processes

MDPI

Article

IoT Access Control Model Based on Blockchain and Trusted Execution Environment

Weijin Jiang [1,2,3], En Li [1,*], Wenying Zhou [1], Ying Yang [1] and Tiantian Luo [1]

1 School of Computer Science, Hunan University of Technology and Business, Changsha 410205, China
2 School of Computer Science and Engineering, Hunan University of Information Technology, Changsha 410151, China
3 Tan Zhou Academy, Xiangtan Institute of Technology, Xiangtan 411100, China
* Correspondence: en.li@foxmail.com

Abstract: With the application and popularization of the Internet of Things (IoT), while the IoT devices bring us intelligence and convenience, the privacy protection issue has gradually attracted people's attention. Access control technology is one of the important methods to protect privacy. However, the existing IoT access control technologies have extensive problems such as coarse-grainedness, weak auditability, lack of access process control, and excessive privileges, which make the security and privacy of our IoT devices face great threats. Based on this, a blockchain-based and encrypted currency-based access control model CcBAC supported by Trusted Execution Environment (TEE) technology is proposed, which can provide fine-graininess, strong auditability, and access procedure control for the Internet of Things. In this study, the technical principle, characteristics, and research status of the control model are introduced, and the framework of the CcBAC model is expounded in detail and formally defined. Moreover, the functions in the model are described in detail, and a specific access control process in general scenarios is presented for the model. Finally, the practicability of this model is verified through theoretical analysis and experimental evaluation, which proves that this model not only enables resource owners to fully control the access to their resources, but also takes into account the fine-graininess and auditable access control.

Keywords: access control; Internet of Things; blockchain; trusted execution environment; cryptocurrencies

check for updates

Citation: Jiang, W.; Li, E.; Zhou, W.; Yang, Y.; Luo, T. IoT Access Control Model Based on Blockchain and Trusted Execution Environment. *Processes* **2023**, *11*, 723. https://doi.org/10.3390/pr11030723

Academic Editors: Danyu Bai, Xin Chen, Dehua Xu and Jedrzej Musial

Received: 22 December 2022
Revised: 9 February 2023
Accepted: 22 February 2023
Published: 28 February 2023

1. Introduction

With the rapid development of the Internet of Things (IoT), IoT devices are becoming smarter and stronger, providing great convenience to people [1]. The popularity of new Internet of Things technology will be the Internet of Things. The application has expanded to a wider range of areas, such as smart homes, industrial Internet of Things, smart medical care, smart cities, smart grids, etc., [2]. However, with the rapid advancement of this process, some potential IoT security issues have gradually appeared [3]. The Internet of Things equipment is not only facing security issues such as software and hardware security and blurring in network protection border, but also new security risks [4] such as springboard invasion, unauthorized access, eavesdropping, data loss, etc. It constitutes a serious threat to users' security and privacy.

Access control is a technology that manages the identity verification and authorization of access, retrieval, and operation of resources, so that resources can be used within a legal range or be restricted in use. It is an important measure for maintaining network security and data security [5]. Traditional access control models are based on centralized authorization decision entities, which make access control decisions based on access control policies and other attribute information. With the deepening of the Internet of Things in the life field, higher requirements have been put forward for the protection of data privacy and personal privacy information [6]. However, each access request points to the same central trusted entity, which stores all information and makes all decisions based on the stored information. This is technically insecure and requires security guarantees from legal aspects outside of technology. In recent years, frequent privacy leaks, such as the information leaks of South Korea's three major credit card companies and the iCloud cloud system vulnerability storm, have raised questions about the credibility of central trusted entities [7]. Therefore, it is necessary to design a reliable and flexible distributed access control scheme for the Internet of Things.

Blockchain, as a decentralized and distributed technology, offers potential solutions. With blockchain's capability to provide decentralized trustworthy storage [8], the access control policies and records in a distributed system can be securely stored on the blockchain without tampering, and the system is also equipped with strong auditability [9]. Furthermore, the emergence of smart contracts expands the ability of blockchain for various decentralized applications [10,11]. Thus, blockchain can act as a secure and trustworthy decentralized platform, providing strong traceability and verifiability for IoT access control.

In recent years, more and more work has been focused on integrating blockchain technology with access control to improve the security of the Internet of Things. Rifi et al. [12] proposed a smart home scenario where the capabilities of ordinary sensors in a smart building are limited and cannot be directly connected to the blockchain, thus a powerful home gateway is installed with a blockchain client to connect to the blockchain, managing all IoT devices in the smart building. Jemel et al. [13] used blockchain for user legitimacy checks in access control. Ying et al. [14] proposed a blockchain-based access control architecture that stores access control policies on the blockchain and manages them through blockchain transactions. Xu et al. [15] proposed an identity-based robust token management policy that utilizes blockchain to construct a decentralized access control policy, with a granularity equivalent to that of Linux. However, existing works are mostly simple combinations of blockchain and traditional access control, and cannot provide finer-grained access control for the IoT. In addition, blockchain's trusted computing capability is also worth applying besides its trusted storage.

To address the challenges of distributed and heterogeneous nature of the Internet of Things (IoT), this paper presents a fine-grained access control mechanism, combining blockchain and TEE technology, to overcome the commonly existing coarse-grained access policies, weak auditability, and lack of access process control in the IoT environment, ensuring secure access authorization of IoT resources. Our main contributions are summarized as follows:

(1) We propose a Cryptocurrency-Based Access Control (CcBAC) model based on the cryptocurrency Ccoin. Specifically, we improve the traditional Policy-Based Access Control (PBAC) to combine with blockchain to accommodate the distributed services of the Internet of Things, thereby preventing the witch attack. The access policy is carried by the blockchain and executed through smart contracts, realizing the automation of policy judgment. The Tendermint-BFT consensus mechanism is adopted to suit the limited Internet of Things devices, thus improving efficiency.

(2) Considering that the blockchain's transaction data structure can only handle fixed transaction data, making it inconvenient to store fine-grained access control policies, we have developed the UnRedeemed Policy Output (URPO) model. It expands the transaction structure to include keys and fields, enabling secure storage of flexible access policies at different granularity levels, achieving the goal of fine-grained, dynamically managed policies, convenient system management, and user-friendly access.

(3) In order to ensure secure execution of off-chain access policies, we have implemented a Reliable Access Control Object (RACO) using a Trusted Execution Environment (TEE). By utilizing the tamper-proof processing environment of TEE to host RACO and other security software, trust is extended from the chain to off-chain, effectively controlling the access process and preventing excessive privileges.

The remaining structure of this paper is as follows: in Section 2, we introduce the related works. In Section 3, we introduce an IoT access control model using blockchain and TEE. In Section 4, we show the implementation and working process of the model.

Finally, in Section 5, we conduct the experimental evaluation and comparative analysis. In Section 6, we present the conclusions and our future work.

2. Related Work

2.1. Internet of Things and Access Control

As the development of Internet of Things (IoT) technology and applications continues, the evolution of IoT from an early logistics network utilizing RFID technology to a connected smart earth has also brought advancements in access control in IoT environments. The traditional access control methods used in IoT include role-based access control (RBAC), attribute-based access control (ABAC), and capability-based access control (CapBAC). RBAC was proposed before the emergence of IoT and was originally aimed at solving the access control problems of large enterprise-level systems. RBAC associates roles with a set of permissions and users obtain corresponding permissions based on the roles assigned by the system. With the development of IoT, scholars have applied RBAC to IoT access control, which can support the scalability [16], cross-domain access control [17], and device heterogeneity [18,19] of IoT environments. However, as a static access control method, RBAC cannot pre-set the (user, role) and (role, permission) relationships, and cannot solve the problem of dynamic node access in IoT.

Unlike RBAC, which requires pre-setting of correspondences such as (role, permission) by administrators, ABAC is a model supporting dynamic access policies, which defines access policies based on attributes [20]. Because attributes are inherent characteristics of participants, this model can easily generate attribute sets without manual input, and quickly determine the relationship between attributes and permissions. ABAC was introduced in the Internet of Things, reducing the number of rules that could cause role explosion and solving the problem of RBAC model in highly decentralized IoT environments [21]. Reference [22] developed an attribute-based access control (ABAC) model on the AWS IoT platform, integrating its existing capabilities and introducing new attributes and attribute-based policies for IoT entities to enable expressive access control in AWS IoT. However, as the number of devices increases, the effort to develop a unified access control policy for different domains will significantly increase the workload of policy management, making attribute-based ABAC models unsuitable for large-scale, distributed IoT.

