MicrothingsChain: Blockchain-based Controlled Data Sharing Platform in Multi-domain IoT

Xinghui Zhu[1], Jiawei Zheng[1], Baoquan Ren[2], Xuewen Dong[1], and Yulong Shen[1]
[1]School of Computer Science and Technology, Xidian University, Xi'an Shaanxi, 710071, China
[2]Institute of Systems Engineering, Academy of Military Science, Beijing, 100101, China

The vision of the Internet of Things (IoT) is creating an open and shared world connected to everything, making people's lives more smart, convenient, and efficient. Currently, there are various isolated IoT applications developed by different IoT manufacturers for a single specific function. The IoT nodes are classified into different administrative domains according to their affiliation, resulting in a multi-domain IoT environment. In order to realize the true IoT vision with multi-domain data sharing, this paper introduces a multi-domain IoT data sharing architecture based on blockchain to address the current phenomenon of information isolation. A control approach including capability-based cross-domain access control and risk-based access control in a domain is proposed to ensure the security of full data sharing processes. With these properties, diverse isolated applications can constitute a whole system for secure data exchange.

***Index Terms*—Blockchain, data sharing, IoT.**

I. Introduction

INTERNET of things (IoT) provides a connection to various network-enabled devices. As the development of 5G, it accelerates the interaction among devices (e.g., sensors and actuators) dedicated to and deployed for an application to fulfill common objectives. Currently, platforms such as Microthings [1], Amazon IoT [2], Azure IoT, and IBM Watson IoT have built tons of IoT applications based on their architectures.

However, these IoT applications are usually deployed in isolated vertical application architecture and designed for a specific function or only for primary usage currently. They are independent of each other, not exchanging and reusing data among them. The data is abandoned or stored in an isolated database after the main usage. We do not have an appropriate approach to combine correlated data from different applications. So, various application-based information isolations emerge, which will cause a huge cost of resources [3].

In IoT scenarios, nodes (including sensors, smart devices, etc.,) are divided into multiple administrative domains according to their ownerships [4]. Each domain collects the data by the nodes and stores them into a database. From this, many IoT applications have developed based on the data and services in different industries. The vision of the next generation of IoT is internet-of-all-thing, realizing the intelligence of IoT. So, multiply domain data sharing becomes the core and foundation. Currently, data sharing between different domains is difficult, due to various communication protocols, a mass of heterogeneous devices, and multiply data sources [5]. On the other hand, the IoT architecture in different industries is diversified, and each domain has its own data description standard. The characteristic of the multiply domain data sharing is summarized as below:

- **Decentralized:** Each administrative domain is independent of the other [6]. They are autonomous with each other. To realize data sharing between different domains, there is no trusted centralized institution to perform and control the corresponding operations [7].
- **Distributed:** Data collected by the devices in each administrative domain is distributed geographically. The data is stored in different locations, some in the cloud, and some locally [8].
- **Fine-grained:** The IoT nodes belong to different administrative domains, each of which also belongs to a different geography domain [9]. Services always include multi-domain data and applications need to collaborate cross-domain services.
- **Privacy:** The IoT data is collected from the real physical world. The data has confidentiality and privacy, which cannot be illegally accessed [10].
- **Cross-domain data stream:** During the process of multi-domain data sharing, the data stream accomplishes transfer from one domain to another domain. It is necessary to monitor stream analysis, inter-domain data conversion, and data forwarding [11].

A traditional data exchange ecosystem is based on a third-party institution acting as the bridge between data providers and consumers. Data providers need to send the data to a trusted data exchange platform that functions as a centralized data broker. However, there are concerns about the lack of trust, traceability, and security in this centralized model. Dishonest data exchange platforms may cache and tamper the data providers' data without the data owners' approval. On the other hand, the centralized data access control is vulnerable to the single point of failure/attack, where an attacker tends to target and compromise the data broker rather than multiple data owners [12][13]. The data sharing approach based on a federation of platforms needs to modify the existing IoT platforms and consider the trust problems between different participants. Thanks to the blockchain technology, transactions on the blockchain are transparent and can be traced by each

Manuscript received May 9, 2020; revised October 9, 2020. Corresponding author: Yulong Shen (email: ylshen@mail.xidian.edu.cn).

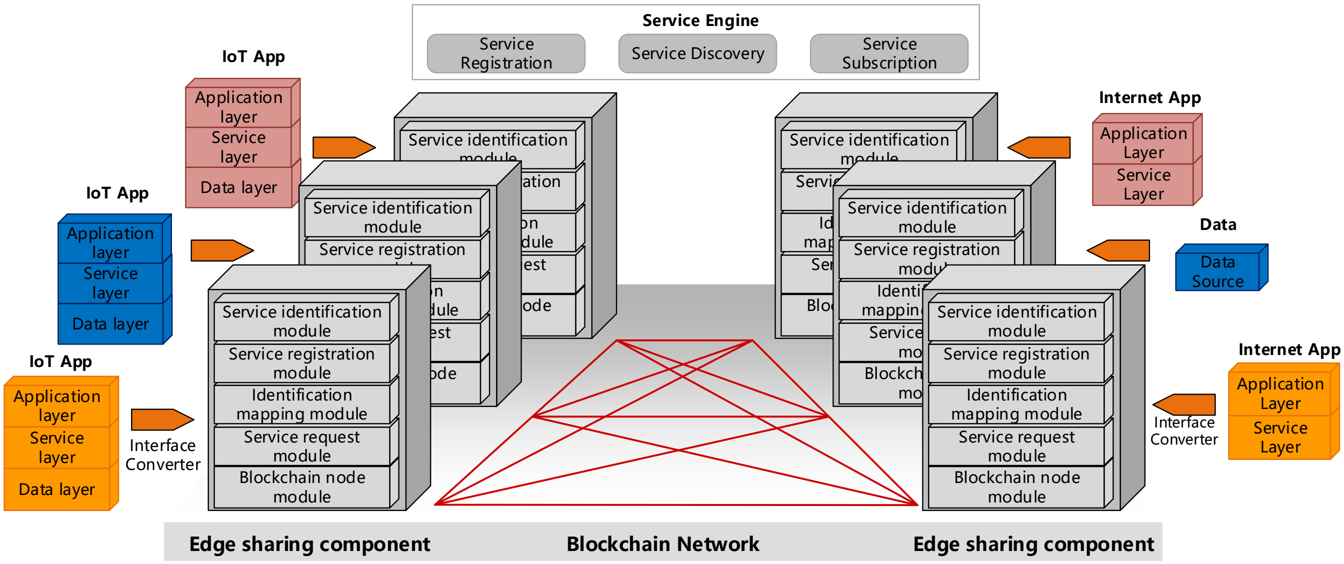


Fig. 1. Overview of the architecture

participant. The transaction process defined by smart contracts can execute automatically without man-made interfaces[14].

