systems

MDPI

Article

FGAC: A Fine-Grained Access Control Framework for Supply Chain Data Sharing

Yang Liu [1], Xiangyu Li [1] and Yan Ma [2,3,*]

[1] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 200120, China
[2] School of Accounting, Nanjing University of Finance and Economics, Nanjing 210023, China
[3] School of Computing, National University of Singapore, Singapore 117417, Singapore
* Correspondence: yanma@nufe.edu.cn

Abstract: With the rapid development of digital economics, a large number of data have been accumulated in the supply chain system, and data islands have appeared. Data sharing is an imperative way to unlock the data value of a supply chain system. A safe and effective access control mechanism for privacy-sensitive data is key in data sharing. At present, traditional access control mechanisms are static, single-factor control, and prone to a single point of failure. For dealing with these, a fine-grained access control (FGAC) framework for supply chain data sharing is proposed, based on the blockchain Hyperledger Fabric. It augments role-based access control (RBAC) by giving different attribute keywords to different types of users. This framework is implemented in smart contract Chaincodes and quantitatively verified by using the model-checking tool UPPAAL. The experiment results show that the FGAC framework enhances the efficiency and safety in the process of data sharing for the supply chain system, compared with the existing works.

Keywords: blockchain; supply chain system; data sharing; access control; system verification

Citation: Liu, Y.; Li, X.; Ma, Y. FGAC: A Fine-Grained Access Control Framework for Supply Chain Data Sharing. *Systems* **2022**, *10*, 208. https://doi.org/10.3390/systems10060208

Academic Editors: Anders Hansen Henten and Iwona Windekilde

Received: 30 September 2022
Accepted: 2 November 2022
Published: 4 November 2022

1. Introduction

With the rapid development of the digital supply chain, data are becoming important production factors and are beginning to affect every aspect of our daily life. A large volume of data, i.e., big data, also raises a series of challenges, some of which include data management inefficiencies, unauthorized access, malicious attacks, single points of failure through centralization, and many others [1]. Moreover, due to the uneven distribution of data resources, the data siphon effect of large enterprises is serious; meanwhile, some phenomena of "data islands" are emerging. Data sharing is an imperative way to unlock the data value of a supply chain system. At present, the centralized storage system, namely the cloud, is the main method for data sharing services. The cloud has the ability to process large volumes of data quickly, and it enables the accuracy, efficiency, and speed of data processing. According to the 2021 Cisco Global Cloud Index report, 94% of workloads will be processed on cloud servers by 2021, compared with 83% in 2016. In fact, using cloud services to share data may lead to the data owner not owning the data or even controlling the risk of data privacy leakage [2]. Especially for some health or financial supply chain systems, their sensitive data have very important value, so any violation of access is not allowed.

Compared with the traditional production factors, data resources own digital characteristics, such as easy-to-copy and difficult-to-maintain confidentiality. Traditional data sharing methods have some drawbacks. First, its dependence on third parties is costly and requires that all users have a high degree of trust in third parties. Secondly, if risks arise in the data sharing process, users will need to validate manually. This also requires exposing confidential data to third parties and incurs complex administrative overhead, such as the need for additional legal contracts, which is less efficient. Therefore, data sharing through traditional sharing methods may lead to the leakage of personal information or interest loss

in multiple links in the supply chain system. As a decentralized architecture, blockchain is a promising technology for data sharing in supply chain systems. It may replace cloud services for data sharing and help to effectively protect data privacy and security.

Access control mechanism refers to the restriction of the permission or ability of the user to access the data. It is a key for data sharing based on blockchain. However, the existing access control mechanisms in blockchain, such as RBAC [3] or ABAC [4], generally verify access rights through centralized entities that are prone to a single point of failure. Moreover, they are static and single-factor control and have coarse granularity. Once the user's role or attribute is set, the user will always have access rights, even though they may not have the assumed role. Therefore, access control mechanisms have been an important research topic for data sharing, either in cloud-service-based data sharing or blockchain-based data sharing. However, traditional access control strategies cannot meet the requirements of security and fine granularity.