CapAC is an implementation of the access control matrix (ACM) model, where each subject is associated with a capability list that records the access permissions of the subject towards other objects. Sheng et al. [23] built a capability-based Internet of Things (IoT) access control architecture based on capabilities, context, and elliptic cryptography. Shigenari et al. [24] adopted the CapBAC (capability-based access control) model to secure IoT by studying the illegal information flow relationships on objects and devices of three types: sensors, actuators, and mixed devices, and designed an information flow control method based on these relationships. To avoid the single point of failure problem caused by using centralized servers, Dina et al. [25] proposed a distributed capability-based access control (DCapBAC) for IoT environments, using a community-based structure to define access rights in distributed IoT environments. This allows intelligent objects with sufficient resources in a community of intelligent objects sharing a common task to evaluate access permissions based on community rules on behalf of intelligent objects with resource constraints. However, IoT devices are vulnerable to attacks by malicious users due to their weak computing and storage capabilities, and cannot serve as a secure decision-making entity.

Besides the above three commonly used access control models, PBAC (policy-based access control) model is a new type of access control model [26], which combines the attributes of resources, external environment, and requesters, and the specific situational status of the access request with a set of rules to determine whether the organizational policy allows authorization to access based on these situational and attribute conditions. The National Institute of Standards and Technology (NIST) in the report "A SURVEY OF ACCESS CONTROL MODELS" pointed out that PBAC is an emerging model that satisfies abstract policy and concrete access control requirements. PBAC model includes using attributes and roles to determine access privileges. PBAC even goes beyond Attribute-Based Access Control (ABAC) to meet the fast and ever-changing access demands [27]. In addition to defining the attributes and roles of the accessor, PBAC can also refine the granularity of access policies to specific resources, access time, and location, etc., which is not available in other access control policies. This paper utilizes the fine-grained characteristics of PBAC model, incorporating blockchain technology and trusted execution environment, to solve the problem of easily attacked IoT devices at the access control level.

2.2. Blockchain-Based Internet of Things Access Control

Blockchain is a decentralized, distributed technology based on cryptography algorithms, and it is a peer-to-peer distributed ledger technology, a shared database technology on the Internet. Technologically, blockchain solves the security problems brought by centralized models based on trust, it guarantees the safe transfer of value based on cryptography algorithms, it ensures the traceability and non-tampering of data based on hash chains and timestamp mechanisms, it guarantees the consistency of block data between nodes based on consensus algorithms, and it ensures the programmable intelligence contract based on automated script codes and Turing-complete virtual machines. In recent years, blockchain technology has expanded from the financial sector to the Internet of Things (IoT) sector. One of the main applications is to use it for IoT access control, replacing the central trustworthy entity of IoT access control.

When blockchain technology is combined with the Internet of Things (IoT), access control, as one of the key technologies for IoT data protection, becomes the main combined area. Currently, there are two ways to combine the two: one is to combine blockchain technology with existing IoT access control models, where blockchain acts as a trusted entity for the existing access control model, and the other is to propose an IoT access control model relying solely on blockchain technology. Blockchain serves as a trusted entity while designing access control methods based on transactions or smart contracts, leveraging the features of blockchain. Cruz, Paul et al. [28] use blockchain to solve the problem of verifying the authenticity of user roles in cross-organizational RBAC by using blockchain as a trusted platform and creating and modifying users and their attributes through smart contracts. Maesa et al. [29] used blockchain technology to extend the ABAC, but only used blockchain to store access policies. Rouhani et al. [30] used blockchain technology to implement an attribute-based access control system, where blockchain mainly provides auditability, ensuring transparency between the requestor and the resource owner. Alansari et al. [31] used blockchain to store access control policies and user attributes while the computationally intensive part was executed in secure hardware Intel SGX outside the chain, and blockchain only acted as a trusted platform to prevent data tampering. However, the limitations of these methods lie in the limited expression capabilities of the access control policies and the inability to effectively address the issue of coarse-grained access control policies, resulting in difficulty to deal with the phenomenon of excessive privileges.

2.3. Trusted Execution Environment and Blockchain

In blockchain applications, methods to enhance privacy protection performance while maintaining data availability mainly fall into two categories. One is to use cryptographic techniques to apply encrypted data, such as homomorphic encryption technology, which can complete data usage without accessing the plaintext information of the data. The second solution is to transfer some computational requirements to a secure environment outside the chain, where the typical representation is a trusted execution environment. Unlike pure cryptographic methods, the solution of executing computations outside the chain can alleviate the performance problem of insufficient calculation efficiency on the chain. However, it is currently difficult to load all the modules of the blockchain into a trusted execution environment [32]. Therefore, transferring part of the calculation steps off-chain, as proposed in this paper, to a trusted execution environment is a more reasonable choice under the current level of technology maturity, thereby extending trust from the chain to the off-chain.

Enkhtaivan et al. [33] proposed a bidding scheme that utilizes trustworthy hardware and blockchain to enhance bidder privacy and auction accuracy, incorporating group signature algorithms in the blockchain to provide anonymity, and utilizing remote authentication based on hardware-trusted computing environment to ensure the correct identification of the winning bidder. Ayoade et al. [34] designed a distributed data management system for IoT devices based on blockchain technology, storing hashed data hashes in the blockchain and using TEE only to store raw data. Liang et al. [35] integrated TEE with blockchain technology to implement a reward mechanism for data providers in Crowdsensing applications and solve the problem of anonymity for malicious requestors. Cheng et al. [36] proposed storing critical data of lightweight mobile clients in the Intel SGX environment to enhance client data security and privacy. Enkhtaivan et al. [37] proposed using TEE technology to coordinate data between IoT systems and blockchain systems, ensuring the completeness and confidentiality of IoT device-generated data before it is uploaded to the blockchain. In these schemes, TEE is more often used as a trustworthy storage solution off-chain, ignoring its secure computing capabilities.

3. CcBAC Access Control Model

The main design idea of the model proposed in this paper is to combine the trusted execution environment with blockchain technology and implement permission determination, transfer, and execution through the deployment of smart contracts. Access permission is initially defined by the resource owner through a transaction, and all transactions of the operating permission policy are published on the blockchain. In the framework, the blockchain is viewed as a reliable database, storing requestor's access control policies in the form of transactions. Meanwhile, cryptographic currency is introduced as an index of the policy, representing the authorization of the creator of the transaction to the recipient. In the process of permission transfer, the access control contract determines the operation on the resource according to the access control policy carried by the cryptocurrency.

In this model, access policies are defined by resource owners in Ccoin according to their own needs, and different granularity values correspond to the *who*, *what*, *where*, *when*, and *how* fields. The access policy specifies legitimate accesses through the "*who*" field, which can be a single object or group meeting conditions. Before Ccoin is exchanged for resource access rights, the resource owner can update the access policy or revoke the Ccoin, achieving fine-grained control over resource access rights. This method has four characteristics:

(1) Only the owner has the authority to create, update, and revoke Ccoin;
(2) The current Ccoin holder can transfer Ccoin freely to other participants;
(3) Ccoin exchange is only permitted when conditions defined in the access policy are met, and access activities must strictly follow the policy;
(4) All access activities are recorded on the chain for audit.

The model entities include the resource owner (RO), the resource user (RU), the permission holder (PH), and the reliable access control object (RACO), and all participants in the model are identified for interaction using addresses. Each participant has a wallet that stores the cryptocurrency Ccoin representing the access permission. The transactions involved in this model mainly include the following three steps, as shown in Figure 1:

Authorization process: The resource owner RO drafts the access policy and creates Ccoin, and then transfers Ccoin to the permission transmitter PT, as shown in process ① in Figure 1.

Transfer process: If the permission holder PH is not the final user, Ccoin needs to be further transferred to other participants, as shown in process ② in Figure 1.

Exchange process: If the current Ccoin holder is the resource requester RU, Ccoin needs to be transferred to the access control object RACO to exchange the resource operation permission, as shown in process ③ in Figure 1. RACO will verify whether it meets the access policy conditions and monitor the access process.

