JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 821-836 (2020)
DOI: 10.6688/JISE.202007_36(4).0008

Decentralized and Secure Cross-Domain Data Sharing Scheme Based on Blockchain for Application-Centric IoT

JIAWEI ZHENG[1], XUEWEN DONG[1,+], YULONG SHEN[1] AND WEI TONG[2]
[1]*School of Computer Science and Technology*
[2]*School of Cyber Engineering*
Shaanxi Key Laboratory of Network and System Security
Xidian University
Xi'an, 710071 P.R. China
E-mail: jwzheng@stu.xidian.edu.cn; xwdong@xidian.edu.cn;
ylshen@mail.xidian.edu.cn; wtong@stu.xidian.edu.cn

In recent years, more and more Internet of Things (IoT) applications are emerging to make our daily life smarter. However, these IoT applications are developed in an isolated vertical architecture and suitable for a specific scene or environment. In other words, they do not exchange and reuse data between each other, resulting in the issue of IoT application isolations. To address this problem, we propose a cross-domain data sharing mechanism named MicrothingsChain, which allows the exchange and reuse of data among various applications in a collaborative way. This paradigm calls for novel security and access control mechanisms to enable resource-limited smart objects to verify a claimed access right without relying on central authorization systems. In this paper, we present a novel smart contract-based cross-domain authentication and access control mechanism for application-centric IoT. Specially, we propose a proof-of-edge computing node consensus protocol to avoid centralization in practice. The security analysis and experimental results demonstrate that MicrothingsChain is suitable for practice.

Keywords: Internet of Things, blockchain, smart contract, edge computing, data sharing, cross-domain authentication, access control

1. INTRODUCTION

The Internet of Things (IoT) provides a connection to various network-embedded devices via the Internet. It allows the interaction among devices (*e.g.*, sensors and actuators) dedicated to and deployed for an application to fulfill common objectives. The universal interconnection of smart devices greatly accelerates the original IoT data perception, aggregation, assort and sharing in the IoT, making IoT easy-to-access infrastructures for various wise IoT applications such as smart medical, smart traffic, smart logistics, *etc.* [1,2].

However, these IoT applications are usually developed in isolated vertical application architecture and suitable for a specific scene or environment [3]. In other words, they do not exchange and reuse data among each other, forming various application-based information isolations. They are not capable of providing more comprehensive services integrate with corresponding services opened by other IoT scenarios due to the lack of interoperability among them [4,5]. If this trend continues, more and more IoT information will be isolated [6], which will cause a huge cost of resources [7].

To achieve a true IoT vision, ensuring security is a key issue [8]. Primary security threats that IoT data sharing tends to face include unauthorized access, illegal modification, and impersonated publication and retrieval [9]. It is necessary to design a flexible

Received August 28, 2019; revised September 25, 2019; accepted November 7, 2019.
Communicated by Xiaohong Jiang.

[+] Corresponding author: xwdong@xidian.edu.cn and secure IoT data sharing scheme. In the relevant literature, the most common access control (AC) models include the AC list (ACLs) [10, 11], Role-based AC (RBAC) [12] and attribute-based AC (ABAC) [13]. One problem common to ACL, RBAC, and ABAC is centralization, as a central entity is responsible for making authorization decisions, and enforcing access policies. However, in a distributed IoT, these models may not meet what requirements. The capability-based authorization approach offers flexibility that can meet the requirements of various IoT architectures. Anggorojati *et al*. [14] provide a vision for identity and capability-based AC model to handle authority delegation in cross-domain IoT environments in which a central entity in each domain is in charge of authorizing a delegation request from a delegator to a delegate. This approach relies mainly on a central entity for asserting a delegation request. This centralization makes the approach susceptible to a single point of failure (SPOF) issues. A capability-based AC is proposed by Hernández-Ramos *et al*. [15] realizing completely distributed access control without intermediate entities implementing AC logic. That is, IoT objects are able to evaluate access authorization requests and thus decide whether to grant or deny access. This fully decentralized access control method, however, ignores those IoT objects equipped with limited resources, so that they are not able to make authorization decisions.

In unmanned aerial vehicles (UAV) enabled IoT scenarios (As shown in Fig. 1) [16], data is collected from the IoT sensors and devices, forwarded to the nearest server through a UAV swarm, which acts as an edge computing server for computing and making decisions, due to the UAVs are currently equipped with high computing ability and storage capacity. When an IoT application of Domain B wants to access the data from Domain A, there is a need to establish a secure cross-domain access control mechanism to ensure legal authorization decisions.

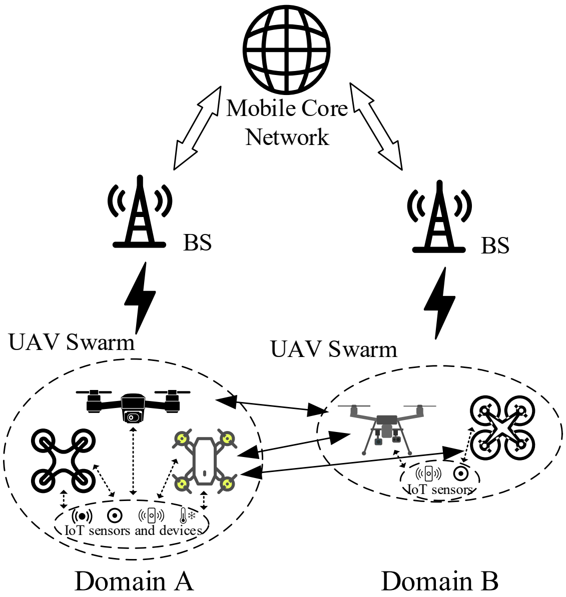


Fig. 1. UAV enabled IoT scenarios.

