Hindawi
Security and Communication Networks
Volume 2021, Article ID 3091104, 11 pages
https://doi.org/10.1155/2021/3091104

Research Article

A Blockchain-Based IoT Cross-Domain Delegation Access Control Method

Chao Li,[1] **Fan Li**,[1,2] **Lihua Yin**,[1] **Tianjie Luo**,[1,2] **and Bin Wang**[3]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510700, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Nanning 541004, China*
[3]*College of Electrical Engineering, Zhejiang University, Hangzhou 310058, China*

Correspondence should be addressed to Lihua Yin; yinlh@gzhu.edu.cn and Bin Wang; bin_wang@zju.edu.cn

Received 18 June 2021; Accepted 23 August 2021; Published 11 September 2021

Academic Editor: Yinghui Zhang

The collaborative demand in the Internet of Things (IoT) is becoming stronger. One of the collaborative challenges is the security of interoperability between different management domains. Although cross-domain access control mechanisms exist in IoT, the majority of them are based on a trusted third party. In addition, the heterogeneity of multidomain policies makes it difficult for authority delegation to satisfy the principle of least authority. In this paper, we propose a blockchain-based IoT cross-domain delegation access control method (CDDAC). The delegation-trajectory-on-blockchain strategy proposed enhances the scalability of the cross-domain delegation system. The presented multidomain delegation trajectory aggregation scheme supports the forensic analysis of the cross-domain delegation system. The performance of CDDAC is evaluated in the Ropsten, which is the Ethereum's official public blockchain test network. The experimental results show that CDDAC has faster delegation verification speed and higher decision-making efficiency than existing work, demonstrating the lightweight and scalability of the method.

1. Introduction

Internet of Things (IoT) has been widely used in many fields, such as smart healthcare [1], smart transport [2], and smart homes [3]. Among these fields, some scenarios have begun to trend towards requiring IoT devices from different domains to share data or collaborate, which makes a significant difference from traditional single-domain applications. In a traditional single-domain application, IoT devices belong to the same domain, in which the domain administrator could manage the devices security policies overall. For example, we assume that a hospital has only one domain, in which many IoT devices such as smart connected-beds, wearable ECG monitors, etc., are deployed to collect patient-related data. The domain administrator could define security policies to manage all the devices in the hospital, to specify which devices can be accessed by whom and under what circumstances; for example, a patient's ECG monitor can be accessed by his family and nurses. In contrast, in a cross-domain application, the users, devices, and data belong to different domains. Many functions require devices and data shared in multiple domains to be achieved. Assume an application requirement that "*If there is a traffic jam, the ambulances nearby then get the alarms and some new recommended navigation routes*," it requires the traffic domain and the hospital domain to collaborate. Traditional single-domain access control mechanisms are difficult to meet this requirement, since each domain administrator cannot manage the other domain devices.

To fulfil the IoT cross-domain access control requirement, Payne et al. [4] connect IoT domains according to certain agreements to form a virtual alliance. They propose the National Health Information Network (NHIN) uniting the IoT domains of multiple hospitals to form a virtual alliance of medical systems. The alliance facilitates a smoother information flow between doctors and patients and initially solves the problem of cross-domain access control. However, this kind of method faces challenges in cross-domain delegation [5]. Access right delegation is one of the ways to realize IoT cross-domain connection [6]. Due

to the heterogeneity of security constraints in IoT domains, universal authorization protocols are difficult to obtain. Authorization capabilities are commonly granted to users in the IoT domain by special carriers (e.g., the OAuth token, the secret URL of IFTTT), allowing independent decisions on whether to transfer access rights to users in another domains for reasons of convenience or emergency response, what is called cross-domain delegation. For example, a patient delegates access right of his wearable ECG monitor to his families and caregivers. In the case of a fire, the homeowner needs to delegate the capability to open smart locks to firefighters, which involves an ad hoc access right delegation. While decreasing the decision-making pressure of the trusted central server, delegation mechanism satisfies the decentralized and dynamic characteristics of IoT, which is considered an indispensable feature in large-scale IoT scenarios.

Previous researches have focused on cross-domain delegation and access control by trusted third parties [7,8], while trusted third parties are at risk of being attacked [9]. Furthermore, cross-domain key distribution is a challenging problem [10], and user privacy is more likely to be exposed. Blockchain, as a decentralized mechanism that does not require a trusted third-party potential, is seen as an opportunity to solve the problem of cross-domain delegation. Existing blockchain-based methods rely on two kind strategies of policy-on-blockchain [11–13] and right-on-blockchain [14,15], both of which can cause difficulties in policy changing and are constrained by blockchain performance. To address the shortcomings of existing work, a blockchain-based cross-domain delegation access control method, CDDAC, is proposed in this paper.

CDDAC leverages the decentralized characteristics of the blockchain, to ensure the feasibility and security of cross-domain delegation, and provides evidence for forensic analysis after malicious events have occurred. Our contributions are as follows:

We propose CDDAC, a cross-domain delegation access control scheme based on the blockchain. The delegation trajectory-on-blockchain strategy makes CDDAC more flexible and usable than other methods.

Based on CDDAC, we propose a multidomain delegation trajectory aggregation method with goal-directed logging. It supports forensic analysis of intra/cross-domain delegation and contributes to suspect validation and accountability after malicious behavior occurs.

We conduct simulation experiments of CDDAC on IoT devices. Experiment results show that CDDAC has a faster token verification speed compared with CapBAC and BlendCAC. In the meantime, CDDAC can maintain a high consensus efficiency in the blockchain system.

The remainder of this paper is organized as follows. In Section 2, we describe the work related to the use of blockchain for cross-domain access control. Section 3 mainly describes the system design of CDDAC. Section 4 describes the multidomain delegation trajectory aggregation scheme and the basic design of forensic analysis in CDDAC. Section 5 performs simulation experiments on CDDAC and analyzes the corresponding experimental results. Section 6 summarizes this paper.