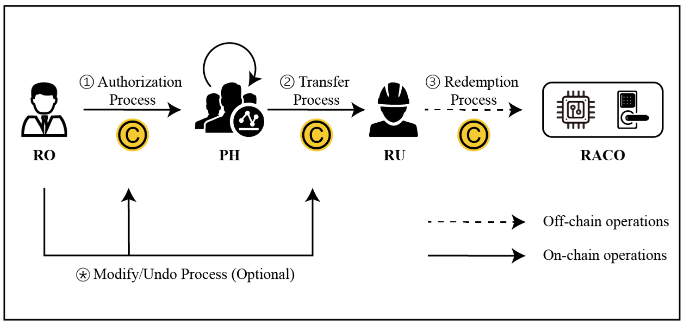


Figure 1. The access control model based on Ccoin.

4. CcBAC System Implementation

4.1. Basic Elements and Functions

Definition 1. *The cryptocurrency of the representative access permissions as Ccoin c = (tokenId, owner, holder, device, policy, timestamp, isValid).*

Table 1 shows the attributes of Ccoin, where *owner* and *holder* represent the owner of Ccoin and the public key address of the current holder; *device* represents the address of the device, which is also the address of RACO; *tokenId* is the only number of Ccoin, and it is required not to be repeated; *policy* stands for the access control strategy defined who can perform and what operations can be performed.; *timestamp* is a timestamp that indicates the time of creating Ccoin; *isValid* represents whether the current status of Ccoin is valid. If it is valid, *isValid* = TRUE, otherwise, *isValid* = false.

Table 1. Attributes of Ccoin.

Type	Name of Variable	Defined by ERC-721
uint256	*tokenId*	Yes
address	*owner*	Yes
address	*holder*	No
address	*device*	No
json	*policy*	No
uint256	*timestamp*	No
bool	*isValid*	No

CcABC utilizes the blockchain to ensure the security of data storage and atomic transmission of data, with the consistency of the ledger maintained by consensus participant nodes. Ccoin is stored on the blockchain, and all operations on Ccoin are executed on the chain by sending messages to all blockchain nodes via the function caller, such as *createCcoin*, *transferFrom*, *updatePolicy*, *revokeCcoin*, and *redeemCcoin*. The function caller must sign the message with a key to ensure the authenticity of the message. In CcBAC, the message format from a calling party *pki* is as follows:

$$msg : \left[tokenId, op, \{policy\}, \{pk_j\}\right]_{\sigma_{pk_i}} \tag{1}$$

where *op* denotes the operation code of the function. The parameters in curly braces are optional. If *op* = *createCcoin* or *updatePolicy*, then {*policy*} must be a valid policy. If *op* = *transferFrom*, then $\{pk_j\}$ is the new recipient's public key. σ_{pk_i} represents the key signature of the function caller's pk_i.

Table 2 shows the main functions and events defined in the CcBAC model interface, where *createCcoin*, *transferFrom*, *updatePolicy*, *revokeCcoin*, and *redeemCcoin* are used to operate Ccoin. The pseudo-code of the functions is shown in Algorithms 1–5. The *policyCheck* is called by the *redeemCcoin* function to redeem policy verification, and its pseudo-code is shown in Algorithm 6.

Algorithm 1 (*createCcoin*) Create a new Ccoin.

```
Input: _device, _policy
Output: tokenId
1: Generate new tokenId
2: Require !_exists(tokenId)
3: Set owner BCA address of tokenId = msg.sender
4: Set holder BCA address of tokenId = msg.sender
5: Set device BCA address of tokenId = _device
6: Set policy of tokenId = _policy
7: Set timestamp = block timestamp
8: Set isValid of tokenId = 1
9: Return tokenId
```

Table 2. Function and events of Ccoins.

Functions and Events	Defined by ERC-721
function *transferFrom* (address _from, address _to, uint256 _tokenId) external payable;	Yes
function *ownerOf* (uint256 _tokenId) external view returns(address);	Yes
function *balanceOf* (address _owner) external view returns(uint256);	Yes
function *createCcoin* (address _device, address _owner) external view returns (uint256);	No
function *updatePolicy* (json _policy, uint256 _tokenId) external;	No
function *revokeCcoin* (uint256 _tokenId) external view returns (bool);	No
function *redeemCcoin* (uint256 _tokenId) external;	No
function *policyCheck* (uint256 _tokenId) external view returns (bool);	No
function *createCcoin* (address _device, address _owner) external view returns (uint256);	No
event *Transfer* (address _from, address _to, uint256 _tokenId);	Yes
event *PolicyModified* (uint256 _tokenId);	No
event *RevokeCcoin* (uint256 _tokenId);	No
event *AccessAllowed* (address _from, address _to, uint256 _tokenId);	No
event *AccessDenied* (address _from, address _to, uint256 _tokenId);	No
event *DissatisfyPolicy*(address _from, address _to, uint256 _tokenId);	No

Algorithm 2 (*transferFrom*) Transfer Ccoin to other participants.

```
Input: _oldHolder, _newHolder, _tokenId
1: Require holder BCA address of tokenId == msg.sender
2: Set holder BCA address of tokenId == _newHolder
3: Send event Transfer
```

Algorithm 3 (*updatePolicy*) Modify the access policy of Ccoin.

```
Input: _newPolicy, _tokenId
1: Require owner BCA address of _tokenId == msg.sender
2: Set policy of tokenId = _newPolicy
3: Send event PolicyModified
```

Algorithm 4 (*revokeCcoin*) Revoke a Ccoin.

```
Input: _tokenId
1: Require owner BCA address of _tokenId == msg.sender
2: Require _exists(tokenId)
3: Set isValid of tokenId = false
4: Send event RevokeCcoin
```

Algorithm 5 (*redeemCcoin*) Redeem resources using Ccoin.

```
Input: _tokenId
1: Require holder BCA address of _tokenId == msg.sender
2: transferFrom(msg.sender, device BCA address of _tokenId, tokenId)
3: If policyCheck(_tokenId) == true
4: then Set isValid of tokenId = false
5: And Send event AccessAllowed
6: else Send event AccessDenied
```

Algorithm 6 (*policyCheck*) Check if the current digital or physical environment satisfies the redemption conditions defined in the access policy.

```
Input: _tokenId
1: Require device BCA address of _tokenId == msg.sender
2: if device.getParam $\subseteq$ _policy of _tokenId
3: then Return true
4: else Send event DissatisfyPolicy
5: and Return false
```

Participants can be regular blockchain nodes or users that can communicate with the blockchain. Participants operate Ccoin by sending signed messages to the blockchain, and the authenticity of the message is verified by the blockchain. Before the Ccoin operation, the function caller's credentials also need to be checked. Ccoin stays on the chain until it is redeemed, and a new transaction is issued every time a Ccoin operation function call is made. The transaction includes Ccoin as well as a script that records the call information and execution activities of the function for auditing purposes.

4.2. Workflow

CcBAC can provide fine-grained and responsible access control and has security trust at the encryption level. Its workflow can be mainly divided into three parts: verification, authorization, and audit, as shown in Figure 2.

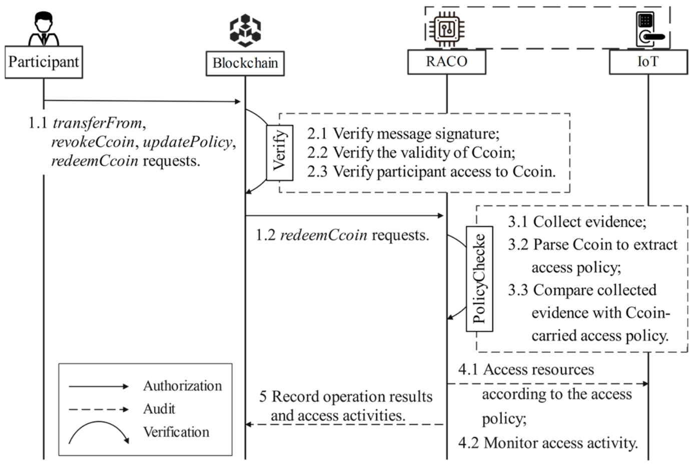

Participant
Blockchain
RACO
IoT
1.1 transferFrom, revokeCcoin, updatePolicy, redeemCcoin requests.
Verify
2.1 Verify message signature;
2.2 Verify the validity of Ccoin;
2.3 Verify participant access to Ccoin.
1.2 redeemCcoin requests.
PolicyChecke
3.1 Collect evidence;
3.2 Parse Ccoin to extract access policy;
3.3 Compare collected evidence with Ccoin-carried access policy.
4.1 Access resources according to the access policy;
4.2 Monitor access activity.
5 Record operation results and access activities.
Authorization
Audit
Verification

Figure 2. Workflow diagram.