Blockchain is an integration of cryptography, public key infrastructure (PKI), and business models, which is used to a distributed peer-to-peer network by decentralized consensus mechanisms to achieve the data consistency. Based on different application scenarios, blockchain can be classified into three types with different characteristics: public blockchain, private blockchain, and consortium blockchain. A public blockchain is a completely decentralized blockchain infrastructure, in which all blockchain members can participate in, publish blocks and access the block's information without any restriction and verification. Compared to the public blockchain, nodes in private blockchain are registered and known in a single corporation or organization. A private blockchain is utilized in single organization solutions, where pre-delegated nodes are responsible to verify the blocks. A consortium blockchain is adapted to those business models across multiple organizations, reaching transparency and immutability among the participating parties. There is no need for processing fees and expensive computation ability to generate new blocks in consortium blockchain. While it provides lower latency in the process of transaction, it is a polycentric solution rather than an entirely decentralized solution[17].

In this article, aimed to adapt current characteristics and address issues of current data brokering infrastructures, we propose a controlled data sharing platform of multi-domain IoT based on consortium blockchain summarizing the data exchange as the service transaction. The motivation of adopting consortium blockchain is its characteristics of permission, low latency, and the organization, which can perfectly adapt to the characteristics of the multi-domain IoT environment. The following section first illustrates the current state of the art and related works. We present data sharing architecture in multi-domain IoT. Then a control approach of multi-domain data sharing is proposed to ensure the security and authorized access of data, where access control policy is executed by smart contracts, which is maintained by each participant without considering the trust problems. Then we conduct the experiment and evaluation. Our platform named $MicrothingsChain$ is described in the case of a smart city in the next section. Finally, we conclude the article.

II. Related Work

In the blockchain-based IoT domain, there has been a number of studies on developing applications to integrate blockchain with IoT. For example, Li *et al.* [8] proposed an energy trading platform based on consortium blockchain in the Industrial Internet of things. Also based on consortium blockchain, Kang *et al.* [18] presented a peer-to-peer electricity trading system among electric vehicles to improve the security of transactions. Some works are concentrated on the blockchain-based IoT architectures and protocols. Li *et al.* [19] proposed a data storage and protection system based on blockchain. Novo *et al.* [20] designed an architecture for scalable access management in IoT.

In the area of data sharing and trading, the GongXinBao (GXS) system [21], a blockchain-based data trading system, bridge the data consumers and providers through blockchain, where there is no need for a centralized third-party to perform middlemen to transmit data and values. Also, Hassija *et al.* [22] proposed a lightweight data sharing framework based on blockchain to record transactions between the vehicles and gird. K. Fan *et al.* [23] introduced a data sharing scheme in vehicular social networks to address the data security problems in the process of data sharing. They use blockchain to record access policy as well as certificating. Medical data sharing scheme [24] is to address the issue of medical data sharing in a trust-less environment based on blockchain and attribute-based encryption.

Finally, in the access control for the IoT domain, due to the centralized approaches such as ACL (access control list), RBAC (Role-based access control), and ABAC (attribute-based access control) are susceptible to a single point of failure (SPOF)issues, many literatures have concentrated on decentralized access control methods. Capability-based access control (CapBAC) methods can adapt to both centralized and distributed architecture. Hernández-Ramos *et al.*[25] proposed a capability-based access control method in distributed IoT, where the process of token's issue and verify is executed by IoT devices without a centralized authorization center. FairAccess [26] proposed to use blockchain as a decision-maker, utilizing a locking script to execute decisions. Although this is an excellent idea, the computing capability of smart contracts are not been exploited. Zhang *et al.* [27] introduced a smart contract-based access control method with access control contracts and judge contracts. This method does not consider the resource-constrained devices that are unable to act as a blockchain node and make a decision.

III. Multi-domain Data Sharing Architecture

Based on Microservice Architecture, this architecture realizes the process of data loading, adaptive, abstraction, transformation, and packaging from different domains. At the same time, we introduce the blockchain to build a secure data exchange approach to ensure legitimate access to data. As illustrated in Fig.1, the data sharing architecture consists of each application's edge sharing components connected through a blockchain network. The service engine is the management center, which is responsible for managing all the services and data from each application. The design of the edge sharing component consists of the following five modules:

Service identification module: Since each IoT application is provided by different manufactories, they all have their own service identification standards. So, a unified service identification is necessary for data sharing among different applications. In this architecture, we create a generic service identification approach. This module is responsible for service identification resolution.

Service registration module: This module is used for switching in the independent domain's data. We all know that an application is a composite of multiple services. These services are registered into data sharing architecture through this module for sharing. During the process of the service registration, service providers need to define the permission of the request. After the registration, services are classified into corresponding categories by the service identification for easy discovery. For example, temperature could be either environmental temperature or body temperature which corresponds to completely different domains, e.g., environmental temperature belongs to weather ontology while body temperature belongs to health ontology.

Identification mapping module: For the heterogeneous data from multiple sources and diverse data types form different applications, the identification mapping module is used to map the identification of other applications to ours. Due to the same data from various applications that may have the same meaning, when the data reaching the edge sharing component, the first step is to apply semantic annotations to the raw data, understanding and applying logic to the application's data. For example, let us assume that the data of three applications are related to temperature measurements. However, if they used their notations to represent temperature, e.g., ¡¯t¡¯, ¡¯temp¡¯ or ¡¯temperature¡¯, it can be understandable by humans but not by machines. Therefore, semantic annotations are important to make it understandable for the machine.