2. Related Work

The existing blockchain-based cross-domain access control strategy can be divided into policy-on-blockchain and right-on-blockchain.

2.1. Policy-on-Blockchain. Writing access control policies into smart contracts is a common method in blockchain-based solutions. Novo et al. [10] propose a blockchain-based IoT access control framework, using a management hub to manage IoT devices and making decisions in accordance with the access control policies on the blockchain. But the access control policies need to be managed in a consistent way, and the policy definition or changing is complicated. As a method supporting flexible customization of access control policies, Ouaddah et al. [11] propose FairAccess, a blockchain-based access control framework. FairAccess allows resource owners to define access control or delegation policies and renew them to the blockchain. When there are more participants, the cost of policy synchronization will become unbearable. As a customized solution for cross-domain access control, IoT Passport proposed by Tang et al. [12] is a cross-platform blockchain framework, which uses blockchain authentication, authorization, and trust as the cornerstone to achieve cross-platform collaboration. However, the cross-domain policies of IoT Passport are recorded on the blockchain, which makes policy changing more expensive. Similarly, Gauhar et al. [16] propose a decentralized blockchain-based IoT access control framework, xDBAuth, used for single-domain or cross-domain access control and implemented a platform authentication mechanism in blockchain. The delegation policies of xDBAuth are saved on blockchain, facing difficulties in policy changing. In general, for the policy-on-blockchain strategy, the design of uploading policies to blockchain not only brings a huge synchronization burden, but also is not convenient for policy changing, which happens so frequently in access control systems.

2.2. Right-on-Blockchain. It is a conventional design to substantively pass access rights, which is used to reduce the complexity of delegation. Recently, this design has also been used in cross-domain access control works. Yuan et al. [17] attribute the challenge of cross-domain access control to security and consistent delegation policies and non-bypassable and transitive delegation control. Maesa et al. [18] propose a blockchain-based access management framework, which describes the operation of permission exchange. But the access control policies and rights delegation process are visible publicly on blockchain, which leaks the privacy of users. A similar study is BlendCAC proposed by Xu et al. [14]. BlendCAC uses capability tokens

managed by smart contracts to delete or revoke permissions. Although the overhead in authentication and token verification has been proven acceptable, the delegation process is much more complex. Nakamura et al. [15] propose a decentralized and trusted capability-based access control method. The issue is raised about the limitations of BlendCAC's entrusted records, and smart contracts are used to save and manage tokens. But it only designs a new token and does not solve the problem of poor scalability of such solutions. In short, for the right-on-blockchain strategy, the additional restrictions imposed will affect the delegation freedom, which goes against the original intention of free delegation.

In conclusion, the strategies of policy-on-blockchain and right-on-blockchain are difficult to achieve sufficiently flexible and scalable access control. We improve the above defects with the help of capability tokens and the strategy of delegation trajectory-on-blockchain. The proposed cross-domain delegation trajectory aggregation scheme provides support for forensic analysis. We only store the hash of the delegated trajectory on blockchain. The decision-making and analysis process are implemented without the blockchain. Our method does not involve the update of access control policies on blockchain, which brings a stronger scalability.

3. System Design

We propose a blockchain-based IoT cross-domain delegation access control method, which is called CDDAC. CDDAC uses the blockchain to ensure the reliability of the capability delegation trajectory in each alliance. To ensure the flexibility of delegation and reduce the complexity of smart contracts, we delegate the right of access decision-making to users and domain managers. The goal-directed logging and forensic analysis of the delegation trajectory provide more security while implementing access control policies.

CDDAC's architecture is shown in Figure 1. Similar to CapBAC [19], we use capability tokens to represent the access right to be delegated. Intra/cross-domain access requests from token owner will be centralized to the domain manager. After the legality verification and the access control policies decision, the delegation topology is generated and aggregated with the trajectory in the Delegation Trajectory Database (DTDB). Cross-domain access requests will be submitted to the managers of other domains; then, the procedure of cross-domain access will be completed. The policy changing is simplified to DTDB updating. The DTDB hash will be packaged and broadcast to all domain managers. After the transaction procedure, the smart contract with the hash will be redeployed to ensure the reliability of trajectory data in DTDB.

In this section, we will introduce the system design of CDDAC, including the capability token structure, domain manager, and delegation process.

3.1. Capability Token Structure. We design a capability token structure for CDDAC. The classic CapBAC uses nesting tokens to trace back the root token and verify its legitimacy to confirm whether the capability is valid. This design makes the size of the capability token increase with the depth of delegation, which is not scalable for IoT. Due to the size limit of smart contracts, nested tokens are not suitable for being used in the blockchain. We find that the token structure is essentially meant to save the delegation trajectory and confirm its legitimacy. However, retaining the delegator's token in each token is a waste of resources. When the delegation trajectory is extracted and its reliability is guaranteed, a nonnested capability token can be obtained. It will greatly reduce the size of tokens and the token processing overhead and make it possible to integrate with the blockchain. We propose the structure of CDDAC's capability token based on the above motivation.

The capability token is defined as

$$\begin{aligned} \mathrm{Cap}_{\mathrm{cross}} &= \{ID_A, ID_B, Cap_{\mathrm{root}}, \mathrm{Trace}, ET, C, \mathrm{Signature}_C\}, \\ \mathrm{Signature}_D &= f\left(ID_A, ID_B, \mathrm{Cap}_{\mathrm{root}}, ET, \mathrm{Trace}, C\right). \end{aligned} \tag{1}$$

(i) ID_A: *User A*'s identity document

(ii) ID_B: *User B*'s identity document

(iii) Cap_{root}: the token of the root owner with the capability

(iv) *Trace*: the delegation trajectory after *User A* adds *Node B*

(v) *ET*: the validity period of the capability

(vi) *C*: the blank bits containing context-related information, or whether the capability can be delegated

(vii) $Signature_C$: signature of capability token by domain manager

(viii) f: one-way hash function

In the capability token used by CDDAC, ID is used to identify the virtual/real identity of users and is used by the delegator to verify the token validity. Cap_{root} is the original capability token. The signature of the root user can be used to verify authenticity. *Trace* is the capability trajectory that the delegator knows, which is constructed by the delegator. The domain manager confirms the authenticity of *Trace* based on the records saved in the DTDB. It is worth noting that the *Trace* in a single capability token is incomplete. The complete delegation trajectory is aggregated by the domain manager based on the *Traces* in all capability tokens. *ET* records the validity period of the capability, which is used to judge the legality of the capability. *C* is used to save the context information, or to mark whether the capability can be delegated. $Signature_C$ is the digital signature of the domain manager, which means that the token has been received and the legality has been initially verified.

