the PS system.

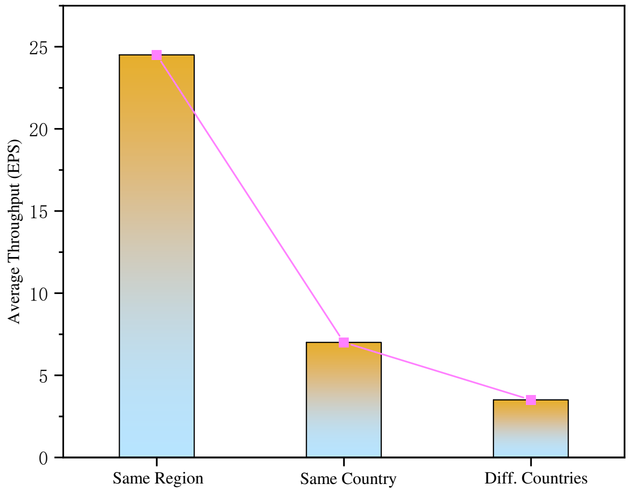

**Fig. 8** Throughput of query with nodes in different locations

## 6 Conclusion and Future Work

In this paper, we propose an access control mechanism based on blockchain and FHE algorithm, which solves the security and privacy problems in the traditional centralized PS system. Our scheme protects the confidentiality of event data by encrypting the publishing data with FHE algorithm. Meanwhile, it replaces the traditional central broker with the blockchain technology to realize decentralized distributed access control, and realizes cross-domain information interaction by storing data in the global ledger. According to the theoretical analysis, it can guarantee the security and correctness of the system, and the experimental results show that our scheme is feasible and efficient to some extent.

However our scheme also has certain deficiencies, such as our solution did not completely realize attribute revocation and update of access policies, and with the rapid growth of the IoT network scale, the attributes of one subscriber and access control policies for publishing events also become increasingly complex, it may take more time in the matching stage, so as to further prolong system response time. In future research work, we will further solve the above problems. We plan to combine the two-strategy attribute-based authorization [31] and time-limited key management to realize more fine-grained access control and efficient key revocation, and further adopt the Bloomer Filter [32] to optimize the matching process to achieve fast authentication.

## Declarations

No funding was received for the submitted work. No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed. All data and materials as well as software application or custom code used to support this study are available from the corresponding author upon reasonable request. All authors contributed to the study conception and design. Conceptualization: Hongliang Tian, Xiaonan Ge; Methodology: Xiaonan Ge; Formal analysis and investigation: Xiaonan Ge, Jiayue Wang, Chenxi Li; Writing - original draft preparation: Xiaonan Ge; Writing - review and editing: Xiaonan Ge; Supervision: Hongliang Tian.

## References

1. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal, 4*(5), 1125-1142, doi:10.1109/JIOT.2017.2683200.
2. Javed, F., Afzal, M. K., Sharif, M., & Kim, B. (2018). Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review. *IEEE Communications Surveys & Tutorials, 20*(3), 2062-2100, doi:10.1109/COMST.2018.2817685.
3. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-2376, doi:10.1109/COMST.2015.2444095.
4. A Younis, Y., Kifayat, K., & Merabti, M. (2014). An access control model for cloud computing. *Journal of Information Security and Applications, 19*, doi:10.1016/j.jisa.2014.04.003.
5. Ericsson mobility report. (2020). https://www.ericsson.com/en/internet-of-things. Accessed on June 2020.
6. Malina, L., Srivastava, G., Dzurenda, P., Hajny, J., & Fujdiak, R. (2019). *A Secure Publish/Subscribe Protocol for Internet of Things*.