Inspired by blockchain [17], a distributed database that is used to maintain a continuously growing list of records, we put forward a secure data sharing mechanism to address the above issues in cross-domain data sharing and access control in distributed IoT systems [18, 19]. Considering its decentralization and tamper-proof features [20], we intend to utilize the blockchain to build a secure data sharing mechanism for application-to-application interconnection. This mechanism aims to achieve distributed and trustworthy data interaction in an application-centric IoT environment. Thanks to the invention of smart contracts [21, 22] (executable codes that reside in the blockchain), the blockchain has now evolved into a promising platform for developing distributed and trustworthy applications. Therefore, we aim to apply the smart contract-enabled blockchain technology to achieve distributed and trustworthy access control for the cross-domain data sharing scheme. The most popular and widely used consensus protocols in blockchain, such as PoW (Proof of Work), PoS (Proof of stake) or DPoS (Delegated Proof of Stake), exist the problem of centralization in practice. They still have privileged nodes that possess stronger computing ability or more stake in the system. These nodes usually generate most of the blocks and control the blockchain to some extent. So, these consensus protocols are inapplicable in a distributed IoT environment, deviating the intention of fairness and decentralization in cross-domain data sharing system.

In this paper, to solve the challenges discussed above, we propose a blockchain-based decentralized and secure cross-domain data sharing scheme, called **MicrothingsChain**. When using the MicrothingsChain, various heterogeneous data will automatically be converted and adapt to the standard data format for sharing without administrators to manually modify and apply multiple data identification in IoT environments. At the same time, MicrothingsChain provides decentralized and secure cross-domain authentication and access control. As mentioned above, the main contributions to the research of this work can be summarized as listed below:

1. To the best of our knowledge, this is the first work to propose a decentralized and secure cross-domain data sharing mechanism based on blockchain. Through decentralized and efficient access control, MicrothingsChain can resist identity forgery and operation tamper attacks effectively.
2. To avoid centralization in practice, we design a proof-of-edge computing node consensus protocol to achieve trust dispersing.
3. We also propose a smart contract-based cross-domain authentication and access control scheme using the blockchain technology to achieve the static and dynamic validation of the cross-domain access control.
4. We develop a prototype system in a user case to implement the proposed MicrothingsChain mechanism and further evaluate its practicability.

The rest of the paper is structured as follows. In Section 2, we give a description of the system model, threat and security requirements. We describe the details of our MicrothingsChain in Section 3 and analyze its security properties in Section 4. Section 5 introduces the implementation of our prototype and evaluation of the MicrothingsChain. Finally, we conclude this paper in Section 6.

2. SYSTEM DESCRIPTION

2.1 System Model

Herein, we build a framework named MicrothingsChain as in Fig. 2 by introducing social networking concepts into IoT, which provides the selection, discovery, and composition of resources through social relationships and circles among objects in the same

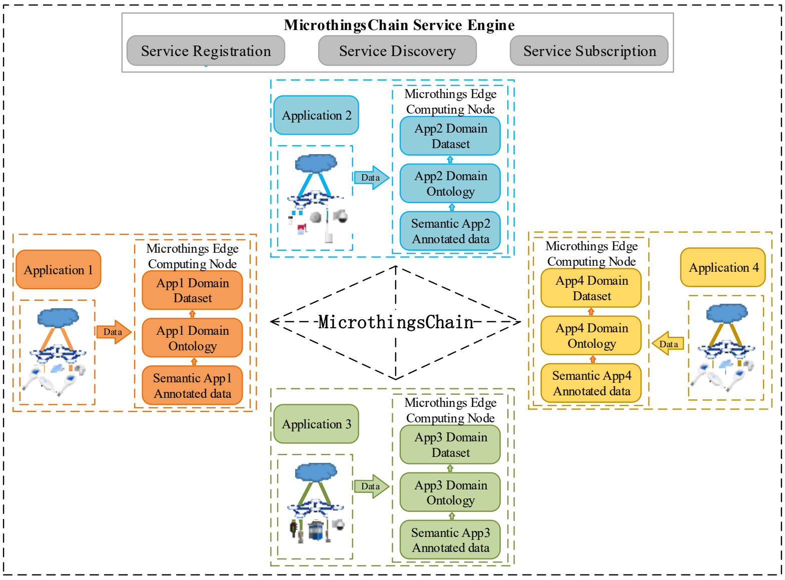


Fig. 2. Overview of the MicrothingsChain.