Service request module: The service request module acts as a data path, transferring the data from an application to the data exchange architecture. The main function of this module is data access and receiving. When a data consumer (e.g., an application, a device from a domain) sends a data request, the data providers' module is responsible for requesting the needed data from the device or data interface and validate the permission, if pass the validation, then the module of data consumers will receive the data. Then the data is transformed into the target type by middleware.

Blockchain node module: Blockchain node module is the core network for connecting every edge sharing component. All edge sharing component maintains the multi-domain data sharing mechanism through consensus algorithm based on blockchain collectively without the trusted third-party medium. Each blockchain node is issued a certificate, which is used to validate the identity, forming a consortium blockchain. The data sharing approach is based on the smart contract of blockchain, ensuring traceability and security.

Each application based on the different administrative domain accesses the edge sharing component of the data sharing architecture through an interface converter without other specific configuration. So, the original business of the application is unaffected. This data sharing architecture adapts to multi-domain scenarios, which enables the application to be chained together through the blockchain network to achieve the purpose of polycentric autonomy. It is noted that the blockchain node module is deployed on the edge sharing component with adequate resources compared with the constrained IoT devices. So, those IoT devices with limited computing and storage capability do not have to deal with complex decision-making matters by delegating the authority power to the edge sharing component of the administrative domain to which they belong.

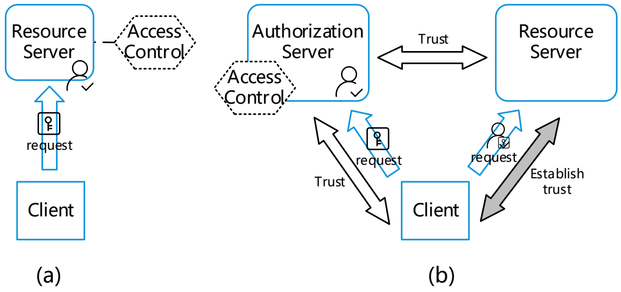


Fig. 2. Centralized (a) and Decentralized (b) access control

IV. Controlled sharing approach of multi-domain data

Based on the multi-domain data sharing architecture, we propose a control approach for multi-domain data sharing. To realize data sharing between different domains, there are demands for data control across and within domains. Based on

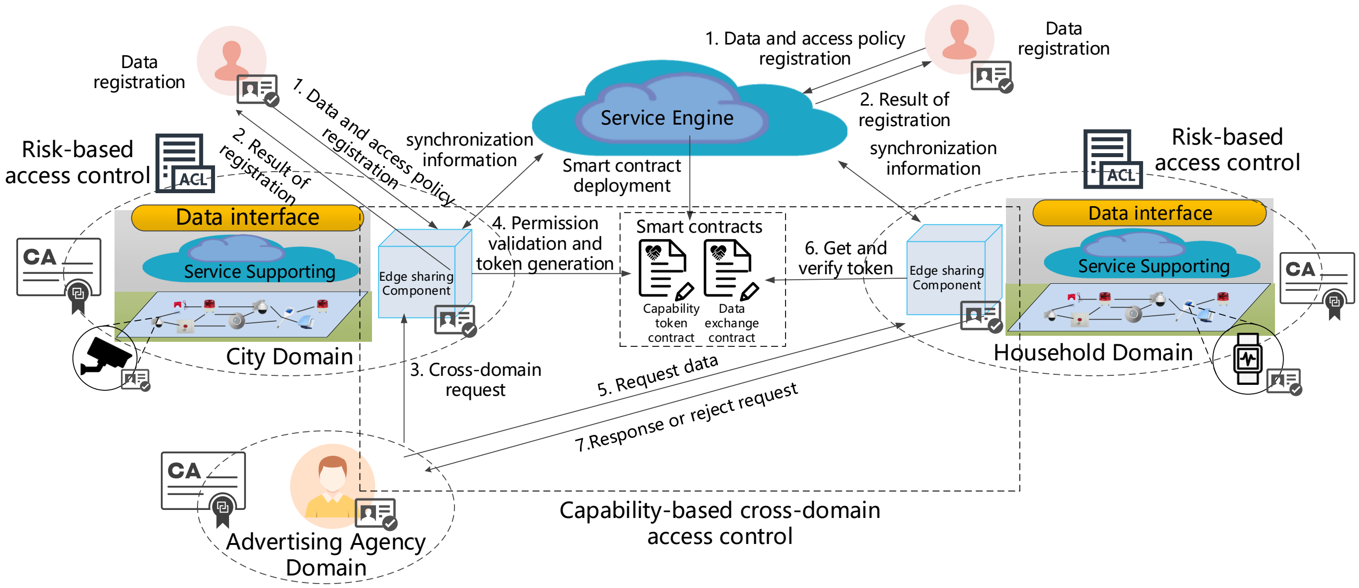

Data registration
1. Data and access policy registration
2. Result of registration
Risk-based access control
ACL
Data interface
Service Supporting
CA
City Domain
synchronization information
Service Engine
Smart contract deployment
1. Data and access policy registration
2. Result of registration
Data registration
synchronization information
Risk-based access control
ACL
Data interface
Service Supporting
CA
Household Domain
Edge sharing Component
4. Permission validation and token generation
Smart contracts
Capability token contract
Data exchange contract
6. Get and verify token
Edge sharing Component
3. Cross-domain request
5. Request data
7.Response or reject request
CA
Advertising Agency Domain
Capability-based cross-domain access control

Fig. 3. Control approach of multi-domain data sharing

reliability and security anticipate, inspired by the current IoT application system, we identify this approach into two levels.

1) Capability-based cross-domain access control.
2) Risk-based access control in the domain.

A. Capability-based cross-domain access control

Traditionally, centralized access control (as shown in Fig.2(a)) requires every IoT application or device to maintain client identities to make authorization operations. In large or super-large scale IoT scenarios this method's ability will not equal its ambition, due to its too constrained resource to store such huge credentials and data. The rapid explosion of the IoT requires high scalability to manage the increasing number of devices. Thus, access control in the IoT requires decentralized models (Fig.2(b)) that allow applying a system's authorization policies across a multitude of devices [27]. In these models, the resource server (RS) delegates authorization to an external entity, an authority in its security domain that is frequently referred to as the authorization server (AS). The RS is no longer pre-configured with the identities of legitimate clients in an access control list. Instead, a client wishing to access the RS needs to obtain authorization from the AS on-demand. When the RS receives a request, it needs to verify that this request has been previously authorized from the AS. However, the AS server is also based on a centralized architecture, which also exists a single point of failure (SPOF) and data tampered problems. Traditional OAuth/JWT authorization methods are also delegate decision-making rights to a centralized server, which also exists SPOF issues and vulnerable to be compromised to tamper the authentication policy.