The verification process is divided into two parts: on the one hand, it includes the verification of identity and transaction authenticity in the blockchain, and on the other hand, it includes the verification of the access control policy in the authorization process, i.e., the *policyCheck* function. Specifically, *policyCheck* relies on the trusted access control object RACO based on TEE to implement. First, after receiving the redemption request, RACO collects data from the physical environment; then, it establishes a secure communication with the blockchain system and extracts the access policy from the received Ccoin; finally, it compares the collected physical environment data with the access policy carried by the Ccoin to determine whether the redemption conditions of the resource are met.

The authorization process in CcBAC evaluates a person's or process's eligibility for a specific access activity. If approved, it allows for accessing resources, modifying permissions, or redeeming Ccoin according to the access policy. This process encompasses five functions: *createCcoin*, *updatePolicy*, *transferFrom*, *revokeCcoin*, and *redeemCcoin*. To facilitate the management of Ccoin in the distributed ledger, we developed the UnRedeemed Policy Output (URPO) model based on the Bitcoin UTXO model, whose block structure is shown in Figure 3. In the URPO model, each block contains multiple Ccoin, and each Ccoin contains its metadata and the TX script field that records the Ccoin transaction activities. The Ccoin access policy is represented as JSON-formatted key-value pairs to achieve flexible access policies at different granularity levels.

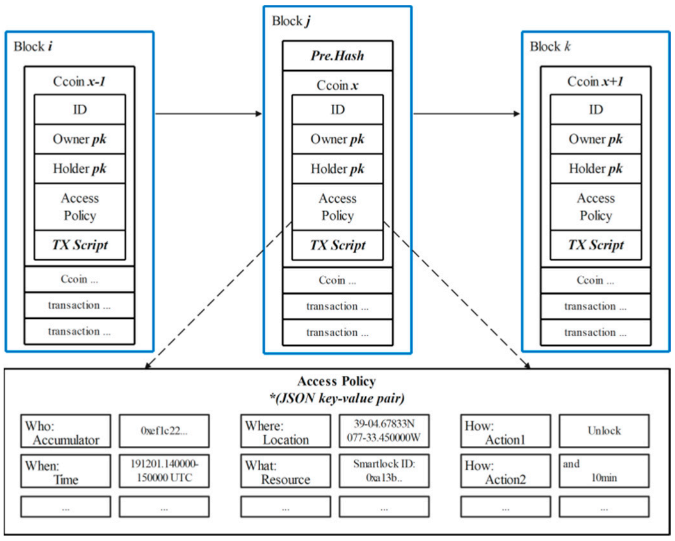


Figure 3. Block structure and URPO.

CcBAC provides native auditability and traceability, as all operations on the chain, including resource access, are stored in the TX script field of the blockchain. All nodes on the blockchain can verify their accuracy.

5. Experiment Simulation and Analysis

5.1. Experiment Analysis

The proposed CcBAC system in this paper comprises two essential components: a blockchain-based distributed ledger and a TEE chip group-based reliable access control object. The blockchain securely manages Ccoin and records all activities in transaction form for auditing purposes; the reliable access control object can embed the blockchain client and sensor driver in its secure area to collect environmental status and make trustworthy access control decisions.

In the first component of the system, the experiment designed two implementations: one is the Golang implementation of Go-Ccoin. The other is the Ethereum-Ccoin based on the Ethereum smart contract. In order to ensure the uniqueness and transferability of Ccoin, this experiment uses the ERC-721 protocol to develop the smart contract interface of CcBAC. Ethereum-Ccoin shows the adaptability of CcBAC to mainstream platforms. These experiments are running on multiple virtual nodes on two PCs, the experimental environment is Intel (R) Core (TM) i7-6700HQ CPU @ 2.6GHz 16GB RAM, the operating system is Ubuntu 18.04.1.

For the second component of the system, we utilize the LPC55S69-EVK microcontroller, which is protected by ARMv8-M TrustZone technology, and sensors (including GPS receiver, temperature sensor, camera) as the main components of RACO, as shown in Figure 4.

ARMv8-M TrustZone is a low-power architecture that uses isolation technology to divide the address space into a secure world and a non-secure world, achieving spatial isolation. Programs in the secure world can not only access resources in the secure world but also those in the non-secure world, while programs in the non-secure world cannot access resources in the secure world. To securely collect the environmental information required for access policies, we implemented the sensor driver program in the TEE secure area to directly connect with the sensor hardware. We also implemented a lightweight blockchain client in the TEE secure area to ensure secure communication with the blockchain through SSL/TLS. The experimental environment and configuration are detailed in Table 3.

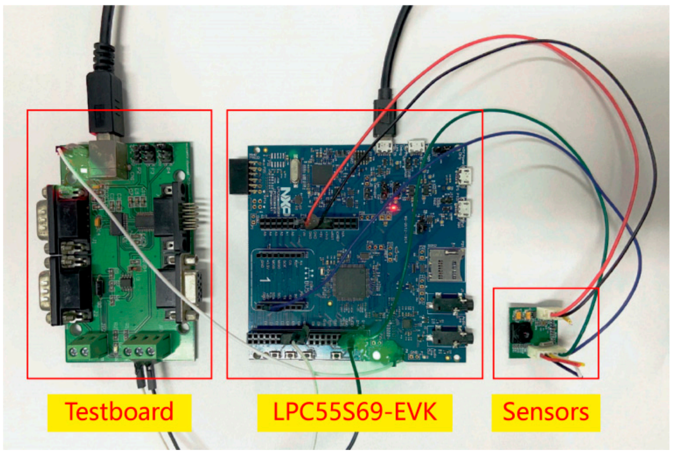


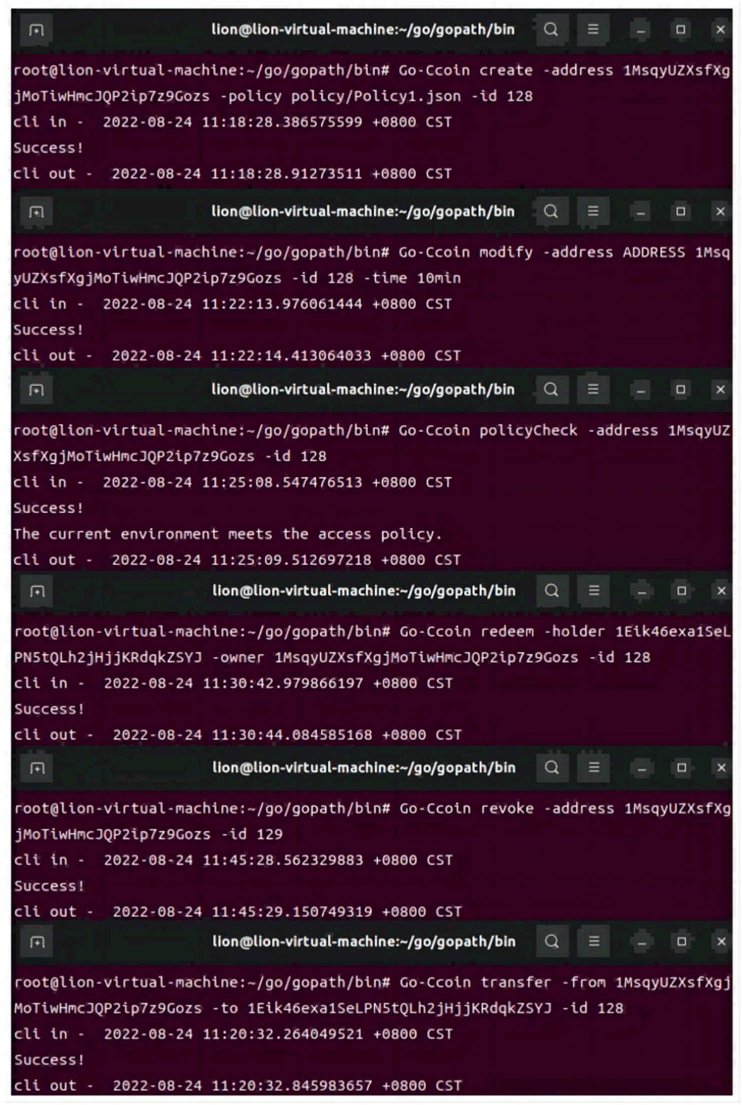
Figure 4. TEE experimental device.