manner as in social relationships among humans in a social network. As shown in Fig. 2, there are four applications (*e.g.*, smart home, smart city and smart medical) in the edge of the IoT network. In the edge IoT network, there are also Microthings Edge Computing Nodes related to each application which is responsible for integrating the computing, storage, decision ability of the IoT devices. In this way, those devices with constraint resources can delegate their corresponding requirements (like computing, decision-making and so on) to the edge computing node to mitigate the pressure of them. Each Microthings Edge Computing node is equipped with a blockchain node, enabling them to make up a blockchain network by connecting together and social application-to-application communication. Data is collected by each application from its underlying deployment and will be forwarded to the Microthings Edge Computing Node. When the data reaches the Microthings Edge Computing Node, it needs to be enriched with semantics in order to achieve interoperability and enable cross-domain data sharing. Due to the possibility that the same data from various applications may have a different meaning or different data from various applications may have the same meaning, the first step is to apply semantic annotations to the raw data of heterogeneous applications. By semantic annotation, the data can be understood and able to apply logic. For example, let's assume that three applications data are related to temperature measurements. However, if they used their own notations to represent temperature, *e.g.*, '*t*', 'temp' or 'temperature', it can be understandable by humans but not by machines. Therefore, Microthings edge computing nodes first apply semantic annotations to describe the data in order to make it understandable by the machines. The second step is to find the appropriate domain ontology of the applications' data. 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. After obtaining the relevant ontologies, the next step is to retrieve the most relevant datasets in order to acquire additional knowledge. Finally, the MicrothingsChain service engine provides service registration, discovery, and subscription, performing the index of each Microthings edge computing node.

The access control (AC) policy can be used to define the access conditions under which a user can access the other IoT applications' data in complex scenarios. As mentioned above, those traditional access control methods do not meet the requirements in the MicrothingsChain distributed IoT system. In a specific scenario, the IoT devices deployed in the application share common missions and access requirements. Thus, the access control mechanism can rely on application-based structure, which enables smart objects with sufficient resources to make authorization decisions on behalf of those with less capability. In view of this, we propose a cross-domain application-based access control mechanism for distributed IoT environments in order to make identity authentication and manage AC rights on behalf of other resource-constrained objects.

2.2 Threats and Security Requirements

In general, an adversary attacks MicrothingsChain for three goals: (1) accessing data from other domains without being authorized; (2) modifying the access control policy for making access validation failure; (3) avoid paying the MicroCoin to the data provider.

To inhibit these attacks and provide flexible fine-grained access control on IoT data, we identify the security requirements as follows.

- **Decentralized and cooperative access control:** To reduce the dependence on the central domain and achieve fair access control. This aims to avoid SPOF issues and solve the trust problem among applications.
- **Efficient and Flexible Authorization:** Data publishers can publish data with publisher-defined policies, and the security mechanism enable the authorization to users for accessing a flexible set of data.
- **Ease of deployment:** All nodes can be included without specific configuration, even for those located on the edge of the Internet.
- **Publisher Identity Authentication:** The architecture of the network should be able to adapt to the changing environment and broaden its use to meet the increasing needs and demands of subscribers.
- **Consensus fairness:** By dynamic accommodation, all Microthings can generate blocks for recoding access operations with almost the same probability.

3. MICROTHINGSCHAIN MECHANISM

3.1 Consensus Protocol

Due to the constraint resource of IoT devices, they cannot guarantee the real-time manner and high throughput in IoT scenarios, as mentioned in Section 2.1, we deployed Microthings Edge Computing Node in the edge of IoT network for each application, as the edge computing node is more resourceful than IoT devices and has same distribution characteristic as IoT environment. Therefore, we delegate the computing, storage ability of IoT devices to the edge computing nodes and design a new consensus protocol to address the low efficiency and performance bottleneck problems resulting from resource-constraint IoT devices. These edge computing nodes have the ability to perform as blockchain nodes and meet the IoT applications' requirements of low latency and efficiency.

Following a new family of Byzantine Fault-Tolerant (BFT) algorithms, named Proof-of-Authority (POA), we design a Proof-of-Edge Computing Node consensus protocol by defining each edge computing node as the blockchain node to validate the operations.

In this protocol, the decentralization and fairness can be guaranteed, since each edge computing node is equally selected as *validators* in sequence to packaged block.

Proof-of-Edge Computing Node consensus protocol

This is a protocol run by each edge computing node N_1, ..., N_n, who are defined as *validators* to validate the operations on blockchain. The protocol proceeds as follows:

1. **Initialization:** When the protocol starts, each edge computing node broadcasts its public key. Then all of them maintain the *validators* list and their genesis block used to initialize the blockchain.

2. **Chain extension:** There are two queues that the edge computing node maintains locally, one of the queues is for transactions Q_{txn} and another for pending blocks Q_b. Each committed transaction is packaged by authorities in Q_{txn}. In this step, they will select one leader l in the order of the *validators* list ensuring the fairness, then the selected leader l packages the transactions committed by other blockchain members in Q_{txn} in a block b. After that, the leader l broadcasts it to the other edge computing nodes. Then each edge computing node sends the received block b to the others to execute the block acceptance. If all the edge computing nodes received the same block b packaged by identified leader l, they append the block b by enqueuing it in Q_b.

3. **Resistance attack:** Any received block sent by an edge computing node different from the current leader is rejected. The leader is always expected to send a block if no transaction is available then an empty block has to be sent. If the edge computing nodes do not agree on the received block during the block acceptance, voting is triggered to decide whether the current leader is malicious and then kick it out. An edge computing node can vote the current leader malicious because:

   (a) The edge computing node has not sent any block even an empty block;

   (b) The edge computing node has proposed more blocks than prospective;

   (c) The edge computing node has proposed different blocks to other edge computing nodes;

   When a majority of votes yes, the current leader l will be kicked out from the set of authority node list. All the blocks in Q_b mined by leader l will be abandoned.