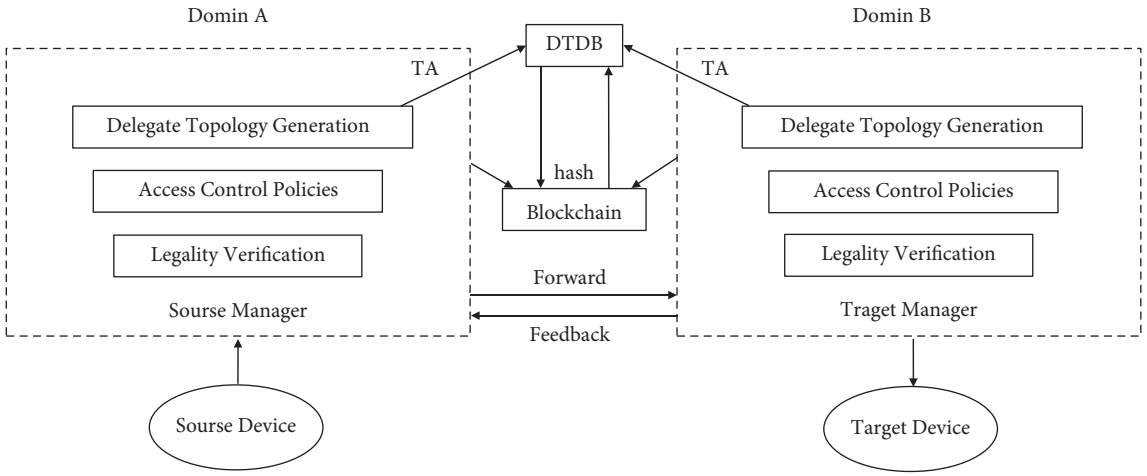

Figure 1: The basic architecture of CDDAC.

3.2. Domain Manager. The domain manager (DM) is responsible for collecting delegated trajectories, updating the DTDB, and updating the smart contract. All cross-domain access will be recorded and uploaded to DTDB by the domain manager, to guarantee the reliability of access with the help of blockchain.

3.3. Delegation Process. The delegate process of CDDAC is shown in Figure 2. First, the delegator sends a delegation request to the domain manager and presents the capability token constructed by itself. This step is to ensure that the domain manager can get a complete delegation trajectory. After the trajectory and root token are certified to be legal, it is deemed that the delegation meets the basic security requirements. The capability token will be digitally signed and transmitted to the delegator. Then, the domain manager will update the information in DTDB according to the delegation trajectory of the effective capability token. The delegated trajectory aggregation will be completed by the domain manager and uploaded to the DTDB. The DTDB returns the new *Hash* and redeploy the smart contract. Then, the access right delegation is completed.

3.4. Intra/Cross-Domain System Implementation

3.4.1. Intradomain System. The blockchain is not required for the intradomain access control system, but the delegation trajectory stored in DTDB is necessary. Figure 3 shows the delegated access control method in a single-domain system. The root token is issued by the cloud platform. To improve the scalability of tokens, we use a single-layer structure of capability tokens and adopt a weakly coupled central structure, which means that the domain manager only participates in the issuance procedure of the token, not in the actual use process of the capability. This method guarantees the freedom of delegation.

The delegation procedure is the same as described in Section 3.3. During the delegation process, all delegation trajectories will be aggregated and stored in DTDB. DM and DTDB record the delegation trajectory and do not evaluate the delegation accuracy. When the resource server (RS) receives an access request with a capability token, it needs to verify whether the token has been authenticated by the DM. If the user accesses RS for the first time, RS can request the DM to verify the security according to the current situation. For example, a new user shows a capability token and asks to open the classroom door at 0 : 00. RS judges that the behavior is abnormal according to the basic access control policies. Then, it queries the delegation trajectory, sends a warning to DM, and requests security analysis. RS can receive DM's instructions to deal with delegation changes, revocations, or other events. Context from IoT sensors can also be used for decision-making.

3.4.2. Cross-Domain System. Figure 4 shows a cross-domain access control system. We divide the cross-domain system into four layers according to their functions. The Delegation Layer is composed of users in multiple domains, including the subject and object of access control. The Access Control Layer consists of multiple-domain managers. In Figure 4, the domain managers of *Domain A* and *Domain B* are two domain servers (DS). This layer is mainly responsible for the collection and cross-domain access control of the multi-domain system. The Trajectory Storage Layer is the DTDB, which is responsible for the storage of delegate trajectories and the generation of new hash. The Blockchain Layer is the blockchain, which provides reliability guarantee for the delegation trajectory [20].

To make a final judgment on the legality, the root token must exist in each issued capability token. Therefore, cross-domain access requires root tokens in other domains, as the seed of the delegation chain. As shown in Figure 4, if a user in *Domain B* wants to access resources in *Domain A*, the resources in *Domain A* must delegate capability to *Domain B*. For example, if *Alice* issues the *Root Token* to *Bob*, *Bob* can access *Alice*'s resources in *Domain A* and continue to delegate the *Token αi*, so that other nodes in *Domain B* have the capability to access the resources. All delegation behavior described above should be recorded in DTDB.