The capability-based cross-domain access control delegates the authorization server to the blockchain, which is maintained by multiple parties. As shown in Fig.3, domain A and domain B use the edge sharing component deployed blockchain node module to implement cross-domain data sharing and service collaboration without establishing a trust relationship. We classify this process into four steps: Data registration, smart contract deployment, capability authorization, and authority verification.

Data registration: All entities in the data sharing architecture must be registered in the data exchange system and will be uniquely identified after successful registrations. There are two cases for registering resources. The first case is in the service engine. In this case, you need to select the domain to which the resource belongs. After the registration is successful, an identification illustrating the identity is generated for authentication. The data owner predefines the data access policy, then the blockchain network synchronizes the resource information and access control policy to each edge sharing component of the domain. After that, the system returns the registration result to the registrant. The other case is that the resource is registered in the edge sharing component. In this case, it is not necessary to select the corresponding domain. The domain where the resource registration is specified is the domain to which the resource belongs. The resource access policy is also predefined by data owners, and the policy and registration information are synchronized to other blockchain nodes, and then the registration result is returned to the registrant. The registered resource is uniquely identified by the service identification module of the edge sharing component. The edge sharing component in each domain stores the unique identification information of the domain resources and the access policy, and the service engine stores the identification information and the access policy of all the registered resources.

Smart contracts deployment: All edge sharing components and the service engine jointly maintain the capability token contract and the data exchange contract. It is noted that only the service engine has the authority to issue these contracts. The edge sharing components only can publish transactions and query smart contracts. The security of the smart contract is guaranteed by the encryption and security mechanism of the blockchain. When the data is updated, the edge sharing components synchronize this update in real-time through the blockchain interfaces using the certificate [15].

The capability token contract is responsible for generating the capability token ($CapToken$) based on their identity, which is defined by the service identification module. We consider multiple criteria including data consumers' identity, authorization relationship, authorization deep, permissions, and context factors to generate the $CapToken$ by a hash method. Take for example the scenario where the host holds the highest privileges of the door, generally, people do not have the permission to enter the room. However, when the pipes of the house break down and need to be repaired and the host is far away from the home, the privilege of opening the door needs to be authorized the repairman temporarily, which is the authorization relationship means. The context factor refers to the time, location, etc.

The data exchange contract is for maintaining the data subscribed record. The data consumers need to discover the data services they needed through the service engine and then subscribe to them. During the process of subscription, the data consumers need pay for the fee according to the predefined rules by data providers. According to the data subscribed record, the capability token contract generates the corresponding token for each entity. In this way, the data exchange contract realizes the data monetization and secure control accompanied by the capability token.

Capability authorization: When the application system of the City domain needs to access the data of the Household domain, the application system needs to launch a cross-domain request to the edge sharing component of the City domain to generate a capability token. The edge sharing component represents the identity of this domain's application, sends an authentication request to the blockchain. If the authentication is passed, the capability token is generated and the token is synchronized through the blockchain. Then the request result is returned to the application system. Otherwise, the cross-domain request fails.

Permission verification: The permission verification process occurs when an edge sharing node receives a request from another domain. When the application system sends a cross-domain data request to the edge sharing node of the domain B, through the capability authorization phase, the application system issues the capability token on the smart contract. After that, each blockchain node synchronizes this update and maintains the same capability token. So the edge sharing component of domain B can obtain the token through interacting with the smart contract by its certificate. Then it verifies the validity of the token and authenticates the request according to the resource access policy. If the result is yes, the request-response is performed. Otherwise, the cross-domain request fails.

In order to meet the scalable, distributed, and fine-grained requirements of access control of IoT solutions, the design of access control should focus on two issues, capability-based token management, and identity-based access authorization. We address the existing problems of the current centralized and decentralized access control system, utilize the decentralized, tamper-proof, and traceable features of the blockchain to guarantee that only authorized users and legal requests are allowed. And the certificates are applied to identify each entity and ensure the security of communications.

B. Risk-based access control in the domain

As mentioned above, IoT applications need service collaboration and data sharing both cross-domain and within a domain. To support a large number of users and resources in a dynamic, heterogeneous environment, the access control mechanism in the domain requires the necessary flexibility and scalability. Traditional access control has the characteristics of static, long-term, and context-insensitive, and cannot meet the dynamic changes of the IoT environment and the sensitive requirements of environmental factors. The idea of a dynamic access control system is that each access request must be dynamically analyzed in its context, considering not only the established access strategy but also the security risks [16].

The risk-based access control model performs risk analysis on the access request based on three elements including the environmental factor, the data private density, and the historical record. Each resource entity has these three attributes, which are stored in blockchain jointly maintained by edge sharing components. When the risk coefficient is less than the preset threshold, the request can be executed. Otherwise, the request is rejected. The main problem solved by this access control model is the flexibility of accessing resources. The system quantifies the risk using the risk metric defined in the risk strategy to achieve access control of the data. The risk policy is defined by the data provides.

When the edge sharing component of the domain where the resource is located receives the data request, it obtains the environmental factor, the data private density, and the historical record from the blockchain. The above factors determine the risk of the request and combine the resource predefined access policy to jointly determine the access request. We classify data requests in two situations. If you only perform a risk assessment on the access request, such as access a device's data, and do not call other services during this period, you do not need to verify the request again. We define this situation as a non-adaptive access model. If the data request calls for other services during the execution process, the permission verification needs to be executed again. Therefore, the execution behavior needs to be monitored during the entire data request process as shown in Fig. 4. We define this situation as an adaptive access model.

Compared with the non-adaptive access model, the adaptive access model not only performs risk assessment when the request is initiated but also monitors the entire request process and implements the permission determination within the entire execution cycle. Assume that an entity invokes an authorized data request, and the initial risk assessment is also passed, but other unauthorized data requests are invoked implicitly during the request execution process. If the access control is not performed, the unauthorized access to data and privacy leakage will also occur. This adaptive model is mainly for preventing authorized entities from invoking services and accessing data in unauthorized manners.