In this paper, we seek to solve the problem of fine-grained access control of data sharing in supply chain systems. Based on the blockchain Hyperledger Fabric, we propose a fine-grained access control framework (FGAC) by extending an access control mechanism, RBAC (role-based access control), and using smart contract Chaincodes in the blockchain to call and trigger FGAC. This will ensure the integrity, fairness, authenticity, and security of data sharing in supply chain systems. FGAC provides different attribute keywords for different types of users on the basis of RBAC and is related to data owners through smart contract Chaincodes when sharing data between users. The specific contributions of this paper are as follows:

1. We extend the RBAC (role-based access control) model with attribute keywords and propose a fine-grained access control (FGAC) framework.
2. We implement the FGAC framework with smart contract Chaincodes in blockchain Hyperledger Fabric and apply it to the data sharing of the supply chain system in Shanghai Port.
3. Using a model-checking tool as the system verification technique, we demonstrate and analyze the feasibility and safety of FGAC framework.

The rest of this article is organized as follows: Section 2 provides an overview of related research about data sharing based on blockchain and access control mechanisms. In Section 3, we propose a fine-grained access control framework (FGAC) and implement it in smart contract Chaincodes in blockchain Hyperledger Fabric. Section 4 presents the actual application scenario, i.e., the data sharing of a supply chain system in Shanghai Port. Section 5 models the FGAC framework by using the model-checking tool UPPAAL and presents system verification results and analysis. Finally, the conclusion and future work are given in Section 6.

2. Related Work

In this section, we highlight some studies that combine blockchain technology with data sharing and access control. In addition, the advantages and disadvantages of some access control strategies based on blockchains are discussed.

2.1. Blockchain and Blockchain-Empowered Data Sharing

Blockchain technology originated from the foundational paper "Bitcoin: A Peer-to-Peer Electronic Cash System" published in 2008 by "Satoshi Nakamoto" [5]. The blockchain does not involve any third-party authority or centralized server [6], and it is implemented in a decentralized network of computing nodes, in which each node keeps the same copy of transaction records [7]. This also enhances the system's ability to handle single points of failure and defend against attacks. In the blockchain, transactions are approved and recorded in the blocks created by miners, which are appended to the blockchain in chronological order. Due to the consensus mechanism implemented by miners' mining tasks on the network, users can trust the globally stored public ledger system instead of

having to establish and maintain a trust relationship with a third party, which effectively solves the drawbacks of traditional data sharing.

As the key underlying technology behind modern cryptocurrency systems such as Bitcoin [5] and Ethereum [8], the blockchain was originally created as a distributed, immutable transaction ledger for cryptocurrency systems. Due to the invention and combination of smart contracts, the blockchain has now developed into an efficient platform for developing distributed and trusted applications and has attracted the attention of a large number of researchers [9]. The smart contract can effectively solve the problems in traditional data sharing and access control and become a link between blockchain technology and access control mechanisms. A smart contract is a coded contract written in a computer language and automatically verified and executed by a computer. Its essence is a collection of predefined instructions and data that have been recorded at a specific address on the blockchain. It automatically executes the contract through a coding program. As long as the contract terms are met, the transaction will be performed automatically without third-party supervision [10]. Like ordinary on-chain transactions, the node will first perform signature verification to ensure the validity of the contract, and the verified contract will be successfully executed after the consensus mechanism. All transactions generated in smart contracts and blockchain networks are saved in a Merkle tree structure in each block. Merkle trees are constructed bottom-up tree data structures, in which all transaction data are hashed and stored as leaf nodes, and the continuous child nodes from leaf to root are hashed until the root hash value is generated and stored in the block header [11].

Some works that combine blockchain systems with data sharing have shown initial results, and most of these works are currently being used in medical electronic records (EHR) and the Internet of Things. Azaria et al. proposed MedRec [3], a decentralized record electronic medical record management system using blockchain technology, which provides patients with a comprehensive, immutable log with easy access to their medical information, covering provider and treatment websites. Ref. [12] systematically discussed how to store, retrieve, and share files using a blockchain structure in a decentralized environment. They used the blockchain to realize the scheme of data integrity, and the main content of the discussion includes the definition of transaction information, block information, and other specific implementation measures. Ref. [13] proposed a medical data management framework named CrowdMed, which designed an access control scheme for medical data, allowing patients to fully control access to their medical data and how their data are accessed and used, and permissions can be revoked or modified according to the patient's wishes. Additionally, it also encourages patients to share more data for research purposes by designing reward tokens and innovative pricing mechanisms. Ref. [14] proposed a blockchain-based privacy and security-protected EHR sharing protocol for improving diagnosis and effective treatment in the TMIS (Telecare Medicine Information System). The study [15] proposed the concept of the data sharing agreement (DSA) as a basic path and template for the data management of AI applications between various actors. Ref. [16] proposed a blockchain-based medical data sharing model, which has the characteristics of decentralization, security, trustworthiness, collective maintenance, and non-tampering, and is suitable for solving the data sharing of various medical institutions. Ref. [17] designed a consortium medical blockchain system based on the Practical Byzantine Fault Tolerance (PBFT), which is maintained and shared by multiple nodes, and can prevent the medical data from being tampered and leaked. In addition, ref. [18] proposed a blockchain-based trusted data sharing scheme that uses the Paillier encryption system to achieve the confidentiality of shared data and realizes the transaction of shared data through the (p, t) threshold Paillier encryption system to protect transaction information. Ref. [19] proposed BMAC, which is a multi-authority access control scheme based on blockchain technology. It introduces the Shamir secret sharing scheme and blockchain authority and realizes the joint management of each attribute by multiple authorities. Additionally, it builds trust among multiple authorities by utilizing smart contracts to calculate tokens for the properties managed across multiple administrative domains. Ref. [20] designed a secure