Table 3. Experimental environment and configuration.

Name	**Configuration**
Operating system	Ubuntu 18. 04. 1 GNU/Linux
CPU	8 Intel (R) Core (TM) i7-6700HQ CPU @ 2. 6 GHz
Network card	Intel Corporation Ethernet Connection (2) I219-LM (rev 31)
Memory	16 G Samsung PC4-2400T-UA2
Hard disk	512 G SSD Samsung SSD, 1 T HDD ST1000DM003-1SB1 CC4
TEE chip set	ARMv8-M TrustZone, LPC55S69-EVK

(1) Adaptive Analysis

In order to verify the adaptability of the prototype system proposed in this paper to mainstream platforms, we implemented the prototype on three different platforms: Golang, Ethereum mainnet, and the Ethereum-based consortium chain Quorum. We performed 50 independent tests on each of the functions in the model, calculating the average time each function took to run on each platform. Figure 5 shows the performance testing process of each function in the Golang prototype, and the results are displayed in Figure 6. The time scatter plot in Figure 6 shows that the time distribution of each function is relatively uniform and performance is relatively stable. Among them, the running time of the functions *createCcoin*, *transferFrom*, *updatePolicy*, and *revokeCcoin* is relatively short, approximately distributed between 30 and 80 milliseconds. The function *redeemCcoin* takes longer because it not only includes the time of *policyCheck*, but also requires the access control object CcBAC to sample and analyze sensor readings and take appropriate measures when necessary, and communicate the data back to the blockchain. Figure 7 shows the total time for each Ccoin operation function in the three prototype systems. Due to the maximum change in confirmation time for Go-Ccoin and Ethereum-Ccoin on three different platforms, the logarithmic scale is chosen to represent the data. Figure 7 displays that the normal Go-Ccoin confirmation time for each transaction is around 40–60 milliseconds, while Ethereum-Ccoin in the Quorum takes approximately 1 s and in the Ethereum mainnet takes approximately 30 to 50 s. It can be seen that the proposed CcBAC model has good adaptability.



```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin create -address 1MsqyUZXsfXg
jMoTiwHmcJQP2ip7z9Gozs -policy policy/Policy1.json -id 128
cli in -  2022-08-24 11:18:28.386575599 +0800 CST
Success!
cli out -  2022-08-24 11:18:28.91273511 +0800 CST
```

```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin modify -address ADDRESS 1Msq
yUZXsfXgjMoTiwHmcJQP2ip7z9Gozs -id 128 -time 10min
cli in -  2022-08-24 11:22:13.976061444 +0800 CST
Success!
cli out -  2022-08-24 11:22:14.413064033 +0800 CST
```

```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin policyCheck -address 1MsqyUZ
XsfXgjMoTiwHmcJQP2ip7z9Gozs -id 128
cli in -  2022-08-24 11:25:08.547476513 +0800 CST
Success!
The current environment meets the access policy.
cli out -  2022-08-24 11:25:09.512697218 +0800 CST
```

```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin redeem -holder 1Eik46exa1SeL
PN5tQLh2jHjjKRdqkZSYJ -owner 1MsqyUZXsfXgjMoTiwHmcJQP2ip7z9Gozs -id 128
cli in -  2022-08-24 11:30:42.979866197 +0800 CST
Success!
cli out -  2022-08-24 11:30:44.084585168 +0800 CST
```

```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin revoke -address 1MsqyUZXsfXg
jMoTiwHmcJQP2ip7z9Gozs -id 129
cli in -  2022-08-24 11:45:28.562329883 +0800 CST
Success!
cli out -  2022-08-24 11:45:29.150749319 +0800 CST
```

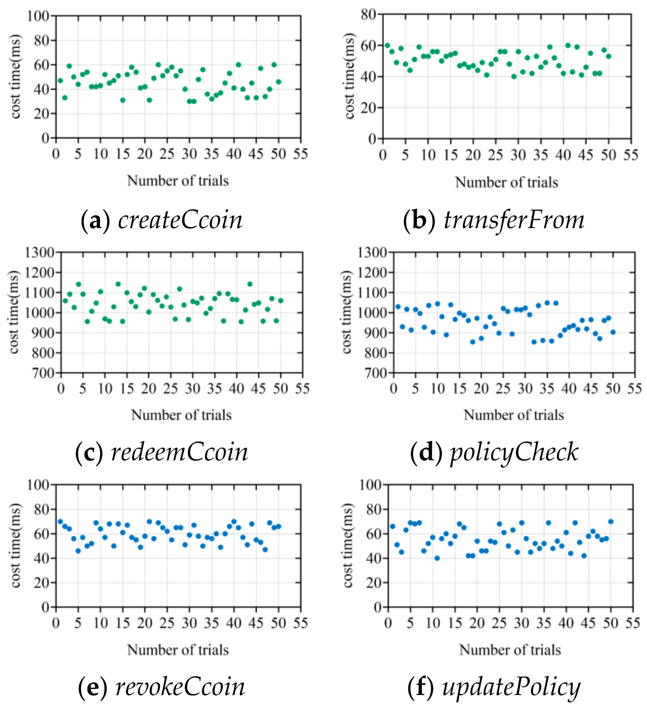
```
lion@lion-virtual-machine:~/go/gopath/bin
root@lion-virtual-machine:~/go/gopath/bin# Go-Ccoin transfer -from 1MsqyUZXsfXgj
MoTiwHmcJQP2ip7z9Gozs -to 1Eik46exa1SeLPN5tQLh2jHjjKRdqkZSYJ -id 128
cli in -  2022-08-24 11:20:32.264049521 +0800 CST
Success!
cli out -  2022-08-24 11:20:32.845983657 +0800 CST
```

Figure 5. Example of function running in Golang prototype.

(2) Performance Analysis

To evaluate the performance of the CcBAC system, we take the Go-Ccoin prototype as an example and design two comparative experiments by simulating the concurrent access of multi-threaded clients to the system. First, we tested the processing time of each function under different concurrent request scenarios with 50, 100, 200, 500, 1000 virtual clients and recorded the time cost for each function to complete its execution. The statistical results can be seen in Figures 8 and 9, with the x-axis representing the number of requests and the y-axis representing the cost in time. Figure 8 shows that the total running time of each function increases as the number of requests increases from 50 to 500, but when the number of requests further increases, the increase in the total running time slows down. Figure 9 shows that when the number of requests increases from 50 to 200, the average time cost of the function slightly increases, but when the number of requests further increases, the average time cost of the function starts to decline and gradually levels off. From these figures, it can be seen that the system's throughput increases with the increase in the number of requests, and when the throughput reaches a certain value, it tends to be stable.

Furthermore, as the number of clients increases, the throughput does not show a noticeable downward trend.



(**a**) *createCcoin* (**b**) *transferFrom*

(**c**) *redeemCcoin* (**d**) *policyCheck*

(**e**) *revokeCcoin* (**f**) *updatePolicy*

Figure 6. Scatterplot of function running time in Golang prototype.

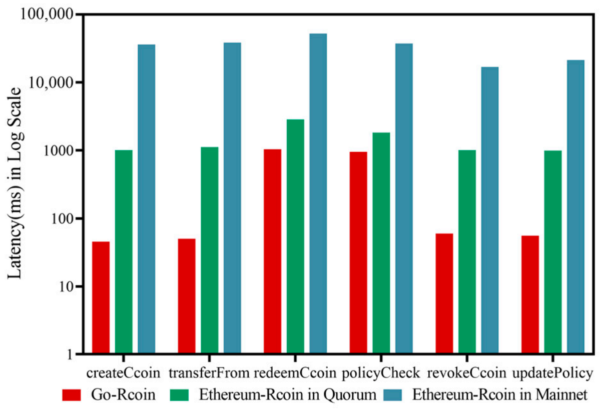


Figure 7. Graph of function running time in three prototypes.