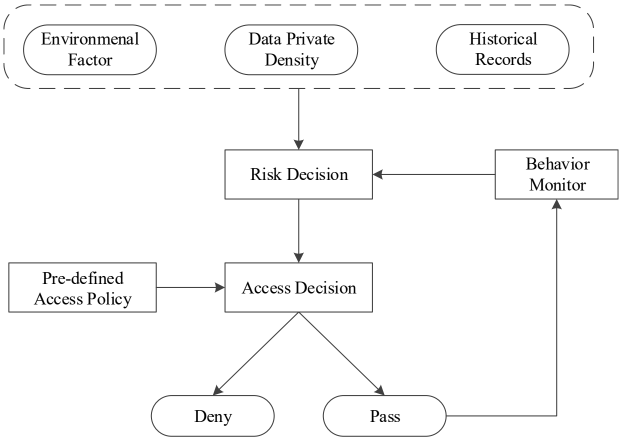


Fig. 4. The decision process of risk-based access control method

V. Experiment and Evaluation

To prove the performance of the proposed model, we conduct an experiment based on a scenario that two IoT domains exchange data with each other. The main process of the controlled data sharing approach is written in Golang and deployed it on the Hyperledger Fabric blockchain network, which is running based on Docker containers. This prototype is composed of two CA nodes, two peer nodes, and one orderer node. Each domain is authenticated by the CA node and issued a certificate for collaborating with blockchain. The peer node is responsible to invoke the transactions to the blockchain, and the orderer node is in charge of ordering the proposed transactions.

Firstly, we evaluate the performance of the capability-based cross-domain access control method. We measured the detailed processing time of this method, including the capability token generation, capability authorization, and permission verification. Measurements are given as the average over 100 test runs and compared the data with the RBAC and ABAC access control method. The result is shown in Fig. 5.

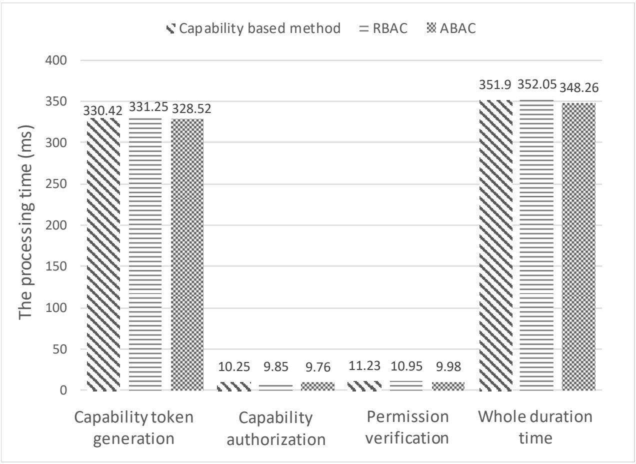


Fig. 5. The processing time (in milliseconds) of the capability-based cross-domain access control method

The total process of capability-based cross-domain access control method costs 351.90ms, which is nearly the same as the RBAC and ABAC method. The whole duration time includes the network delay, capability token generation, capability authorization, parse JSON data, and permission verification. The process of capability token generation costs the most, which accounts for 88% of the whole duration time. Permission verification includes capability token legitimacy verification and authorization verification, and the average verification time is 14.49ms, similar to the other two methods. However, the RBAC and ABAC are based on a centralized database to maintain these permission policies, which is vulnerable to be compromised. Our proposed method can ensure all the verification processes are executed by the consensual contract and the history can be recorded on immutable blockchain permanently.

Next, we investigate the feasibility of the risk-based access control model. We conduct an experiment to compare the performance of ACL and our proposed model with the increase in the number of historical records. When the number of access policies is constant, we compare the average processing time of two with the increase of history records, the result is shown in Table I.

TABLE I
Average processing time (in milliseconds) of ACL and Risk-based

History records	ACL	Risk-based
1	50	1.04
100	50	12.36
200	50	20.34
300	50	34.85
400	50	47.95
500	50	61.53
600	50	73.25

As the verification process of ACL is based on predefined access policies, the verification time is not related to the historical records. The duration is mainly due to the database retrieval time, so the processing time remains the same. When the history records below 400, the processing time of the Risk-based method is less than the number of ACL.

Finally, the concurrency performance of the model is tested by stressing testing, and the results are shown in Fig. 6.

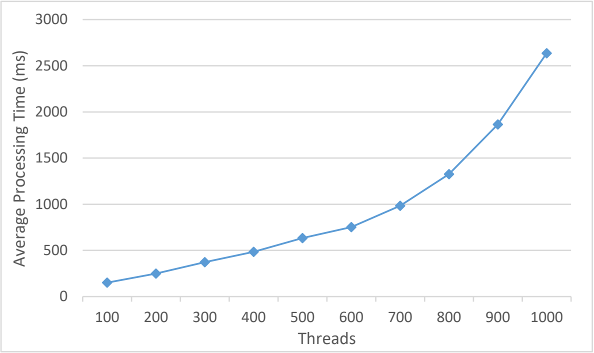


Fig. 6. The average processing time (in milliseconds) of the Risk-based method as the thread increases

We can see that when the number of threads is less than 800, the average process time is within 1000ms and shows a slow growth trend. When the number of threads is greater than 800, the number is more than 1000ms, and the growth trend is higher, indicating that the concurrency value of the model is 800 in the case of no load-balancing optimization. According to these analyse, the risk-based access control method is flexible and efficient. It can adapt to the heterogeneous IoT environment and realize dynamic access control.

VI. Case study and benefits

To further prove the characteristics of our architecture, we conduct a case study in a smart city scenario. We applied our developed platform called $MicrothingsChain$ into Dream Town in Chang'an College City, Xi'an, China, which is a future town integrating multiple information technologies. There are more than 50 isolated IoT applications here. They can exchange data across-domain through the $MicrothingsChain$, which enables a business model where data owners (who collected data by their IoT applications) are rewarded for sharing their data with data consumers, from another perspective, data consumers can better understand their customers and provide them with efficient advice.