data sharing framework based on identity authentication and the blockchain Hyperledger Fabric and proposed a community detection algorithm that can divide clients into different data sharing communities based on the similarity of labeled data, select the scope of data sharing according to the community's detection results of sharing degree evaluation, and improve the efficiency of data sharing. Ref. [21] designed a compressed private data sharing framework that can provide efficient private data management for the product data stored on the blockchain. The scheme uses off-chain procedures to compress and encrypt product data before submitting them to the blockchain and designs two types of transactions to support off-chain/on-chain data access. Ref. [22] used smart contracts and inadvertent transfer protocols, combined with the proposed ether check system, to achieve transaction fairness, autonomy, and transaction time control. Ref. [23] introduced a new multi-keyword, searchable encryption technique that improved the accuracy of the retrieved results and proposed a secure, searchable encryption system based on attribute encryption (ABE), searchable encryption, and blockchain used in the data sharing framework for the letter. However, the above works only consider the security risks in data sharing frameworks and do not consider the security of access control mechanisms in data sharing.

2.2. Access Control Mechanisms

At present, traditional access control models mainly include the discretionary access control (DAC) model, the role-based access control (RBAC) model [3], the attribute-based access control (ABAC) model [4], and the capability-based access control (CapBAC) model [24]. In the studies of [25,26], RBAC models are used as access control mechanisms for blockchain data sharing. In an RBAC model, roles are associated with access rights (e.g., invoke, edit, and execute) and assigned to subjects, and a many-to-many relationship is established between access rights and subjects [27]. However, RBAC, which is widely used, has inherent problems that it cannot overcome. For example, RBAC can no longer restrict access to a role after it is set, unless the role is manually revoked, and RBAC cannot solve the problem of individual user authentication in an organization (people in a department have almost the same role attributes). In this way, once the data are obtained through role attributes, even if the user changes departments or even work units, he can still obtain the desired information (data) and even use it for editing and tampering, which obviously has the great hidden danger of security for the protection of private data. In addition, Wang et al. proposed a data access control and sharing model using a blockchain system [28] that uses attribute-based encryption to control and share enterprise data to achieve fine-grained access control and secure sharing. Ref. [29] proposed a medical data security sharing scheme with a time dimension based on an alliance chain. This scheme uses cloud storage to store medical data ciphertext, uses the alliance chain to store metadata, and encrypts smart contracts and ciphertext strategy attributes. Combined with ciphertext-strategy attribute-based encryption (CP-ABE) technology, a data security sharing protocol is designed to realize fine-grained access control with the time dimension. However, they only improved the access control policy in terms of encryption and did not solve the fundamental problem. Among the classic access control models, ABAC is the most promising model for fine-grained access control. This is because ABAC introduces the contextual information and attributes of subjects and objects in its access control policies. By adding more topic attributes, object attributes, and contextual information to the strategy, we can greatly improve the dynamics and granularity of ABAC. To implement an access control strategy using an ABAC-based scheme, ref. [30] combined various types of attributes in data sharing, such as user attributes, object (i.e., the entity that holds the resource) attributes, environmental attributes, etc. The strategy itself defined a set of rules that indicate the conditions under which the data owner can be granted access rights, but this work did not limit or describe the user role. Moreover, when setting access control policies, we also need to consider the required decision-making continuity and attribute variability; that is, the user still needs to be restricted in access after setting the role, or once the attributes of the

role are changed, their permissions need to be changed; otherwise, it will still cause data leakage and damage the privacy of users.