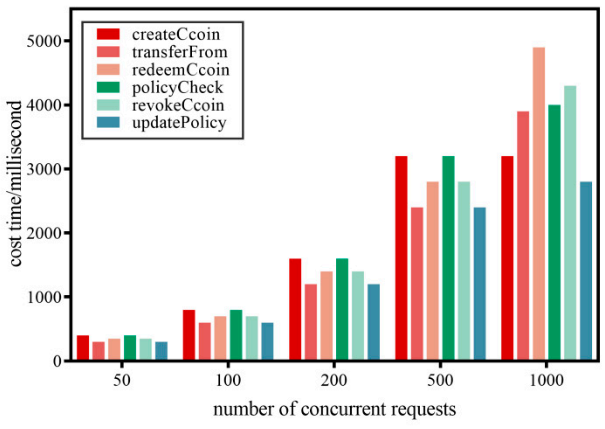


Figure 8. Time cost of functions under different numbers of concurrent requests.

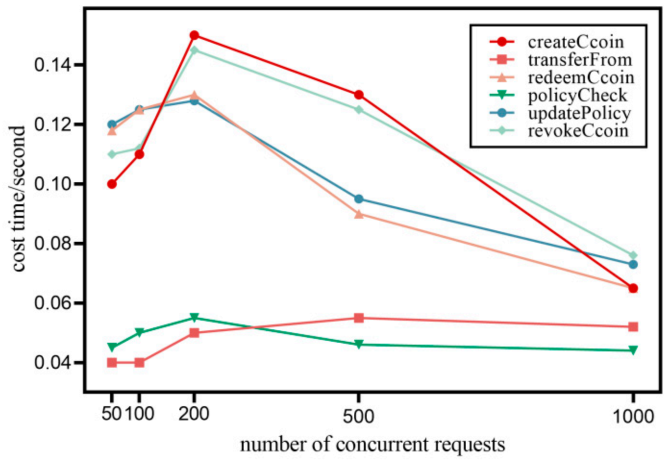


Figure 9. Trend of average time cost of functions under different numbers of concurrent requests.

Second, we implemented CcBAC and PoW consensus algorithms using Golang according to their mechanism principles, and set a reasonable difficulty for the PoW algorithm. We tested the efficiency of data consistency of the system by comparing the time cost of Tendermint-BFT and PoW consensus mechanisms under different numbers of nodes. The number of nodes in the experiment was set to 5 to 100. The calculation results are shown in Figure 10. As can be seen from the figure, under the assumption of a secure PoW difficulty, the cost time of CcBAC consensus is far less than that of PoW.

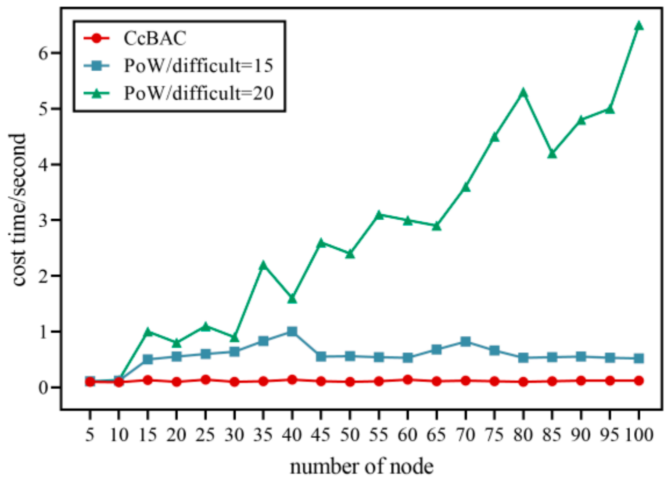


Figure 10. Comparison of CcBAC and PoW consensus speed.

(3) Performance Comparison

Finally, we compared the performance of three different schemes (the schemes of Pal et al. [27], the schemes of Alansari et al. [31], and our CcBAC) in the evaluation process of their policies. We randomly defined 20 policies and executed each of them 50 times, recording the time spent in the decision-making process. Here, we only consider the decision-making process, not the process of RACO collecting environmental parameters. The test results are shown in Figure 11, where the horizontal axis represents the policy number, and the vertical axis represents the running time.

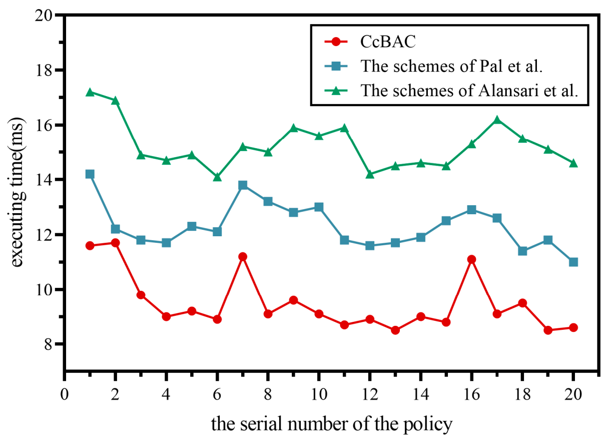


Figure 11. Performance comparison of policy evaluation [27,31].

Figure 11 shows that the execution time of policies using the CcBAC scheme is shorter than the time used by the other two schemes. We speculate that there might be two reasons for this.

(1) We used the JSON syntax format to describe the access control policy, which occupies less memory and has less parsing time compared to the XML used by the other two schemes.

(2) The decision process for the other two schemes requires building a policy decision tree, while our control policy can be directly executed by the script interpreter.

The above experiments and analysis prove that our CcBAC model can not only take into account the advantages of fine-grained, autonomous authorization, security, and auditability, but also maintain high performance in large-scale request environments. In distributed systems, it can effectively achieve consensus and ensure data consistency.

5.2. Security Analysis

In practical situations, an access control system may be subject to two types of attacks: the forging of access privileges and the violation of access policies. Based on the analysis in reference [38], this paper adopts a binomial random process to describe the competitive relationship between honest nodes and attack nodes. Only when the length of the block created by the attack node is greater than that created by the honest node can the attack be successful. Suppose the honest node extends z blocks, the attack node extends k blocks, the probability that the attack node obtains the right to account is q, and honest nodes have a probability p of obtaining the next block, $p + q = 1$, then the likelihood that the attacker closes the block gap is equivalent to the gambler’s ruin problem (Gambler’s Ruin Problem, referred to as GRP). Because k may be any non-negative integer. The probability of k taking values follows a Poisson distribution, and the probability of k occurring is:

$$p_k = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2}$$

where λ is the mean of k, satisfying the following proportionality:

$$\lambda = z \cdot \frac{q}{p} \tag{3}$$

When the tampering chain extends k blocks, the probability of catching up to the honest chain is:

$$q_z = f(x) = \begin{cases} 1, & p \le q \\ \left(\frac{q}{p}\right)^{z-k}, & p > q \end{cases} \tag{4}$$

We calculate the probability for all values of k ranging from 0 to positive infinity and sum them up to determine Attacker's likelihood of successfully tampering the block data:

$$P = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} \left(\frac{q}{p}\right)^{z-k}, & k \leq z \\ 1, & k > z \end{cases} \tag{5}$$

To avoid infinite series summation when calculating the final result, we further transform it:

$$P = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \left(1 - \left(\frac{q}{p}\right)^{(z-k)}\right) \tag{6}$$

We use MATLAB to conduct simulation experiments, and obtain the relationship between the probability P of the attacker successfully tampering with the block and the number n of blocks, as shown in Figure 12. When q is less than 0.5, the probability of successful attack decreases rapidly with the increase of the number of blocks. On the contrary, only when the probability q that the attacker obtains the ownership of the next node is greater than or equal to 0.5, the attacker can possibly succeed in tampering with the next block. In other words, only when the attacker obtains more than 50% of the nodes on the block chain can, it control the data flow of the entire block chain. Because the number of devices in the IoT ecosystem is usually huge, each device can serve as a light node of the block chain. This paper is based on the block chain IoT device access control mechanism, which can effectively resist the access rights of the forged attack, meeting the security requirements of the device access control in the IoT ecosystem.