Compared to other consensus protocols like PoW, PoS, they are centralized to some extent. Proof-of-Edge Computing Node consensus protocol is maintained by each edge computing node, which is fair, decentralization and represents each application's mission. This consensus protocol generates a block at the millisecond level in comparison to the 10 minutes of PoW. So the Proof-of-Edge Computing Node consensus protocol is suitable in the cross-domain application-centric IoT scenario.

3.2 Authentication Contract

The *Authentication Contract* consists of multiple cross-domain authentication contracts (CDAC), each of which implements the cross-domain authentication for members of the blockchain network, one judge contract (JC), which judges the misbehavior and determines the corresponding penalty. *Authentication Contract* realizes not only static validation by checking predefined policies but also dynamic validation by checking the behavior of the domain.

Table 1. The CDAC policy list.

Domain	Accesstoken	ToAG	Expire
domain1	Ac******dgf	1527759579	3600
domain2	Be******d4w	1521797103	3600
...	...	...	...

Each member maintains the same cross-domain authentication policy. In this way, this mechanism satisfies the decentralized and cooperative access control requirements. To achieve the cross-domain authentication, the CDAC maintains a policy list as illustrated in Table 1, in which each row corresponds to the policy defined on a certain pair. The basic fields of each row are:

- Domain: the account of own edge computing node;
- Accesstoken: an access token contains security information for this certified operation, which is performed to verify if the access token corresponds with the domain;
- Time of the Accesstoken generation (ToAG): the time of generation for the access token;
- Expire: the time of the access token expiration.

To record the misbehavior that the domain has exhibited as well as the corresponding penalty, the CDAC also maintains a misbehavior list for each domain, which facilitates the misbehavior judging at the judge contract (as illustrated in Section 3.4). To help to characterize the misbehavior, we added the following fields to the rows in Table 1:

- MinInterval: the minimum allowable time between two successive requests;
- Number of frequent requests (NoFR): the threshold of frequent requests in a short period;
- Time of last request (ToLR): the time of last certification request from the domain.

Based on the above fields, we designed the CDAC protocol as in Algorithm 1, which receives the inputs of *Domain*, *OperaType* and *Access*, and returns the authentication result and this operation info (*i.e.*, the hash of transaction, block and the number of the block number, *etc*.). After that, the CDAC protocol also contains a JC instance, through which the Judge function of the JC can detect the misbehavior of this domain and record it into a misbehavior list in CDAC and misbehavior record in JC, assisting the CDAC protocol to complete verification.

3.3 Access Control Contract

An access control contract (ACC) carries out the validation, by deploying distributed and trustworthy access control strategies on each Microthings edge computing node. An ACC is deployed by a member of MicrothingsChain who wants to access data from another member. We assume that the other member can agree on multiple access control policies. An example of the ACC is given as follows. To achieve the access control, the ACC maintains a policy list as illustrated in Table 2, in which each row corresponds to the policy defined on a certain pair. The basic fields of each row are:

- Domain: the account of own edge computing node;
- Access domain: the account of the access domain that is defined as the domain to access;
- Permission: the static permission predefined on the action, such as allow, deny, *etc*.;
- Time of last request (ToLR): the time of last access request from the domain.

Table 2. The ACC policy list.

Domain	Access domain	Permission	ToLR
domain1	domain2	allow	1521797903
domain2	domain3	deny	1527760834
...	...	...	...

Algorithm 1: The CDAC Protocol

```
Input: $Domain, OperaType, Access.$
Output: $Result, Tx_{info}.$
  $token \leftarrow Hash(Access)$
  if $Reg = OperaType$ then
    $DomainDetail_{Domain}.Accesstoken \leftarrow token;$
    $DomainDetail_{Domain}.ToAG \leftarrow T_{current};$
  else
    if $Check = OperaType$ then
      $r \leftarrow DomainDetail_{Domain};$
      if $1 \leftarrow Verify(token, r.Accesstoken)$ then
        $penalty \leftarrow MisbehaviorList_{Domain}.Penalty$
        $blockTime, Time \leftarrow Extract\{penalty\};$
        if $T_{current} \geq blockedTime + Time$ then
          $T_{ToAG}, T_{Expire} \leftarrow Extract\{r\};$
          if $T_{current} \leq T_{ToAG} + T_{Expire}$ then
            return 1;
          end if
          if $T_{current} - r.ToLR \leq r.MinInterval$ then
            $num \leftarrow num + 1;$
            if $num \geq r.NoFR$ then
              Detect a misbehavior $msb$;
              $Penalty \leftarrow Judge(Domain, msb);$
              Push $msb$ into the misbehavior list;
              $mis \leftarrow MisbehaviorRecord_{Domain};$
              $mis.Penalty \leftarrow Penalty;$
              $mis.Misbehavior \leftarrow msb;$
              $mis.Time \leftarrow T_{current};$
            end if
          end if
        end if
      end if
    end if
  end if
```

The *Permission* field can be used for static validation, and the *ToLR* can be used for dynamic validation, such as detecting the misbehavior that the domain sends access requests too frequently in a short period. The access control validation process is shown as Algorithm 2, including the registration of access control policy. Notice that only the object domain can register, update and delete the corresponding ACC policies.