2.3. Summary

In short, some works have tried to combine blockchain technology with access control for data sharing. However, in the existing works, the access rights of its validating principals are usually handled by a centralized entity, which can throw the entire system into a single point of failure if something goes wrong. In RBAC, if the role is set, it can no longer be restricted unless it is manually revoked, and there are many restrictions on the setting of the role. ABAC does not solve the problem of individual user authentication in an organization, which can lead to a department with almost the same attributes among its users. This makes it possible to obtain the desired information (data) even if the department or work unit is changed for the data obtained through the attributes of the user, which poses a huge security risk. Therefore, this paper proposes a blockchain-based, fine-grained access control mechanism for supply chain data sharing.

3. Fine-Grained Access Control Framework

The FGAC framework sets keywords for different roles, such as the environment, department, project name, etc., and when identifying access permissions, it not only lists the role attributes within the scope of permissions but also matches their attribute keywords to view and transfer data. In addition, it also considers decision-making continuity and the attributes' variability; that is, users still need to be restricted in access after setting their roles, and if role attributes change, their permissions need to also be changed. The FGAC framework is the extension of the RBAC model in essence and is implemented through smart contract Chaincodes. This section describes the model architecture and workflow of the proposed FGAC in detail.

3.1. Access Control Model and Workflow

The architecture and workflow of the FGAC model are shown in Figure 1. It involves four types of smart contract Chaincodes, IPFS [31], encryption algorithms, etc. The specific definition of the events and functions of the four smart contract Chaincodes are described in Sections 3.2 and 3.3.

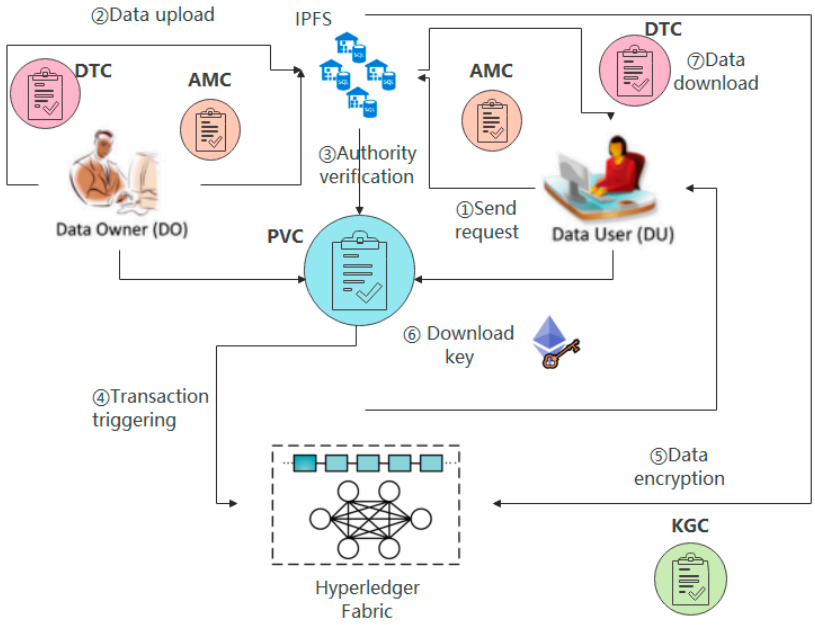

Figure 1. Fine-grained access control framework.

The overall process of the FGAC framework consists of six steps, each of which is explained in the following paragraphs:

1. Send Request: The DU sends a transaction to the data transmission contract to call the access request function in the contract. The transaction contains the target data and the name of the DU device. By calling the access request function in the contract, the DU role attribute is obtained from the database and triggered. The request access event in this function sends the transaction to the DO.
2. Data Upload: The DO sends a transaction request to the data transmission contract through the client to call the data upload function in the contract. The transaction includes the DO device name, the shared data information, and the role attribute keywords.
3. Authority Verification: When both the DO and DU are given their role information, and the attribute keywords are set, the transaction is sent to the attribute validation contract, and the validation function in the contract is called to validate the role attributes and data keywords of the DO and DU. Then, the function queries the attribute parameters in the contract according to the passed parameters, obtains the DO and DU parameters such as domain name, attribute, and blockchain account address, and triggers the matching request event in the function to verify whether both parties meet the permissions.
4. Trigger Transaction: When the data sharing parties satisfy the access control strategy, and the verification is successful, the transaction is packaged and executed. The request processing function in the property matching contract will call the DO and DU attribute information and send the result to the data transmission contract, triggering data sharing.
5. Data Encryption: The key generation contract is called to encrypt the identity information of the DO and DU and the requested shared data. When the verification result is successful, the contract distributes the key to the DU for decryption.
6. Data and Key Download: The DU obtains the key through the key generation contract after satisfying the access control strategy, triggers the transaction to view the shared data stored in IPFS, and matches the plaintext and ciphertext to decrypt the required shared data, to obtain plaintext data.