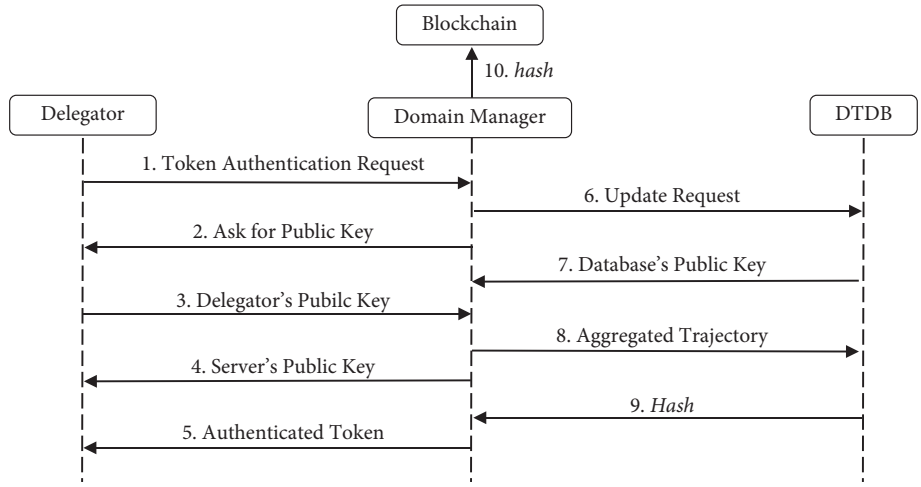

Figure 2: Delegation process of CDDAC.

The basic data structure in smart contract is a six-tuple$\langle \mathrm{ID}, Statue, URI, Hash, C, Signature_D \rangle$, where the following hold:

(i) ID: the identity of the domain manager

(ii) Statue: the statue of change, such as access recording or access, has been successful

(iii) *URI*: the storage location in DTDB

(iv) Hash: the hash value of DTDB

(v) *C*: the blank bits containing context-related information

(vi) $Signature_D : f \longrightarrow \mathrm{ID} \times \mathrm{Statue} \times URI \times \mathrm{Hash} \times C$ is a one-way hash function used as the signature of the domain manager

The trajectory modification behavior corresponds to the domain manager identity by *ID*. The $Signature_D$ also has the same effect. *Statue* is the statue of change, which is part of the goal-directed logging, to facilitate forensic analysis. *URI* indicates the data storage location in the DTDB. *Hash* is to confirm the integrity of DTDB, and it is also the main content of the smart contract. *C* can be used to record the context information such as time information.

4. Trajectory Aggregation and Forensic Analysis

In this section, we will introduce the multidomain trajectory aggregation method and the forensic analysis method in CDDAC.

4.1. Trajectory Aggregation. The delegation trajectory is written into the token by the delegator. Different branch nodes on the delegation chain do not have a comprehensive understanding of the delegation trajectory; therefore, the domain manager needs to extract and aggregate the delegation trajectory. We will introduce the workflow of delegation trajectory aggregation and goal-directed logging.

4.1.1. Delegation Trajectory Aggregation. As shown in Figure 5, *Users A–D* are delegated capability in turn. We find that the delegate trajectories from *Users A*, *B*, *C*, and *D* are not the same. They do not entirely contain each other. When *User B* delegates the capability to *User D*, he does not know that *User A* has delegated the capability to *User C*. The inherent information gap in the delegation process leads to the incompleteness of the delegation trajectory in a single capability token. Therefore, the domain manager needs to collect all the capability tokens, extract the topology of delegation trajectories, and aggregate them to obtain the complete delegation trajectory.

The delegation trajectory may cross multiple domains, and the domain managers may not know the identity of each user. Cross-domain access control also faces the challenge of devices shared by multiple domains, or the free devices. These make it difficult to obtain a complete cross-domain delegation trajectory. Therefore, the trajectories submitted by different domain managers should be aggregated twice for cross-domain delegation.

4.1.2. Goal-Directed Logging. Goal-directed logging mainly provides the ability to quickly respond to security incidents. When a security incident occurs, people usually hope to find vulnerabilities from the messy resource server logs. Additional log analysis [21] will extend the existence time of access control vulnerabilities, causing more serious damage to the system. In the meantime, relying on the access log that records the occurrence of access, the system has almost no dangerous warning functions. The dangerous behavior has occurred at least once when it is recorded. The importance of the goal-directed logging is thus reflected.

We propose a goal-directed logging for CDDAC. The delegation trajectory of the cross-domain delegation is saved in the DTDB. Figure 5 shows the workflow of delegation trajectory aggregation. The node structure of delegation trajectory in DTDB is [*Cap*, *User*, *EffectTime*, *AccessTime*]. Among them, [*Cap*] records the description of the capability. [*User*] records the user ID who has used this capability.

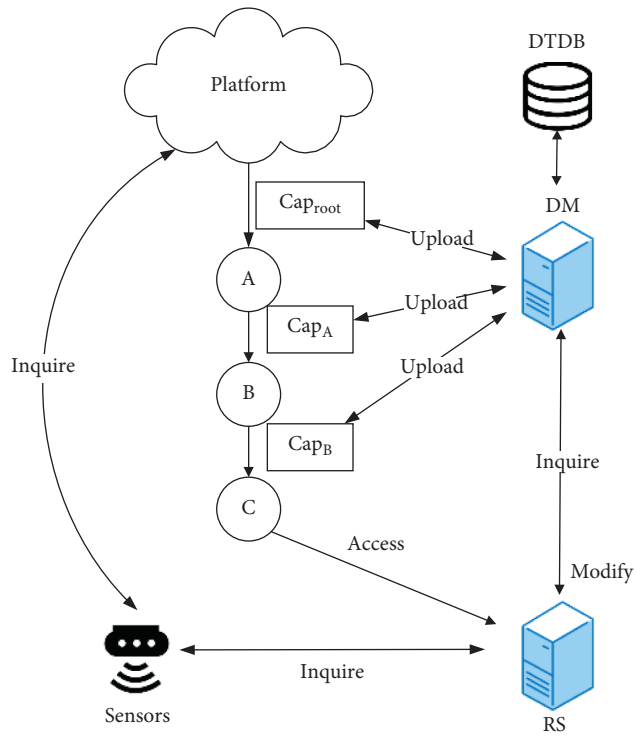

Figure 3: Delegated access control method in a single-domain system.