3.4 Judge Contract (JC)

The JC implements a misbehavior judging method, which judges the misbehavior of the domain and determines the corresponding penalty when receiving a potential misbehavior report from the CDAC. The penalty can be based on the misbehavior history of the domain, so the JC may need to keep a record of the misbehavior history of all domains. After determining the penalty, the JC returns the decision to the CDAC for further operation.

Algorithm 2: The ACC Protocol

```
Input: $Domain, OperaType, AccessDomain, Action.$
Output: $Result, Tx_{info}.$
  $subject \leftarrow Domain;$
  $object \leftarrow AccessDomain;$
  if $Reg = OperaType$ then
    if $object = req.sender$ then
      $p \leftarrow ACCpolicies_{object};$
      $p.Domain \leftarrow subject;$
      $p.AccessDomain \leftarrow object;$
      $p.Permission \leftarrow Action;$
      $p.ToLR \leftarrow T_{current};$
      return 1;
    end if
  else
    if $policyCheck = OperaType$ then
      if $1 \leftarrow PolicyCheck(subject, object)$ then
        $c \leftarrow ACCpolicies[subject][object];$
        if $c.Permission = "allow"$ then
          return 1;
        end if
      end if
    end if
  end if
```

3.5 Incentive Mechanism

In this section, we introduce one cryptocurrency into our ecology named MicroCoin as an incentive medium for data sharing. The incentive mechanism is realized by making data monetization and forming a loop system of the data stream and value flow. At first, for data publishers, when they want to publish data, they should set the price of corresponding data according to different usages (like pricing by period or times). Next, users or application developers should subscribe to the data publisher, who has published the data they needed from the edge computing node pool. Then, the data publisher will supply the subscribed data (*e.g.* the status data of a certain area and the medical information of someone). In the process of providing the needed data, the data publisher will submit a transaction to the blockchain equipped smart contract to handle the pre-setting business model. All edge computing nodes will reach a consensus through the Proof-of-Edge computing node consensus protocol. Finally, the user will pay for the MicroCoin to data publishers for using the data. Further, this mechanism takes not only the economic benefit of providing data but also the misbehavior into consideration. If a domain has some misbehavior monitored by the Judge Contract (JC), such as too frequent requests, it will be punished in the form of reducing its MicroCoin. Through this mechanism, data publishers would be encouraged to provide more economic data, at the same time, the misbehavior of the data consumers would be restricted.

4. SECURITY ANALYSIS

Theorem 1: In MicrothingsChain, the cross-domain authentication operation can be traced efficiently, and access control checking can be fed back accurately without false positives.

Proof: For a given cross-domain authentication request, we can get the history cross-domain authentication operations efficiently without traversing the b lockchain. The reason is that the blockchain provides authentication operation chains in the form of transactions for all domains whose authentication operations are recorded in the blockchain. For access control checking, it is well-known that, compared to traditional data access control structure exploited only by the center domain, smart contract-based access control is maintained in a distributed network in the blockchain. In MicrothingsChain, since all the access control operations are recorded in the blockchain, it is easy to get the accurate status of the request operations through the smart contract-based access control policy.

Theorem 2: By consensus mechanism and judge contract, this architecture can tolerate the failure of the single point. By traversing blockchain, the operations cannot tamper.

Proof: Note that the process of access validation launched by other domain includes the cross-domain authentication and access control in the blockchain. In addition, the judge contract can complete the dynamic validation and punish the misbehavior. According to this, all the corresponding operations must be recorded in blockchain to receive self and other domain audits, otherwise, the access validation cannot pass the validation. What's more, the Proof-of-Edge computing node consensus mechanism based on POA can automatically kick the suspicious node out, and the blocks they have mined can be abandoned. Therefore, even if an attacker compromises a node and creates malicious operations on the blockchain, the other node can detect this node by judge contract and consensus mechanism. Tampered operations can be retrospective by traversing blockchain. Thus, the compromised node would be identified quickly.

5. EXPERIMENT AND EVALUATION

In order to validate and evaluate the MicorthingsChain's practicability, the use case and implementation provided in this section are designed to demonstrate MicrothingsChain operation in real life.

Since the IoT devices deployed in the application share common missions and access requirements, we realize the application-based structure for distributed IoT environments to make AC decisions on behalf of other resource-constrained IoT objects. These applications all equipped with Microthings edge computing nodes to realize the data sharing among other applications. Each edge computing node is given a blockchain account and password, so as to execute the behavior of mining block and block acceptance. We implement the MicrothingsChain prototype to demonstrate the access token generation and permission validation. The goal is to demonstrate the effortless deployment of our proposed MicrothingsChain in real-life scenarios.