Users request to share supply chain data through smart contract Chaincodes that execute different functions. When a user requests the supply chain data in the hands of the DO, the DU starts to set user role information and attribute keywords and initiates data sharing with the DO. After receiving the request, the DO verifies the role with user rights and the attribute keywords assigned to the role. If the role information can be matched, the user can read the data but cannot download or use the data. When the attribute keywords are matched, the user who initiates the sharing request can obtain permission to use the data for data sharing.

3.2. Event Definition

This sub-section introduces the access control model and workflow and explains the framework of each step in detail. In FGAC, each user in the supply chain needs to set their own role information and attribute keywords and then can download and use sensitive data by verifying the attribute keywords. This effectively strengthens the security of data in the supply chain.

The four types of smart contracts in Chaincodes are the attribute management contract (AMC), the property verification contract (PVC), the key generation contract (KGC), and the data transmission contract (DTC). Users send request transactions to contracts, call contract-related functions, and complete specific operations to achieve fine-grained access control. In order to explain the detailed access control process in the data sharing between the data owner (DO) and the data user (DU), the definitions of some basic operations are given.

3.2.1. Transaction Sending and Processing

Transaction sending mainly refers to the signed packet of a message sent by an external account to another account on the blockchain. Transaction processing is a process that starts from the account initiating a transaction request and ends when the block containing the transaction is synchronized by the consensus node. When a transaction is sent through the contract, the contract will return the hash address of the transaction, which can be used to query the sender's address, receiver's address, and other related personal information during the data sharing process.

3.2.2. Event Notification and Execution

Events are the communication bridge between contracts and users. Events can be used to notify users (sender and receiver) so that they can easily query and access events through the client. In the actual supply chain data sharing process, users need to send transactions to call smart contract Chaincodes to execute the corresponding request. When the transaction is sent but not packaged and executed, the user will not be able to obtain the return value of the smart contract Chaincodes immediately. When certain operations are completed inside the contract function, the transaction is packaged and executed by triggering an event notification. Additionally, only after the contract writes the event to the blockchain can the front end respond accordingly.

3.2.3. Call of Functions

There are two types of function calls in smart contract Chaincodes, namely internal function calls and external function calls. An internal function call refers to a function calling another function in the same contract Chaincodes; an external function call refers to a function calling a function of another contract. In smart contract Chaincodes, users can perform their desired actions by calling functions.

3.2.4. Attribute Information

The administrator and related users publish their attributes and attribute relationship information to the blockchain, and the attribute management contract collects and integrates the corresponding attribute information and relationship. In addition to the role attribute of the user, for the user in the data sharing process, the attribute keywords can include address, trust degree, status, working time, department and level, etc. It is also possible to add environmental attributes such as the environmental conditions when the data sharing process occurs, the current time of the system, the security level of the system, the IP address it belongs to, etc.

3.2.5. Strategy Match

The administrator (user) publishes the access control strategy to other users in the blockchain, and the smart contract Chaincodes combine the attribute information to describe, collect, and integrate the access control strategy in the blockchain transaction to evaluate the access request. The property verification contract verifies whether the property information of the data requester meets the required requirements, thereby implementing fine-grained access control.

The relevant events are defined in this sub-section, in which the detailed access control process for data sharing between the data owner (DO) and the data user (DU) is described.

3.3. Access Control of Data Sharing Process Using Smart Contract Chaincodes

In this part, we introduce the smart contract part of the FGAC framework and explain the specific functions. According to the different purposes of the smart contracts involved in FGAC, we divide them into four types: the attribute management contract (AMC), the data transmission contract (DTC), the property verification contract (PVC), and the key generation contract (KGC).

3.3.1. Attribute Management Contract (AMC)

This contract is the most important contract in the process of realizing access control. In the process of supply chain data sharing, the access control strategy set by the data owner is collected and integrated by the contract, and the attribute information and its relationship with the data user are collected and integrated. The user's role attributes and the setting of their attribute keywords are received and distributed by the administrator or the user through the attribute management contract. In FGAC, when the data requester sends a transaction request to the system, the contract is executed. After the user registers and authorizes its role and its attribute keywords, the data user's request is forwarded to the data owner, and the properties are executed. The verification contract verifies whether the properties meet the requirements.