[*EffectTime*] records the validity period of the capability. [*AccessTime*] records the time when the user uses the capability. As each cross-domain access will be captured by the domain manager, all information needed can be obtained from the data packet. We use this method to achieve goal-directed logging of CDDAC.

4.2. Forensic Analysis. Due to the uncertainty of the capability flow, the cross-domain delegation systems are more likely to suffer security incidents caused by misallocation of capabilities. Since the capability delegation is determined by users, we cannot determine whether the delegation conforms to the principle of least privilege. It is also difficult to determine whether the capability delegation is under attack. The resource server can only detect the process legitimacy and passively provide capabilities recorded in the token. Therefore, the occurrence of security incidents is difficult to predict. Forensics analysis is particularly important to deal with such problems. CDDAC provides forensic analysis by establishing DTDB. In this section, we present the definition and examples of forensic analysis in CDDAC, including change warning and record analysis.

4.2.1. Definition. A stable access control scheme can be defined as a pair$\langle \Gamma, \Psi \rangle$, where $\Gamma$is a set of states and Ψ is the state-changing policies. Then, we can describe a forensic instance as follows:

$$F = < \gamma, \psi, p, q, \pi, L >, \quad (2)$$

where $\langle \gamma, \psi \rangle$ is a specific access control system, $\gamma \in \Gamma$ is the system state and $\psi \in \Psi$ is the rule caused the state change; p is the known past state of the system; q is a query by which we can get the information of the past state; π is the result of a forensic analysis; and L is the system logs.

In fact, we can get the past system state from p, $p \longrightarrow_{\psi}^{*} \gamma$. Then, we can establish a state-changing sequence $\gamma_1 \longrightarrow_{\psi} \gamma_2 \longrightarrow \ldots \longrightarrow_{\psi}^{*} \gamma$ as the evidence chain or the backtracking of errors. We use DTDB to catch the delegation behaviors as p and save the access logs as L in CDDAC. The DTDB supports multiple query methods q, constructs the capability trajectory, and finally gives a forensic analysis result π to users.

4.2.2. Change Warning. In a stable access control system, there will be fewer transfers, revocations, and modifications to sensitive capabilities. In other words, any changes to sensitive capabilities should be considered as threats, and administrator should be warned. The delegation system transfers the right of delegation capabilities to users, and the occasional misjudgment of the node brings a number of risks to the system. We believe that we should pay more attention to the change of sensitive capabilities, while ensuring the scalability of the system. Therefore, we reserve the user's delegation right for all capabilities, and the verification right of sensitive capabilities is given to the domain administrator. The specific approach is as follows. First, we divide the set of sensitive capabilities. For the sensitive delegation trajectory that we want to closely monitor, when the number of nodes increases (only the increase of nodes will bring a threat to a stable system), an early warning will be issued to

administrator, and the legitimacy of the completed delegation trajectory will be checked. If it does not meet the access control policies, we can revoke the capability at the first time. We use this method to achieve change warning in CDDAC.

4.2.3. Suspect Analysis. The suspicion analysis is mainly for the policy allocation errors. Thanks to the change warning mechanism, we have eliminated the misallocation of capabilities due to user's decision-making errors. However, the following situation is still possible: a "security" capability actually has the danger to cause a security incident, which is called the capability configuration error. Capability configuration errors often occur in the actual operation of system [22,23]. We cannot completely avoid the occurrence of security incidents. What is important is how to accurately and quickly conduct suspicion analysis.

In CDDAC, we use the *AccessTime* stored in the delegation trajectory for suspicion exclusion/conviction. *AccessTime* plays a role in goal-directed logging when security incidents occur. Goal-directed logging only records things that may be useful for forensic analysis. It has been expected to greatly reduce the size of log records that require forensic query [24]. In DTDB, the *AccessTime* stored by each node records the operation type and access time, which are the core information required for forensic analysis. *AccessTime* is automatically generated in delegated trajectory aggregation, which avoids additional data reduction work. We can easily record the location and time of a security incident and directly query the *Cap* and *AccessTime* stored in the DTDB. Then, we can evaluate whether the node may have the capability, whether it does possess the capability, and whether the capability is used. We can exclude users from suspicion, or convict users, in need of different situations.

5. Evaluation

We conduct experiments on the performance of CDDAC and compare it with existing works.

5.1. Implementation. Domain managers are two laptops with the following configurations: the CPU is 1.6 GHz Intel Core i5 (4 cores), the RAM is 8 GB, and the operating system is Ubuntu 16.04. Each laptop manages 2 to 8 Raspberry PI 4 Model B as IoT devices. Redis 5.0.8 is used to manage the delegation trajectory data. Ropsten is used as the blockchain on the public network, which is the Ethereum's official public test network [25]. We use it to evaluate the performance of our approach.

5.2. Performance

5.2.1. Token Processing Overhead. To evaluate the token efficiency of CDDAC, we randomly generate capability tokens with different delegation depths, to evaluate the token processing overhead of existing works. We define the token processing overhead as the time cost of obtaining the required information from the capability token. Because of the different token structures and processing flows of each scheme, the composition of the token processing overhead is different. In CapBAC, the token processing overhead is the time, which can be expressed as $\sum_{n=1}^{N}(T_n^{\text{Decryption}} + T_{\text{n}}^{\text{Verification}})$, where n is the number of token layers. In BlendCAC, the token processing overhead mainly includes the time for querying capability data from the smart contract and the time for parsing JSON data from the request. In CDDAC, the total processing time is composed of the decryption time, the verification time, the smart contract running time, and DTDB querying time. Token efficiency, which is the token processing overhead, represents the minimum resource consumption when the capability is used. High-efficiency tokens are accompanied by lower resource overhead, which is more suitable for lightweight IoT devices. It can be known from the calculation method that the token processing overhead is associated with the token layers and the number of nodes. Although we randomly generated the delegation trajectories, there is only one node at each layer of the capability token. We use this method to control irrelevant variables. RSA1024 is used for encryption and signature of all the three schemes. The experimental result is shown in Figure 6.