Firstly, we give the configuration of the MicrothingsChain. The configuration file is shown in Fig. 3, in which the authority nodes are listed as *validators* for packaging and validating the block. By setting the step duration as 5 seconds, the blocks are generated every 5s. With this configuration, a new node can be involved without other settings, meeting the requirement of easy deployment.

```
{
  "name": "MicrothingsChain",
  "engine": {
    "authorityRound": {
      "params": {
        "stepDuration": "5",
        "validators" : {
          "list": [
              "0x00Aa39d30F0D20FF03a22cCfc30B7EfbFca597C2",
              "0x004ec07d2329997267Ec62b4166639513386F32E",
              "0x00a94Ac799442FB13De8302026fd03068bA6A428",
              "0x002E28950558Fbede1A9675Cb113F0BD20912019",
              "0x00D4f0e12020C15487B2a525ABCB27dE647C12de",
              "0x001f477a48A01d2561E324F874782B2Dd8167772",
              "0x006137D98307aB6691CCedb7A10B295Da8ae1035",
              "0x003f3b1f635B2dD9a4518c33098e5f72214d6a1E",
              "0x008272A8CFd2D3D0F3EDc823b1BB729cb73F09db",
              "0x001CE0F63558e2Fe10806d132d64D2B2F63Ef64e"
              ]
        }
      }
    }
  },
  "params": {
    "gasLimitBoundDivisor": "0x400",
    "maximumExtraDataSize": "0x20",
    "minGasLimit": "0x1388",
    "networkID" : "0x2323"
  },
}
```

Fig. 3. The configuration file of MicrothingsChain.

Secondly, we compare the performance of MicrothingsChain with different amounts of blockchain nodes. We measured the detail processing time of three steps that interact with the blockchain. Measurements are given as the average over 100 test runs, and the results are presented in Fig. 4. The results show that MicrothingsChain is practical and enables elastic expansion since the processing time of these three steps is nearly stable as the number of nodes increases.

Thirdly, we present the use case, which takes place in a smart city. Note that MicrothingsChain is composed of ten IoT applications deployed with the specific devices and both Alice and Bob are blockchain nodes.

1. Alice needs to develop a smart city application demanding Bob's transportation data from other domains, so Alice should subscribe to Bob's data through MicrothingsChain and pay the responded MicroCoin for the fee. After that, Bob sets policy on

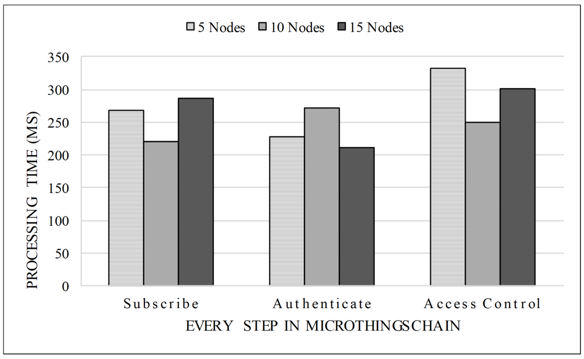


Fig. 4. Processing time (in milliseconds) of every step.

CDAC and ACC that Alice is authenticated and accessible to the data. The CDAC generates the access token for Alice and sends it to Alice. This process happens on blockchain as a transaction, so it will be recorded by all blockchain nodes.

2. Alice tries to access the data with the access token. When receiving Alice's request, Bob sends a transaction to the blockchain, which contains the information necessary for authentication, to call the CDAC. This transaction will be packed in a new block and will be included in the blockchain. During the CDAC verification process, the CDAC will send a message to call the JC, if some potential misbehavior of the subject is detected.

3. After the CDAC protocol verification, the CDAC will return the result and the transaction information to the Bob.

4. If the CDAC verification passes, the Bob sends another transaction, which contains the required information for access control, to call the ACC. Then ACC will decide that Alice has access to the data as pre-specified by Bob.

5. After the request to the ACC, the access result will be returned to the Bob, then the access control process finishes. With the access token held by Alice, she can access the transportation data provided by Bob. Based on blockchain technology, each member of the blockchain holds its private key to sign transactions. The CDAC and ACC evaluate the request to make sure the identity has not been forged before allowing or denying access to the object. Later, Bob can revoke Alice's access to the data.

Steps 1 to 5 are illustrated in Fig. 5.

Alice subscribes to Bob's data through MicrothingsChain Service Engine by paying MicroCoin to MicrothingsChain DApp, as shown in Fig. 6(a)1. This transaction will be mined into a block and recorded in the blockchain, returning the transaction hash for tracking user behavior, as shown in Fig. 6(a)2. MicrothingsChain DApp consisting of authentication contract and access control contract is deployed on each Microthings edge computing node. An authentication contract is responsible for matching an access request with its assigned access rights (using the blockchain account and access token as a pair key for the matching). If the pairing is correct, a successful channel to access control contracts will be given and sent to the requester. A light sensor data request is used to demonstrate successful access authorized by Microthings DApp, as shown in Fig. 6(b).

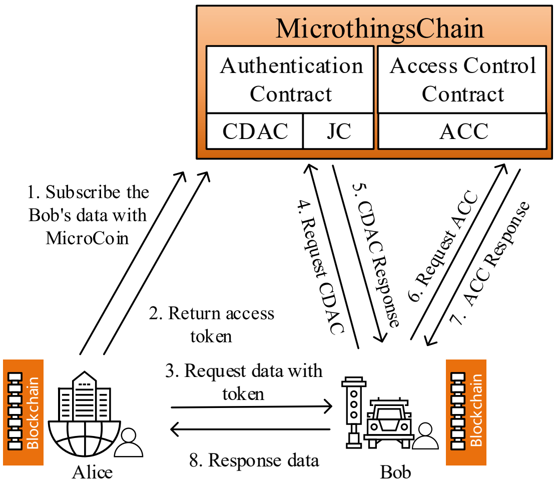


Fig. 5. A use case of MicrothingsChain.