$MicrothingsChain$ consists of four main parts: IoT App, Edge sharing component, Blockchain Network, and Service Engine. Based on our previous work, we had developed a generic IoT platform named Microthings, composed of the information aggregation environment, centralized controller, and application environment. We use Microthings as an edge sharing component, which is deployed on the edge computing server close to IoT App with Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10Ghz, 4GB RAM, and Centos 7.3 64bit operating system. The information aggregation environment of Microthings is used to standardize data formats and data delivery. The blockchain network is based on the Hyperledger Fabric and consists of Microthings with blockchain nodes deployed in Docker containers. Service registration module and service request module are implemented through chaincodes (smart contracts) including data exchange contract and capability token contract, realizing a map of physical data and assets of the blockchain, and controlling data sharing scheme. The service engine is a user interface to interact with chaincodes. The chaincodes and service engine is written in Golang and Node.js respectively.

The major user interfaces provided by the $MicrothingsChain$ are shown in Fig.4 to illustrate how a data provider and consumer participates in the $MicrothingsChain$. Fig. 7(a) shows the screen of the user center, where users can get their blockchain account and blockchain token used for transactions. The screen of the service center is shown in Fig. 7(b), in which data providers can publish and manage data services. Once data providers publish their data, they will tape into Fig. 7(c). This screen allows data providers to set description, usage, URL, price of the service, and then publish it. In Fig. 7(d), data consumers can select needed services according to the usage they are acceptable and then click the buy button to subscribe to the service. The transaction hash is displayed to indicate that the subscription is complete and you can access the data. We can also see the transaction detail in Blockchain Explorer in Fig. 7(f). We use blockchain token, capability token, and other required parameters to access the data we just subscribed to through Postman. The data of audible and visual alarm is showed in Fig. 7(e).

VII. Conclusion

This paper discusses the vision and requirements of next-generation IoT according to the characteristics of multi-domain IoT. To address the existing issues in traditional data brokering infrastructures and satisfy the future requirements, we propose a multi-domain data sharing architecture to address the current problem of data not sharing between different applications, which forms information isolation resulting in resource waste. Specifically, the architecture enables different applications with heterogeneous data to connect together through blockchain without specific configurations for data sharing. The decentralized control approach of data sharing realizes access control both cross-domain and within a domain. Each data request is required to verify by smart contract and recorded in an immutable distributed ledger for trace-back, effectively preventing unauthorized data requests and centralized data brokers from tampering data records. As a result, $MicrothingsChain$ unifies the isolated applications into a whole system, facilitates the data sharing between different domains, ensures security in the full process of data exchange. Compared to solutions based on the federation of platforms, $MicrothingsChain$ realizes the decentralized marketplace without considering the trust and consistency problems, keeping their own IoT applications unmodified. Data exchange records stored in each edge sharing component, which produces benefits for both sides, on the one hand, data providers can easily monitor the cross-domain data stream and protect the data from illegal access, on the other hand, data consumers access data from data providers directly without the consideration of middleman tampering.

Acknowledgment

This research was supported in part by the National Key Research and Development Program of China (Grant No. 2018YFE0207600 and No.2017YFB1400700), the National Natural Science Foundation of China (Grant No. U1736216, 61571352, 61602365, 61602364), Key R&D Program of Shaanxi Province (Grant No. 2019ZDLGY12-03, 2019ZDLGY13-06).