We compare the token processing overhead of CDDAC with CapBAC [19] and BlendCAC [14]. CDDAC uses the proposed single-layer capability tokens, CapBAC uses classic nested capability tokens, and BlendCAC uses capability tokens based on smart contract. The size of token increases with the number of layers in the three approaches, but the token processing procedure is different, which makes the difference of token processing overhead. Due to its layer-by-layer decryption design, the processing time of CapBAC increases greatly with the delegation depth, which makes it difficult to be used in large-scale systems. BlendCAC adopts the right-on-blockchain strategy. The key value of the token is extracted by the smart contract, so the token processing speed has little to do with the depth of delegation, which is about 200 ms. CDDAC uses the single-layer capability token, which prevents the excessive token complexity. The single verification greatly reduces the token processing overhead, so the processing time is always kept to a minimum. It shows that CDDAC's token design is more suitable for lightweight IoT devices.

5.2.2. Smart Contract Decision-Making Overhead. The decision-making overhead of a strategy is defined as the smart contract running time, and its changing trend is the manifestation of the scheme scalability. The decision-making cost of a scalable scheme should not increase drastically as the number of access control policies increases. We correspond policy entry to delegated behavior and evaluate the smart contract running time of the policy-on-blockchain, the right-on-blockchain, and the trajectory-on-blockchain strategies (which is used by CDDAC). The experimental result is shown in Figure 7.

It can be seen that the decision-making overhead of existing blockchain-based approaches increases as the smart contract complexity increases. Since the policy-on-blockchain

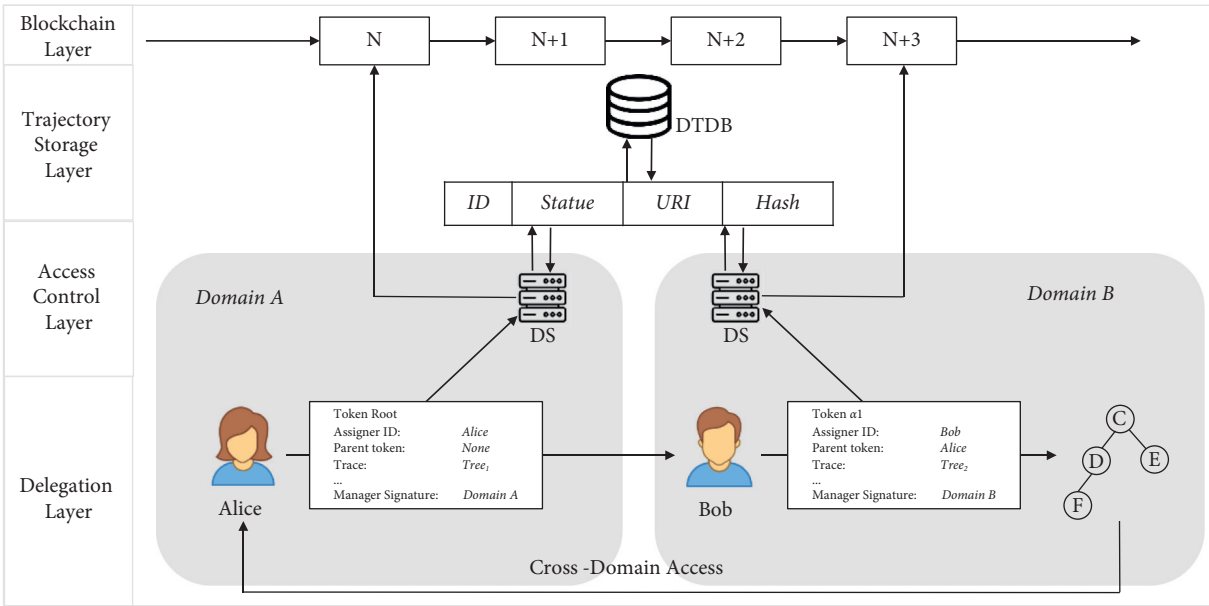

Figure 4: Delegated access control method in cross-domain system.

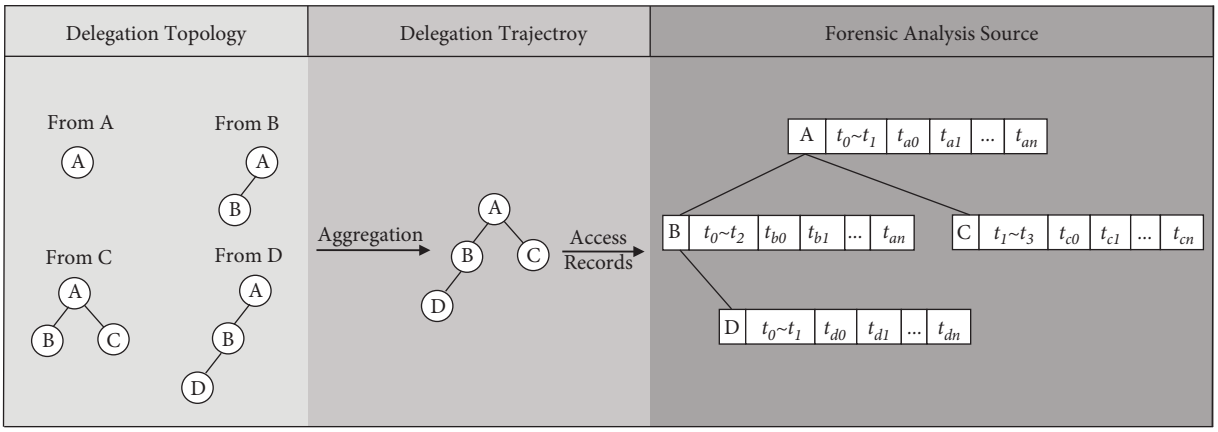

Figure 5: Workflow of delegation trajectory aggregation.

strategy writes access control policies by the form of smart contract, the complexity of smart contract will increase with the scale of policies, and its decision-making time will increase linearly with the number of policies, with the largest slope. The right-on-blockchain strategy extracts token attributes and checks whether the attributes exist in the smart contract, to make an access control decision. The decision-making time also expands with the expansion of the token scale, but the slope is smaller than the strategic plan. In trajectory-on-blockchain strategy, after obtaining the token, an inquiry will be initiated to DTDB, then decision will be made after verifying the authenticity. This procedure has little relevance to the number of policies, which brings an excellent scalability to CDDAC.