6. CONCLUSION

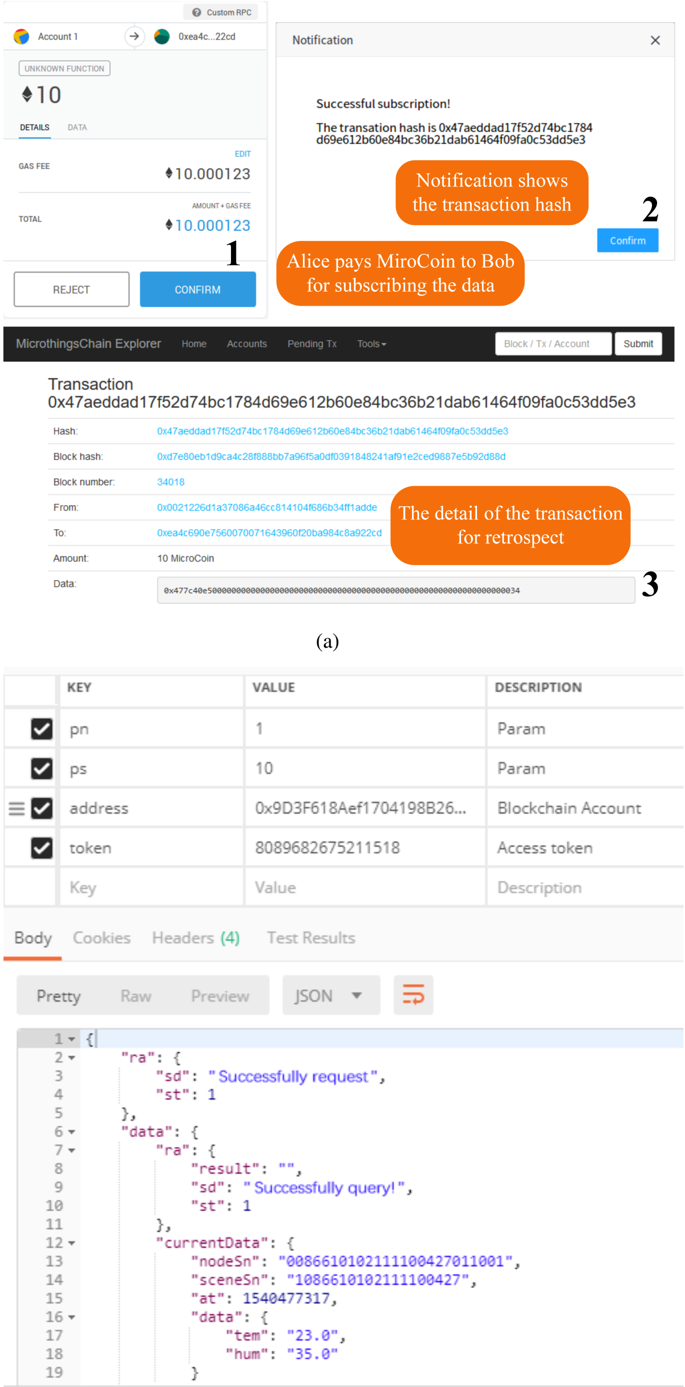
In this paper, we proposed a novel decentralized and secure cross-domain data sharing scheme based on blockchain (MicrothingsChain) to bridge the gap of IoT applications' isolation. It applies characteristics of blockchain to provide a decentralized and tamper-proof access control mechanism. Specially, we design a proof-of-edge computing node consensus protocol to avoid centralization in practice. To facilitate data sharing between different applications, we present an incentive mechanism by MicroCoin. Secure proof and experimental results show that MicrothingsChain is suitable for practice.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China (No. 2017YFB 1400700), Key R&D Program of Shaanxi Province (No. 2019ZDLGY12-03, 2019ZD LGY13-06, 2019ZDLGY13-01), National Natural Science Foundation of China (No. 61972310, 61972308, 61902295), Project of Hainan Province Key Research and Development Program (No. ZDYF2019202).

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, 2017, pp. 86-93.
2. 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*, Vol. 6, 2018, pp. 1594-1605.
3. 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," in *Proceedings of IEEE International Conference on Networking and Network Applications*, 2018, pp. 350-355.



♦10

GAS FEE ♦10.000123

AMOUNT + GAS FEE

TOTAL ♦10.000123

Successful subscription!