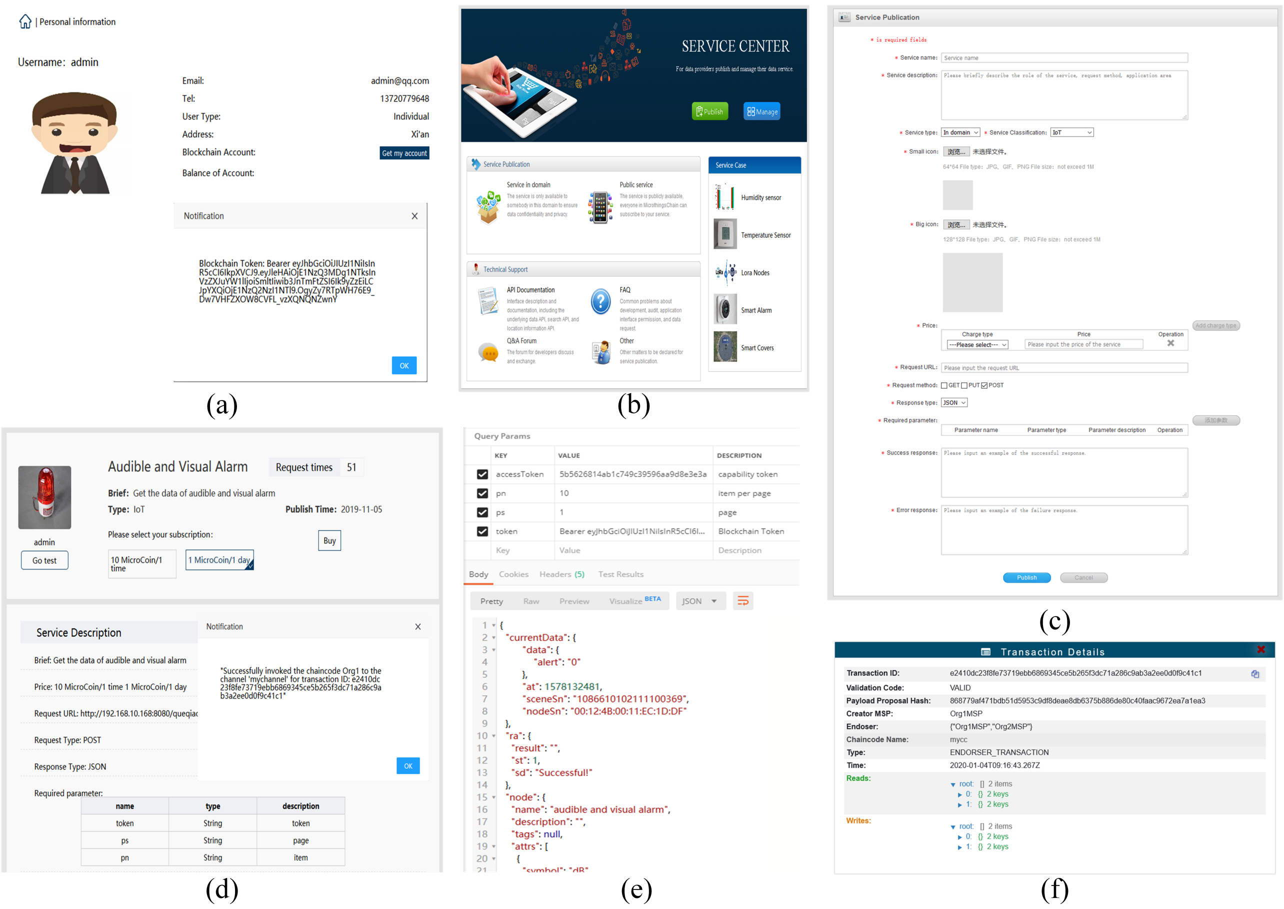
References

[1] Y. Shen, T. Zhang, Y. Wang, H. Wang, and X. Jiang, Microthings: A generic iot architecture for flexible data aggregation and scalable service cooperation, IEEE Communications Magazine, vol. 55, no. 9, pp. 86-93, 2017.

[2] S. Bhatt, F. Patwa, and R. Sandhu, Access control model for aws internet of things, International Conference on Network and System Security. Springer, 2017, pp. 721-736.

[3] C. Perera, M. Barhamgi, S. De, T. Baarslag, M. Vecchio, and K.-K. R. Choo, Designing the sensing as a service ecosystem for the internet of things, IEEE Internet of Things Magazine, vol. 1, no. 2, pp. 18-23, 2018

[4] W. Tong, X. Dong, Y. Shen, and X. Jiang, A hierarchical sharding protocol for multi-domain iot blockchains, International Conference on Communications, pp. 1-6, 2019



| Personal information

Username: admin

Email:	admin@qq.com
Tel:	13720779648
User Type:	Individual
Address:	Xi'an
Blockchain Account:	Get my account
Balance of Account:	

Notification

Blockchain Token: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NzQ3MDg1NTksInVzZXJuYW1lIjoiSmltliwib3JnTmFtZSI6Ik9yZzEiLCJpYXQiOjE1NzQ2NzI1NTl9.OgyZy7RTpWH76E9_Dw7VHFZXOW8CVFL_vzXQNQNZwnY

(a)

SERVICE CENTER

For data providers publish and manage their data service.

Service Publication

Service in domain
The service is only available to somebody in this domain to ensure data confidentiality and privacy.

Public service
The service is publicly available, everyone in MicrothingsChain can subscribe to your service.

Technical Support

API Documentation
Interface description and documentation, including the underlying data API, search API, and location information API.

FAQ
Common problems about development, audit, application interface permission, and data request.

Q&A Forum
The forum for developers discuss and exchange.

Other
Other matters to be declared for service publication.

Service Case

- Humidity sensor
- Temperature Sensor
- Lora Nodes
- Smart Alarm
- Smart Covers

(b)

Service Publication

* is required fields

* Service name: Service name

* Service description: Please briefly describe the role of the service, request method, application area

* Service type: In domain * Service Classification: IoT

* Small icon: 浏览... 未选择文件。
64*64 File type: JPG、GIF、PNG File size: not exceed 1M

* Big icon: 浏览... 未选择文件。
128*128 File type: JPG、GIF、PNG File size: not exceed 1M

* Price: Add charge type

Charge type	Price	Operation
---Please select---	Please input the price of the service	✖

* Request URL: Please input the request URL

* Request method: □ GET □ PUT ☑ POST

* Response type: JSON

* Required parameter: 添加参数

Parameter name	Parameter type	Parameter description	Operation

* Success response: Please input an example of the successful response.

* Error response: Please input an example of the failure response.

(c)

Audible and Visual Alarm Request times 51

Brief: Get the data of audible and visual alarm

Type: IoT **Publish Time:** 2019-11-05

Please select your subscription:

10 MicroCoin/1 time 1 MicroCoin/1 day

Service Description

Brief: Get the data of audible and visual alarm

Price: 10 MicroCoin/1 time 1 MicroCoin/1 day

Request URL: http://192.168.10.168:8080/queqiao

Request Type: POST

Response Type: JSON

Required parameter:

name	type	description
token	String	token
ps	String	page
pn	String	item

Notification

"Successfully invoked the chaincode Org1 to the channel 'mychannel' for transaction ID: e2410dc23f8fe73719ebb6869345ce5b265f3dc71a286c9ab3a2ee0d0f9c41c1"

(d)

Query Params

	KEY	VALUE	DESCRIPTION
☑	accessToken	5b5626814ab1c749c39596aa9d8e3e3a	capability token
☑	pn	10	item per page
☑	ps	1	page
☑	token	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I...	Blockchain Token
	Key	Value	Description

```
{
  "currentData": {
      "data": {
         "alert": "0"
      },
      "at": 1578132481,
      "sceneSn": "1086610102111100369",
      "nodeSn": "00:12:4B:00:11:EC:1D:DF"
  },
  "ra": {
    "result": "",
    "st": 1,
    "sd": "Successful!"
  },
  "node": {
    "name": "audible and visual alarm",
    "description": "",
    "tags": null,
    "attrs": [
     {
       "symbol": "dB"
```

(e)

Transaction Details

Transaction ID:	e2410dc23f8fe73719ebb6869345ce5b265f3dc71a286c9ab3a2ee0d0f9c41c1
Validation Code:	VALID
Payload Proposal Hash:	868779af471bdb51d5953c9df8deae8db6375b886de80c40faac9672ea7a1ea3
Creator MSP:	Org1MSP
Endoser:	{"Org1MSP","Org2MSP"}
Chaincode Name:	mycc
Type:	ENDORSER_TRANSACTION
Time:	2020-01-04T09:16:43.267Z
Reads:	▼ root: [] 2 items ▶ 0: {} 2 keys ▶ 1: {} 2 keys
Writes:	▼ root: [] 2 items ▶ 0: {} 2 keys ▶ 1: {} 2 keys

(f)

Fig. 7. User interfaces of MicrothingsChain

[5] J. Zheng, X. Dong, T. Zhang, J. Chen, W. Tong, and X. Yang, Microthingschain: Edge computing and decentralized iot architecture based on blockchain for cross-domain data shareing, 2018 International Conference on Networking and Network Applications (NaNA). IEEE, 2018, pp. 350-355.