5.2.3. Policy Changing Overhead. Policy changing is an essential feature in an IoT access control system, which is always completed as a transaction in blockchain-based schemes. It is defined as the transaction time. We evaluate the policy changing overhead of three strategies. The result is shown in Figure 8.

The policy changing of the three strategies all involve the changing of smart contract. It will cost a long time to redeploy the smart contract, which brings the large cost of policy changing. The average cost is about 12 seconds. In policy-on-blockchain strategy, smart contracts need to be rewritten and deployed once a policy is changed, which enhance the policy changing complexity. The right-on-blockchain strategy modifies the token parameter set in the smart contract, to change access control policies. There is no need to rewrite the smart contract logic, so the policy is easier to change. The trajectory-on-blockchain strategy changes the content of DTDB, generates a new hash, and publishes it to a new smart contract. It has the lowest policy changing complexity, but it is still limited by the redeployment overhead of smart contract. When the

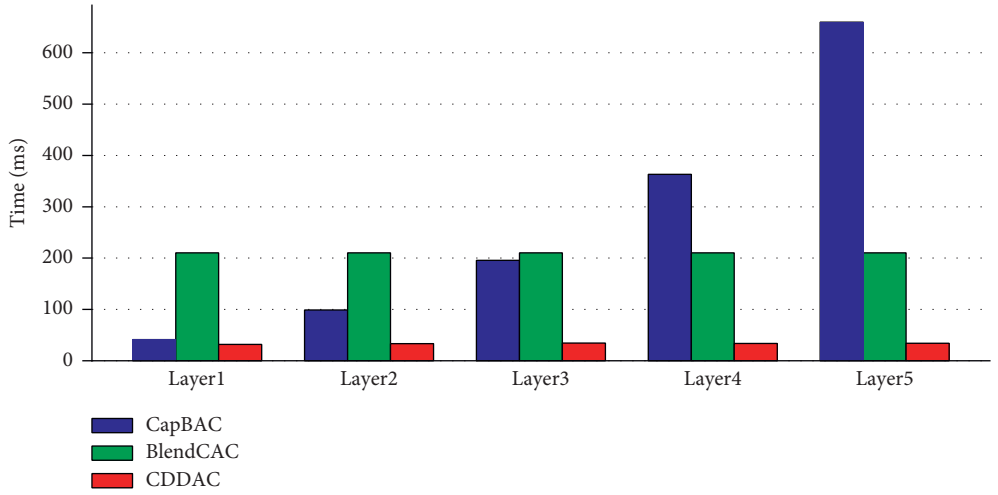

Figure 6: The token processing overhead of the three approaches under different capability token layers.

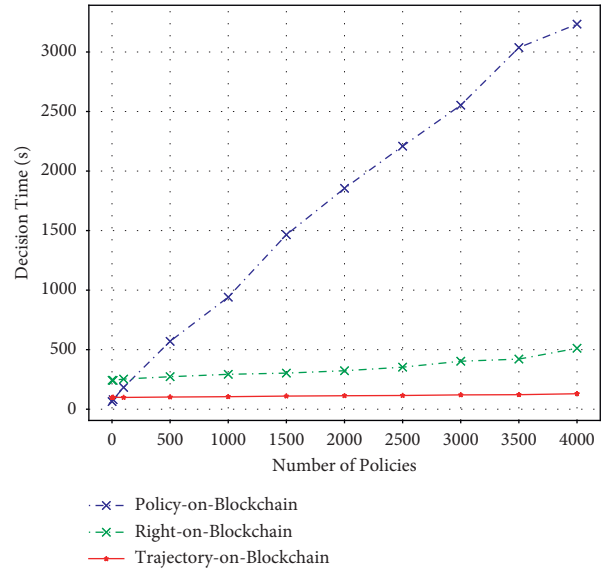

Figure 7: The smart contract decision-making overhead of the three strategies under different number of access control policies.

smart contract redeployment overhead is shortened, the policy changing overhead of CDDAC will gain a clear advantage.

5.3. Discussion. Benefiting from the design of extracting delegation trajectory, the complexity of CDDAC's capability token has been simplified, which greatly reduces the token processing overhead. The design of policy-on-blockchain and right-on-blockchain has been cancelled, which reduces the complexity of smart contracts, reduces the cost of decision-making, and ensures the high scalability of CDDAC. In the meantime, the policy changing process of CDDAC has been simplified. Although the test on the public blockchain network does not show obvious advantages, when the redeployment time of smart contracts is shortened, its efficiency will be reflected. The test of CDDAC has proved its significant progress in token processing overhead and smart contract decision-making overhead. CDDAC can maintain a token processing speed about 30 ms and a decision-making speed of about 110 ms. In short, as a delegation-oriented IoT cross-domain access control system, the lightweight and scalability of CDDAC show obvious advantages compared with existing works.