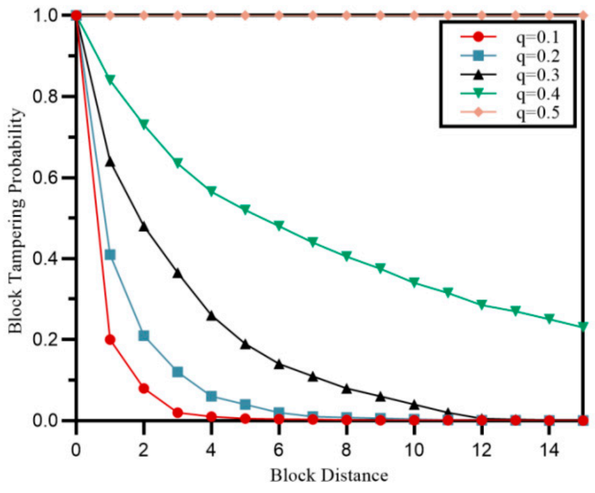


Figure 12. Probability of a successful attack.

For the second type of attack, we consider two types of access policy violations. On the one hand, it stems from the IoT device itself having weak security, unable to guarantee the confidentiality and integrity of the code. Most IoT devices are lightweight, and typical lightweight devices cannot guarantee their own security. Attackers may be able to compromise the security of access control through IoT devices with weak security. Our CcABC model adopts TEE to implement a reliable access control object, providing a secure environment for executing code and storing data. Interacting with the real world through TEE ensures privacy and trust, establishes a secure connection with the blockchain, extends trust from the chain to the off-chain, and ensures the safe retrieval of access policies from Ccoin in the blockchain. Based on the access conditions, it makes access decisions and monitors the access process, ensuring that the resource accessor strictly follows the access policy defined by the resource owner, and safely defending against violations of access policies. On the other hand, the coarse-grained nature of access control models themselves is also an important factor contributing to policy violations and attacks. The PBAC access control model that this paper is based on can ensure sufficient fine-grainedness because its primary focus is on the resources, while other access control models focus on the users. For example, other models ask "What users do I have and what can they do to which resources?" Control is composed of subjects (who is gaining access), permissions (what is being accessed or used), and roles (what permissions can be assigned to the subject). Our access control model asks "what types of resources do I have and who/how is using or managing these resources?" Control is determined by the subject (who is granted access), the action (what behavior is being discussed), the object (what resource is being accessed or used), and the context (defining the acceptable environment or other parameters for access). Defining *who*, *what*, *where*, *when*, and *how* provides the access control policy with finer granularity, thus preventing policy violations.

5.3. Comparative Analysis

The following is a translation of the content: In terms of comparison, the proposed access control model in this paper is evaluated from two aspects: first, it is compared with traditional models (as shown in Table 4), and second, the advantages and disadvantages of the CcBAC model in this paper are analyzed. It can be seen that the model in this paper has certain advantages in the aspects of massiveness, dynamicity, and distributed of the new Internet of Things.

Table 4. Comparison analysis of our model and traditional access control models.

Name	Massiveness	Dynamicity	Distributed
Traditional Access Control Models	As the data and access requests increase, access control policies grow exponentially, leading to high system overhead and low access efficiency.	Static allocation of access rights is unable to meet dynamic access needs; coarse-grained granularity cannot flexibly respond to frequent changes in access requests.	Does not support a unified standard for access control policies, making it difficult for parties to share information.
Our Model	As the amount of data and access requests increases, the access control policy grows linearly, resulting in low system overhead and high access efficiency.	With the increase of data and access requests, the access control policy grows linearly, resulting in low system overhead and high access efficiency.	Fine granularity and good flexibility can support uniform access control policy standards among parties, and the use of blockchain storage strategy for policy storage facilitates information sharing.

Furthermore, in comparison with the existing research models, as shown in Table 5, the CcBAC model has the following advantages:

Fine-grained: CcBAC access control can precisely define *who*, *what*, *where*, *when* and *how*, making the access policy have better granularity.

Security: The model in this paper performs all access control operations on the blockchain, ensuring secure storage and atomic data transitions. The RACO implemented by TEE ensures the accuracy of access policies and makes proper decisions while monitoring the access process.

User-friendly access: If you need to request a resource, you only need to use Ccoin to exchange the corresponding resource. You can also flexibly transfer Ccoin from one holder to another through the audit program, realizing dynamic transfer of access permissions.

Self-authorization: Access to resources can only be decided by the resource owner when needed, through Ccoin definition and issuance, without the need for third-party intervention.

Table 5. Comparison and analysis of the proposed model with existing research solutions.

Model	Based on Blockchain	Fine-Grainedness	Security	Ease of Access	Self-Granting Authorization	Auditability	Access Process Control
DCapBAC [21]	No	Yes	Access control mechanism	Different requests are required for different resources, which is inconvenient for access.	No	No	No
BlendCAC [15]	Yes	Yes	Access control mechanism; Blockchain	As above, not convenient for access.	Yes	Yes	No
The model in reference [31]	Yes	No	Access control mechanism, Asymmetric encryption; TEE	As above, not convenient for access.	No	Yes	No
Our model	Yes	Yes	Access control mechanism; Blockchain; TEE	Flexible access policy formulation, no need to send different requests, easy access.	Yes	Yes	Yes

Auditability: The CcBAC model in this paper has native auditability because all operations on the chain are stored in the TX script field of the blockchain. All nodes on the blockchain can verify its accuracy.

Access process control: RACO is a reliable access control object representing the resource owner, which is responsible for verifying the validity of the Ccoin function, checking whether access policy are met, and monitoring the access process. By combining with the trusted execution environment (TEE), all activities in the access process are ensured to be securely recorded.

6. Conclusions

Applying blockchain technology to access control is a major trend in current blockchain research. In this paper, we adopt blockchain technology and propose a crypto-currency-based access control model CcBAC. On the one hand, it fully plays the characteristics of decentralization and tamper resistance of blockchain; on the other hand, it effectively controls and manages resources through fine-grained access control. The core idea of this paper is: using blockchain as the carrier of access control policy, changing the traditional "centralized" policy decision-making mode to automatic mode by smart contract, and making the policy execution process and result public and verifiable; by using the crypto-currency Ccoin, access permissions are solidified as secure digital assets, allowing the resource owner to independently define access to their resources without the need for third-party intervention; access constraints and access process and its contextual relationship can be precisely defined in Ccoin, and access control can be implemented in fine granularity by constructing access policies; all operations on Ccoin, whether on-chain or off-chain, can be safely recorded, and policy compliance can be verified by blockchain and powerful access control objects, thus achieving powerful auditability.

In recent years, with the rapid development of the Internet of Things, the trend of IoT access control policies has become increasingly complex, presenting new challenges for the efficiency of IoT access control policies. Optimization algorithms such as the bat algorithm [39] and particle swarm algorithm [40] provide us with new ideas for solving this problem. In future work, we will use the bat algorithm or low discrepancy sequence initialization methods to evaluate user/resource access behavior and adjust the access control policy in real-time based on the evaluation results, thereby improving the security of the system and allowing administrators to better control and monitor access control policies.