[6] M. Sun and W. P. Tay, On the relationship between inference and data privacy in decentralized iot networks, IEEE Transactions on Information Forensics and Security, 2019.

[7] M. Ali, M. Vecchio, and F. Antonelli, Enabling a blockchain-based iot edge, IEEE Internet of Things Magazine, vol. 1, no. 2, pp. 24-29, December 2018.

[8] R. Li, H. Asaeda, and J. Li, A distributed publisher-driven secure data sharing scheme for information-centric iot, IEEE Internet of Things Journal, vol. 4, no. 3, pp. 791-803, 2017.

[9] Q. Huang, N. Li, and Y. Yang, Dacsc: Dynamic and fine-grained access control for secure data collaboration in cloud computing, 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1-7.

[10] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. Rodrigues, Privacy preserving data aggregation scheme for mobile edge computing assisted iot applications, IEEE Internet of Things Journal, 2018.

[11] J. Zheng, X. Dong, W. Tong, Q. Liu, and X. Zhu, Blockchain-based secure digital asset exchange scheme with qos-aware incentive mechanism, 2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR). IEEE, 2019, pp. 1-6.

[12] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, Sdte: A secure blockchain-based data trading ecosystem, IEEE Transactions on Information Forensics and Security, vol. 15, pp. 725-737, 2019.

[13] Zheng J, Dong X, Shen Y, et al. Decentralized and Secure Cross-Domain Data Sharing Scheme Based on Blockchain for Application-Centric IoT[J]. Journal of Information Science & Engineering, 2020, 36(4).

[14] Tong W, Dong X, Shen Y, et al. BC-RAN: Cloud radio access network enabled by blockchain for 5G[J]. Computer Communications, 2020.

[15] M. Wang, C. Qian, X. Li, and S. Shi, Collaborative validation of public-key certificates for iot by distributed caching, in IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019, pp. 847-855.

[16] H. F. Atlam and G. B. Wills, An efficient security risk estimation technique for risk-based access control model for iot, Internet of Things, vol. 6, p. 100052, 2019.

[17] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli and M. H. Rehmani, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1676-1717, Secondquarter 2019.

[18] Kang, Jiawen, et al. "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains." IEEE Transactions on Industrial Informatics 13.6 (2017): 3154-3164.

[19] Li, Ruinian, et al. "Blockchain for large-scale internet of things data storage and protection." IEEE Transactions on Services Computing 12.5 (2018): 762-771.

[20] Novo, Oscar. "Blockchain meets IoT: An architecture for scalable access management in IoT." IEEE Internet of Things Journal 5.2 (2018): 1184-1195.

[21] China Hangzhou Credit Data Technology Co, (2020). Gxb Blockchain White Paper. [Online]. Available: https://static.gxb.io/files/GXChain_WhitePaper_v3.0_CN.pdf.

[22] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum and D. N. K. Jayakody, "A Blockchain-Based Framework for Lightweight Data Sharing and Energy Trading in V2G Network," in IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5799-5812, June 2020.

[23] K. Fan et al., "A Secure and Verifiable Data Sharing Scheme Based on Blockchain in Vehicular Social Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5826-5835, June 2020.

[24] X. Yang, T. Li, X. Pei, L. Wen and C. Wang, "Medical Data Sharing Scheme Based on Attribute Cryptosystem and Blockchain Technology," in IEEE Access, vol. 8, pp. 45468-45476, 2020.

[25] Hern¨¢ndez-Ramos, Jos¨¦ L., et al. ”Distributed capability-based access control for the internet of things.” Journal of Internet Services and Information Security (JISIS) 3.3/4 (2013): 1-16.

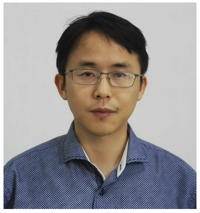
[26] Ouaddah, Aafaf and Abou Elkalam, Anas and Ait Ouahman, Abdellah, ”FairAccess: a new Blockchain-based access control framework for the Internet of Things,” in Security and Communication Networks, vol. 9, no.18, pp. 5943-5964, 2020.

[27] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, Smart contract-based access control for the internet of things, IEEE Internet of Things Journal, 2018.

Xinghui Zhu received his BS and MS degrees in computer science from Xidian University, China in 2014 and 2017, respectively. He is working toward the Ph.D degree in the school of computer science and technology, Xidian University. His research interests include data security and IoT security.

Jiawei Zheng received his BS degree in electronic and information engineering from Liaoning University of Technology, China in 2017, and MS degree in school of computer science and technology, Xidian University, China in 2020. He is now working toward the Ph.D degree in the School of Informatics, University of Edinburgh, UK. His research interests include Edge computing, access control in IoT and application of Blockchain.

Baoquan Ren received his B.S. degree from the Zhangjiakou Communication College, Xuanhua, China in 1997, and M.S. degree from the Commanding Communications Academy, Wuhan, China in 2002. Then he was with the Institute of China Electronic System Engineering Corporation as an engineer and received his Ph.D degree from the Institute of China Electronic System Engineering Corporation, Beijing, China in 2010. He is now with the Institute of Systems Engineering, Academy of Military Science as a senior engineer. His current research interests include Internet of things, wireless communication, and mobile communication network technology.



Xuewen Dong received the BE, MS and Ph.D degrees in computer science and technology from the Xidian University of China, Xi¡¯an, China, in 2003, 2006 and 2011, respectively. From 2016 to 2017, he was with the Oklahoma State University, Stillwater, Oklahoma as a visiting scholar. Currently, he is an associate professor with the School of Computer Science, Xidian University, China. His research interests include cognitive radio network, wireless network security, and big data privacy

Yulong Shen received the BS and MS degrees in computer science and the Ph.D degree in cryptography from Xidian University, Xian, China, in 2002, 2005, and 2008, respectively. He is currently a professor with the School of Computer Science and Technology, Xidian University, Xian, China. He is also an associate director of the Shaanxi Key Laboratory of Network and System Security, Xidian University, China. He has also served on the technical program committees of several international conferences, including ICEBE, INCoS, CIS, and SOWN. His research interests include wireless network security and cloud computing security