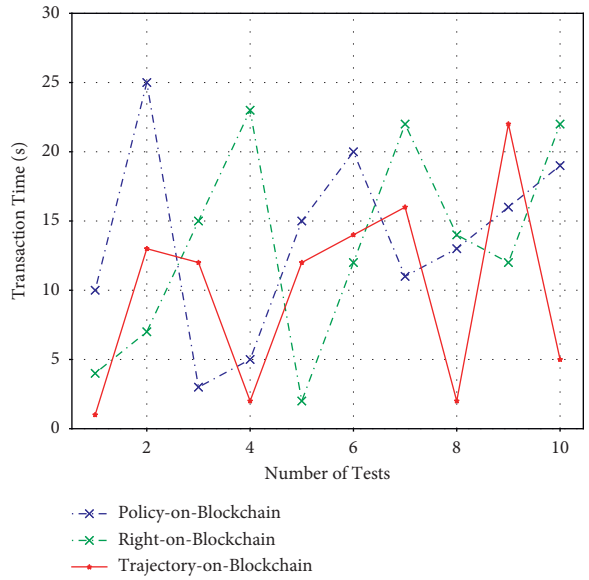

Figure 8: The policy changing overhead of the three strategies.

6. Conclusion

In this paper, we introduce CDDAC, a blockchain-based IoT cross-domain delegation access control method. The capability token structure of CDDAC is more suitable for lightweight devices. The adopted trajectory-on-blockchain strategy greatly enhances the scalability of the system and has a simpler policy changing process. We propose a multidomain delegation trajectory aggregation mechanism to support forensic analysis of intra/cross-domain delegation, which is beneficial to the confirmation and accountability of suspicion after malicious behavior occurs. We evaluate the performance of CDDAC in the Ropsten. The results show that CDDAC has the advantages of lightweight and scalability compared with similar research. In the future, we will design a privacy protection mechanism in the process of cross-domain delegation based on the idea of CDDAC and combine CDDAC with the alliance chain to achieve a more complete cross-domain delegation access control system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2018YFB2100400), National Science Foundation of China (no. 61872100), Industrial Internet Innovation and Development Project of China (2019), State Grid Corporation of China Co., Ltd., Technology Project (no. 5700–202019187A-0-0-00), and Guangxi Key Laboratory of Cryptography and Information Security (no. GXIS202119).

References

[1] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[2] W. Tärneberg, V. Chandrasekaran, and M. Humphrey, "Experiences creating a framework for smart traffic control using aws iot," in *Proceedings of the 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, pp. 63–69, Shanghai, China, December 2019.

[3] S. Zheng, N. Apthorpe, M. Chetty, and N. Feamster, "User perceptions of smart home IoT privacy," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, pp. 1–20, 2018.

[4] T. H. Payne, D. E. Detmer, J. C. Wyatt, and I. E. Buchan, "National-scale clinical information exchange in the United Kingdom: lessons for the United States," *Journal of the American Medical Informatics Association*, vol. 18, no. 1, pp. 91–98, 2011.

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 103–114, Redmond, WA, USA, November 2009.

[6] Q. Alam, M. Alani, G. Ali, and F. Azim, "Towards a formal framework for cross domain access control," *International Information Institute (Tokyo). Information*, vol. 15, no. 10, p. 4303, 2012.

[7] J. Sun J and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 6, pp. 754–764, 2009.

[8] M. Alam, X. Zhang, K. Khan, and G. Ali, "XDAuth: a scalable and lightweight framework for cross domain access control and delegation," in *Proceedings of the 16th ACM symposium*

on Access control models and technologies, pp. 31–40, Innsbruck, Austria, June 2011.

[9] S. Sheikh and A. K. Chaturvedi, "Analysis of sensitive data security on trusted third party in cloud computing," *management*, vol. 17, p. 18, 2014.

[10] Q. Alam, S. Tabbasum, A. Malik, and M. Alam, "Formal verification of the xDAuth protocol," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1956–1969, 2016.

[11] O. Novo, "Blockchain meets IoT: an architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[12] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[13] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport: a blockchain-based trust framework for collaborative internet-of-things," in *Proceedings of the 24th ACM symposium on access control models and technologies*, pp. 83–92, Toronto, Canada, May 2019.

[14] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Blendcac: a blockchain-enabled decentralized capability-based access control for iots," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1027–1034, Halifax, Canada, August 2018.

[15] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Exploiting smart contracts for capability-based access control in the Internet of Things," *Sensors*, vol. 20, no. 6, p. 1793, 2020.

[16] A. Gauhar, N. Ahmad, Y. Cao et al., "xDBAuth: blockchain based cross domain authentication and authorization framework for internet of things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020.

[17] B. Yuan, Y. Jia, L. Xing, D. Zhao, X. Wang, and Y. Zhang, "Shattered chain of trust: understanding security risks in cross-cloud iot access delegation," in *Proceedings of the 29th ${$USENIX$}$ Security Symposium (${$USENIX$}$ Security 20)*, pp. 1183–1200, Boston, MA, USA, August 2020.

[18] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proceedings of the IFIP international conference on distributed applications and interoperable systems*, pp. 206–220, Neuchâtel, Switzerland, June 2017.

[19] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5–6, pp. 1189–1205, 2013.

[20] J. Chen, W. Gan, M. Hu et al.,et al. "On the construction of a post-quantum blockchain for smart city," *Journal of Information Security and Applications*, vol. 58, Article ID 102780, 2021.

[21] B. J. Jansen, ""Search log analysis: what it is, what's been done, how to do it," *Library & information science research*, vol. 28, no. 3, pp. 407–432, 2006.

[22] T. Das, R. Bhagwan, and P. Naldurg, "Baaz: a system for detecting access control misconfigurations," *USENIX Security Symposium*, vol. 17, pp. 161–176, 2010.

[23] T. Xu, H. M. Naing, L. Lu, and Y. Zhou, "How do system administrators resolve access-denied issues in the real world?" in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 348–361, Denver Colorado USA, May 2017.

[24] N. Juma, X. Huang, and M. Tripunitara, "Forensic analysis in access control: foundations and a case-study from practice," ", in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1533–1550, Virtual Event USA, November 2020.

[25] Ethereum/ropsten. ethereum, 2021, https://github.com/ethereum/ropsten.