Author Contributions: Writing—original draft preparation, methodology, software, and validation of the proposed scheme, E.L.; conceptualization, W.Z.; writing—review and editing and funding acquisition, W.J., E.L.; supervision, W.J.; project administration, W.J.; visualization, Y.Y. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 61801489, Natural Science Foundation of Hunan Province of China under Grant No. 2020JJ4249, and Education Department of Hunan Province of China under Grant No. 21A0374.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alwarafy, A.; Al-Thelaya, K.A.; Abdallah, M.; Schneider, J.; Hamdi, M. A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Internet Things J.* **2020**, *8*, 4004–4022. [CrossRef]
2. Yang, Y.Y.; Zhou, W.; Zhao, S.R.; Liu, C.; Zhang, Y.; Wang, H.; Wang, W.; Zhang, Y. Survey of IoT security research: Threats, detection and defense. *J. Commun.* **2021**, *42*, 188–205.
3. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
4. Shuqin, Z.; Guangyao, B.; Hong, L.; Minzhi, Z. IoT Security Knowledge Reasoning Method of Multi-Source Data Fusion. *J. Comput. Res. Dev.* **2022**, *59*, 2735–2749.
5. Liu, Q.; Jin, Z.; Chen, C.; Gao, X.; Zheng, N.; Fang, Y.; Feng, Y. Survey on Internet of Things Access Control Security. *J. Comput. Res. Dev.* **2022**, *59*, 2190–2211.
6. Ravidas, S.; Lekidis, A.; Paci, F.; Zannone, N. Access control in Internet-of-Things: A survey. *J. Netw. Comput. Appl.* **2019**, *144*, 79–101. [CrossRef]
7. Krishna, R.R.; Priyadarshini, A.; Jha, A.V.; Appasani, B.; Srinivasulu, A.; Bizon, N. State-of-the-art review on IoT threats and attacks: Taxonomy, challenges and solutions. *Sustainability* **2021**, *13*, 9463. [CrossRef]
8. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. In *Decentralized Business Review*; Elsevier: Amsterdam, The Netherlands, 2008; p. 21260.
9. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.-Y. Blockchain-enabled smart contracts: Architecture, applications, and future trends. IEEE Trans. *Syst. Man. Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
10. Peng, K.; Li, M.; Huang, H.; Wang, C.; Wan, S.; Choo, K.-K.R. Security challenges and opportunities for smart contracts in Internet of Things: A survey. *IEEE Internet Things J.* **2021**, *8*, 12004–12020. [CrossRef]
11. Lin, S.-Y.; Zhang, L.; Li, J.; Ji, L.; Sun, Y. A survey of application research based on blockchain smart contract. *Wirel. Netw.* **2022**, *28*, 635–690. [CrossRef]
12. Rifi, N.; Rachkidi, E.; Agoulmine, N.; Taher, N.C. Towards using blockchain technology for IoT data access protection. In Proceedings of the 2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB), Salamanca, Spain, 12–15 September 2017; pp. 1–5. [CrossRef]
13. Jemel, M.; Serhrouchni, A. Decentralized access control mechanism with temporal dimension based on blockchain. In Proceedings of the 2017 IEEE 14th International Conference on E-Business Engineering (ICEBE), Shanghai, China, 4–6 November 2017; pp. 177–182. [CrossRef]
14. Mei, Y. Simplification model construction of internet access control based on blockchain. *J. Commun. Univ. China* **2017**, *24*, 7–12.
15. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. 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), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1027–1034. [CrossRef]
16. Qiu, J.; Tian, Z.; Du, C.; Zuo, Q.; Su, S.; Fang, B. A survey on access control in the age of internet of things. *IEEE Internet Things J.* **2020**, *7*, 4682–4696. [CrossRef]
17. Salonikias, S.; Gouglidis, A.; Mavridis, I.; Gritzalis, D. Access control in the industrial internet of things. In *Security and Privacy Trends in the Industrial Internet of Things*; Springer: Cham, Switzerland, 2019; pp. 95–114.
18. Rashid, M.; Parah, S.A.; Wani, A.R.; Gupta, S.K. Securing E-Health IoT data on cloud systems using novel extended role based access control model. In *Internet of Things (IoT) Concepts and Applications*; Springer: Cham, Switzerland, 2020; pp. 473–489. [CrossRef]

19. Zhang, Y.; Nakanishi, R.; Sasabe, M.; Kasahara, S. Combining IOTA and attribute-based encryption for access control in the Internet of Things. *Sensors* **2021**, *21*, 5053. [CrossRef]
20. Yuan, E.; Tong, J. Attributed based access control (ABAC) for web services. In Proceedings of the IEEE International Conference on Web Services (ICWS'05), Orlando, FL, USA, 11–15 July 2005. [CrossRef]
21. Smari, W.W.; Clemente, P.; Lalande, J.-F. An extended attribute based access control model with trust and privacy: Application to a collaborative crisis management system. *Future Gener. Comput. Syst.* **2014**, *31*, 147–168. [CrossRef]
22. Bhatt, S.; Pham, T.K.; Gupta, M.; Benson, J.; Park, J.; Sandhu, R. Attribute-based access control for AWS internet of things and secure industries of the future. *IEEE Access* **2021**, *9*, 107200–107223. [CrossRef]
23. Shen, H.; Liu, S. A context-aware capability-based access control framework for the Internet of things. *J. Wuhan Univ. (Nat. Sci. Ed.)* **2014**, *60*, 424–428.
24. Nakamura, S.; Enokido, T.; Takizawa, M. Information flow control based on the CapBAC (capability-based access control) model in the IoT. *Int. J. Mob. Comput. Multimed. Commun. (IJMCMC)* **2019**, *10*, 13–25. [CrossRef]
25. Hussein, D.; Bertin, E.; Frey, V. A community-driven access control approach in distributed IoT environments. *IEEE Commun. Mag.* **2017**, *55*, 146–153. [CrossRef]
26. Zhi, L.; Jing, W.; Xiao-su, C.; Lian-xing, J. Research on policy-based access control model. In Proceedings of the2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, 25–26 April 2009; Volume 2, pp. 164–167. [CrossRef]
27. Pal, S.; Hitchens, M.; Varadharajan, V.; Rabehaja, T. Policy-based access control for constrained healthcare resources in the context of the Internet of Things. *J. Netw. Comput. Appl.* **2019**, *139*, 57–74. [CrossRef]
28. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
29. Di Francesco Maesa, D.; Mori, P.; Ricci, L. Blockchain based access control. In *Distributed Applications and Interoperable Systems: 17th IFIP WG 6.1 International Conference, DAIS 2017, Proceedings of the 12th International Federated Conference on Distributed Computing Techniques, DisCoTec 2017, Neuchâtel, Switzerland, 19–22 June 2017*; Springer: Cham, Switzerland, 2017; pp. 206–220.
30. Rouhani, S.; Belchior, R.; Cruz, R.S.; Deters, R. Distributed attribute-based access control system using permissioned blockchain. *World Wide Web* **2021**, *24*, 1617–1644. [CrossRef]
31. Alansari, S.; Paci, F.; Sassone, V. A distributed access control system for cloud federations. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2131–2136. [CrossRef]
32. Yan, Y.; Wei, C.; Guo, X.; Lu, X.; Zheng, X.; Liu, Q.; Zhou, C.; Song, X.; Zhao, B.; Zhang, H.; et al. Confidentiality support over financial grade consortium blockchain. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 2227–2240. [CrossRef]
33. Enkhtaivan, B.; Takenouchi, T.; Sako, K. A fair anonymous auction scheme utilizing trusted hardware and blockchain. In Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST), Fredericton, NB, Canada, 26–28 August 2019; pp. 1–5. [CrossRef]
34. Ayoade, G.; Karande, V.; Khan, L.; Hamlen, K. Decentralized IoT data management using blockchain and trusted execution environment. In Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 6–9 July 2018; pp. 15–22. [CrossRef]
35. Liang, Y.; Li, Y.; Shin, B.-S. FairCs—Blockchain-based fair crowdsensing scheme using trusted execution environment. *Sensors* **2020**, *20*, 3172. [CrossRef] [PubMed]
36. Cheng, J.; Li, J.; Xiong, N.; Chen, M.; Guo, H.; Yao, X. Lightweight mobile clients privacy protection using trusted execution environments for blockchain. *CMC-Comput. Mater. Contin.* **2020**, *65*, 2247–2262. [CrossRef]
37. Enkhtaivan, B.; Inoue, A. Mediating data trustworthiness by using trusted hardware between iot devices and blockchain. In Proceedings of the 2020 IEEE International Conference on Smart Internet of Things (SmartIoT), Beijing, China, 14–16 August 2020; pp. 314–318. [CrossRef]
38. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
39. Bangyal, W.H.; Hameed, A.; Ahmad, J.; Nisar, K.; Haque, M.R.; Ibrahim, A.A.A.; Rodrigues, J.J.P.C.; Khan, M.A.; Rawat, D.B.; Etengu, R. New modified controlled bat algorithm for numerical optimization problem. *Comput. Mater. Contin.* **2022**, *70*, 2241–2259. [CrossRef]
40. Bangyal, W.H.; Nisar, K.; Ibrahim, A.A.B.A.; Haque, M.R.; Rodrigues, J.J.P.C.; Rawat, D.B. Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems. *Appl. Sci.* **2021**, *11*, 7591. [CrossRef]