7. Esposito, C., & Ciampi, M. (2015). On Security in Publish/Subscribe Services: A Survey. *IEEE Communications Surveys & Tutorials, 17*(2), 966-997, doi:10.1109/COMST.2014.2364616.
8. Uzunov, A. V. (2016). A survey of security solutions for distributed publish/subscribe systems. *Computers & Security, 61*, 94-129, doi:https://doi.org/10.1016/j.cose.2016.04.008.
9. Banks, A., Gupta, R. (2014). MQTT version 3.1.1. *OASIS Standard*.
10. Hughes, D., Thoelen, K., Horré, W., Matthys, N., del Cid, P., Michiels, S., et al. (2009). *LooCI: A loosely-coupled component infrastructure for networked embedded systems*.
11. Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., et al. (2004). *The Emergence of Networking Abstractions and Techniques in TinyOS*.
12. Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. *Cryptography Mailing list at https://metzdowd.com*.
13. Golosova, J., & Romanovs, A. The Advantages and Disadvantages of the Blockchain Technology. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), 8-10 Nov. 2018 2018* (pp. 1-6). doi:10.1109/AIEEE.2018.8592253.
14. Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2011). (Leveled) Fully Homomorphic Encryption without Bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC), 18*, 111, doi:10.1145/2090236.2090262.
15. Ren, J., Pan, Y., Goscinski, A., & Beyah, R. A. (2018). Edge Computing for the Internet of Things. *IEEE Network, 32*(1), 6-7, doi:10.1109/MNET.2018.8270624.
16. Dastjerdi, A. V., & Buyya, R. (2016). Fog Computing: Helping the Internet of Things Realize Its Potential. *Computer, 49*(8), 112-116, doi:10.1109/MC.2016.245.
17. Ge, X., Tian, H., Pan, H., Wang, J., & Li, C. (2020). Research on Distributed Blockchain-Based Privacy-Preserving and Data Security Framework in IoT. *IET Communications, 14*, doi:10.1049/iet-com.2019.0485.
18. Duan, L., Sun, C., Zhang, Y., Ni, W., & Chen, J. (2019). A Comprehensive Security Framework for Publish/Subscribe-Based IoT Services Communication. *IEEE Access, 7*, 25989-26001, doi:10.1109/ACCESS.2019.2899076.
19. Yang, K., Zhang, K., Jia, X., Hasan, M. A., & Shen, X. (2017). Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms. *Information Sciences, 387*, 116-131, doi:https://doi.org/10.1016/j.ins.2016.09.020.
20. Wang, Q., Chen, D., Zhang, N., Ding, Z., & Qin, Z. (2017). PCP: A Privacy-Preserving Content-Based Publish–Subscribe Scheme With Differential Privacy in Fog Computing. *IEEE Access, 5*, 17962-17974, doi:10.1109/ACCESS.2017.2748956.
21. Diro, A. A., Chilamkurti, N., & Kumar, N. (2017). Lightweight Cybersecurity Schemes Using Elliptic Curve Cryptography in Publish-Subscribe fog Computing. *Mobile Networks and Applications, 22*(5), 848-858, doi:10.1007/s11036-017-0851-8.
22. Borcea, C., Gupta, A. B. D., Polyakov, Y., Rohloff, K., & Ryan, G. (2017). PICADOR: End-to-end encrypted Publish–Subscribe information distribution with proxy re-encryption. *Future Generation Computer Systems, 71*, 177-191, doi:https://doi.org/10.1016/j.future.2016.10.013.
23. Zhao, Y., Li, Y., Mu, Q., Yang, B., & Yu, Y. (2018). Secure Pub-Sub: Blockchain-Based Fair Payment With Reputation for Reliable Cyber Physical Systems. *IEEE Access, 6*, 12295-12303, doi:10.1109/ACCESS.2018.2799205.
24. Lv, P., Wang, L., Zhu, H., Deng, W., & Gu, L. (2019). An IOT-Oriented Privacy-Preserving Publish/Subscribe Model Over Blockchains. *IEEE Access, 7*, 41309-41314, doi:10.1109/ACCESS.2019.2907599.
25. Tariq, M. A., Koldehofe, B., & Rothermel, K. (2014). Securing Broker-Less Publish/Subscribe Systems Using Identity-Based Encryption. *IEEE Transactions on Parallel and Distributed Systems, 25*(2), 518-528, doi:10.1109/TPDS.2013.256.
26. Ramachandran, G., Wright, K.-L., Zheng, L., Navaney, P., Naveed, M., Krishnamachari, B., et al. (2019). *Trinity: A Byzantine Fault-Tolerant Distributed Publish-Subscribe System with Immutable Blockchain-based Persistence*.
27. Eugster, P., Felber, P., Guerraoui, R., & Kermarrec, A.-M. (2003). The Many Faces of Publish/Subscribe. *ACM Comput. Surv., 35*, 114-131, doi:10.1145/857076.857078.
28. Li, J., Yao, W., Han, J., Zhang, Y., & Shen, J. (2018). User Collusion Avoidance CP-ABE With Efficient Attribute Revocation for Cloud Storage. *IEEE Systems Journal, 12*(2), 1767-1777, doi:10.1109/JSYST.2017.2667679.
29. Hyperledger Fabric. (2020). https://www.hyperledger.org/projects/fabric. Accessed on 7 July 2020.
30. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A., et al. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains.
31. Duan, L., Zhang, Y., Chen, S., Wang, S., Cheng, B., & Chen, J. (2016). Realizing IoT service's policy privacy over publish/subscribe-based middleware. *SpringerPlus, 5*(1), 1615, doi:10.1186/s40064-016-3250-x.
32. Barazzutti, R., Felber, P., Mercier, H., Onica, E., & Rivière, E. (2017). Efficient and Confidentiality-Preserving Content-Based Publish/Subscribe with Prefiltering. *IEEE Transactions on Dependable and Secure Computing, 14*(3), 308-325, doi:10.1109/TDSC.2015.2449831.