The transation hash is 0x47aeddad17f52d74bc1784d69e612b60e84bc36b21dab61464f09fa0c53dd5e3

Notification shows the transaction hash

Alice pays MiroCoin to Bob for subscribing the data

Transaction 0x47aeddad17f52d74bc1784d69e612b60e84bc36b21dab61464f09fa0c53dd5e3

Hash:	0x47aeddad17f52d74bc1784d69e612b60e84bc36b21dab61464f09fa0c53dd5e3
Block hash:	0xd7e80eb1d9ca4c28f888bb7a96f5a0df0391848241af91e2ced9887e5b92d88d
Block number:	34018
From:	0x0021226d1a37086a46cc814104f686b34ff1adde
To:	0xea4c690e7560070071643960f20ba984c8a922cd
Amount:	10 MicroCoin
Data:	0x477c40e500000000000000000000000000000000000000000000000000000000000034

The detail of the transaction for retrospect

(a)

	KEY	VALUE	DESCRIPTION
☑	pn	1	Param
☑	ps	10	Param
☑	address	0x9D3F618Aef1704198B26...	Blockchain Account
☑	token	8089682675211518	Access token
	Key	Value	Description

```
{
    "ra": {
        "sd": " Successfully request",
        "st": 1
    },
    "data": {
        "ra": {
            "result": "",
            "sd": " Successfully query!",
            "st": 1
        },
        "currentData": {
            "nodeSn": "0086610102111100427011001",
            "sceneSn": "1086610102111100427",
            "at": 1540477317,
            "data": {
                "tem": "23.0",
                "hum": "35.0"
            }
```

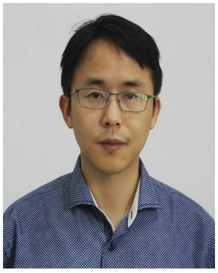
(b)

Fig. 6. A use case of MicrothingsChain.

4. W. Tong, X. Dong, Y. Shen, and X. Jiang, "A hierarchical sharding protocol for multi-domain iot blockchains," in *Proceedings of IEEE International Conference on Communications*, 2019, pp. 1-6.
5. Z. Tao, L. Zheng, Y. Wang, Y. Shen, X. Ning, J. Ma, and J. Yong, "Trustworthy service composition with secure data transmission in sensor networks," *World Wide Web-internet & Web Information Systems*, 2017, pp. 1-16.
6. J. Zheng, X. Dong, W. Tong, Q. Liu, and X. Zhu, "Blockchain-based secure digital asset exchange scheme with qos-aware incentive mechanism," in *Proceedings of IEEE 20th International Conference on High Performance Switching and Routing*, 2019, pp. 1-6.
7. L. Wang and R. Ranjan, "Processing distributed internet of things data in clouds," *IEEE Cloud Computing*, Vol. 2, 2015, pp. 76-80.
8. J. S. Kumar and D. R. Patel, "A survey on internet of things: Security and privacy issues," *International Journal of Computer Applications*, Vol. 90, 2014, pp. 20-26.
9. A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, Vol. 112, 2017, pp. 237-262.
10. J. Qian, S. Hinrichs, and K. Nahrstedt, "Acla: A framework for access control list (acl) analysis and optimization," *Communications and Multimedia Security Issues of the New Century*, Springer, 2001, pp. 197-211.
11. P. Samarati and S. C. de Vimercati, "Access control: Policies, models, and mechanisms," in *International School on Foundations of Security Analysis and Design*, Springer, 2000, pp. 137-196.
12. D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*, Artech House, Inc., Norwood, MA, 2003.
13. C. A. Ardagna, S. D. C. di Vimercati, G. Neven, S. Paraboschi, F.-S. Preiss, P. Samarati, and M. Verdicchio, "Enabling privacy-preserving credential-based access control with xacml and saml," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, 2010, pp. 1090-1095.
14. B. Anggorojati, P. N. Mahalle, N. R. Prasad, and R. Prasad, "Capability-based access control delegation model on the federated iot network," in *Proceedings of the 15th IEEE International Symposium on Wireless Personal Multimedia Communications*, 2012, pp. 604-608.
15. J. L. Hernández-Ramos, A. J. Jara, L. Marın, and A. F. Skarmeta, "Distributed capability-based access control for the internet of things," *Journal of Internet Services and Information Security*, Vol. 3, 2013, pp. 1-16.
16. A. Islam and S. Y. Shin, "Bus: A blockchain-enabled data acquisition scheme with the assistance of uav swarm in internet of things," *IEEE Access*, Vol. 7, 2019, pp. 231-249.
17. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.
18. A. Ramachandran, D. Kantarcioglu, *et al.*, "Using blockchain and smart contracts for secure data provenance management," *arXiv preprint arXiv:1709.10000*, 2017.
19. K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, Vol. 4, 2016, pp. 2292-2303.
20. P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-vn: A distributed blockchain based vehicular network architecture in smart city," *Journal of Information Processing Systems*, Vol. 13, 2017, p. 84.

21. An Introduction to Ethereum Platform, http://ethdocs.org/en/latest/introduction/what-is-ethereum.html, 2018.
22. Introduction to Smart Contracts, http://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html, 2018.

Jiawei Zheng is now a M.S. candidate in School of Computer Science and Technology, Xidian University. His research interests include edge computing, access control in IoT and application of Blockchain.



Xuewen Dong is now an Associate Professor and M.S. tutor at the School of Computer Science and Technology, Xidian University. His research interests include wireless network security, IoT, blockchain theory and application.

Yulong Shen is a Professor at the School of Computer Science and Technology, Xidian University, China. He is a Director of the Shaanxi Key Laboratory of Network and System Security and a member of the State Key Laboratory of Integrated Services Networks. He has served on the Technical Program Committees of several international conferences, including ICEBE and INCoS. His research interests include IoT, wireless network, and cloud computing security.

Wei Tong is now a Ph.D candidate in School of Cyber Engineering, Xidian University. His research interests include Blockchain theory and application.