3.3.2. Data Transmission Contract (DTC)

This contract is used to upload and download the related data involved in packaging and sharing, as well as the role information and attribute keywords of different users. When the data requester makes an access request and obtains access permission, the data owner packages the data and uploads it to the blockchain through the contract, after which the data requester can obtain the required data and decrypt it.

3.3.3. Property Verification Contract (PVC)

The user's permission verification is performed by this contract. The nature verification contract identifies and evaluates the roles of the data owner and data user and compares whether their attribute keywords are consistent or similar. If the roles are the same, but the attributes are different, the user can only view the data, and data sharing is not available. If the verification is passed, the transaction process is triggered to allow data sharing, and the data user obtains the data through the data transmission contract.

3.3.4. Key Generation Contract (KGC)

When the nature of the data user is verified, the contract uses an encryption algorithm to encrypt the relevant data to be shared and upload the ciphertext to the blockchain. Once the verification of the nature verification contract is completed, the transaction can be triggered, and the data user can obtain the ciphertext and key required for the download through the data transmission contract, and after decryption, the required shared data can be obtained.

In FGAC, each user in the supply chain needs to set their own role information and their own related attribute keywords. At this time, the system obtains the sharing request sent by other users. Through the verification of the role attribute, the users in the supply chain can obtain viewing permission for the data, and then the key sensitive data can be downloaded and used by matching the verification attribute keywords. This effectively strengthens the security of the data in the supply chain.

4. Application Scenario

In this section, we use the supply chain in Shanghai Port as an actual scenario to illustrate the role and necessity of FGAC. The supply chain in Shanghai Port accumulates a large volume of data, such as maritime ships, maritime cargo, port data, and shipping routes. Through the sharing of supply chain big data, various departments in the supply chain can not only track ships and transport goods at sea but also provide supply chain participants with real-time, accurate, and visible individual dynamic information. The scattered dynamic data are precipitated, collected, sorted, and modeled to form a multi-dimensional basic big data model, which provides supply chain dynamics and industry intelligence for the relevant departments of the supply chain. Some cargo owners in the supply chain, especially direct cargo owners or large cargo owners, have higher requirements for information acquisition of transportation and hope that they can always grasp the information of cargo transportation. In the process of data sharing, this information not

only needs to protect the privacy of users but also should not be easily viewed or tampered with by anyone. The FGAC framework can satisfy these requirements well. It is suitable for supply chain data sharing in Shanghai Port. The application diagram of the FGAC framework in the supply chain in Shanghai Port is shown in Figure 2.

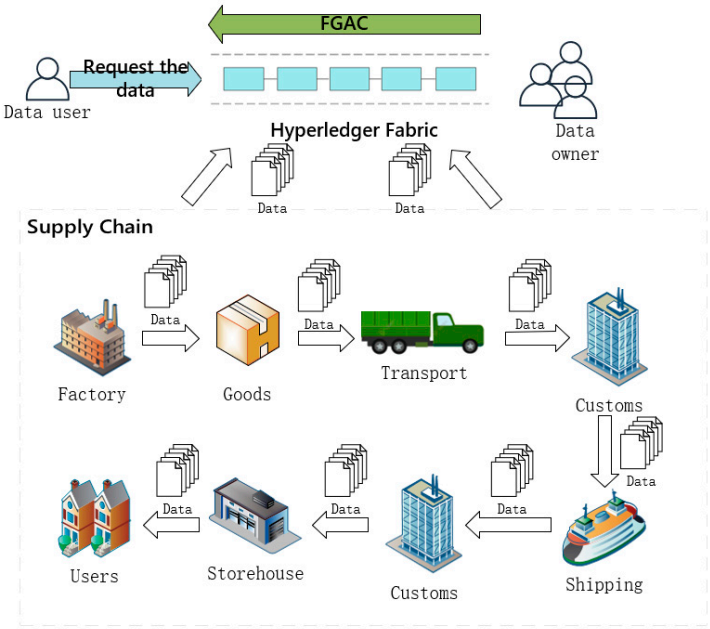

Figure 2. FGAC in supply chain of Shanghai Port.

We divided the data of the supply chain in Shanghai Port into four types: sensitive data, management data, transportation data, and personal data. In the proposed FGAC, the following points are taken into account when setting the access control rights:

- The data owner (DO) has access rights to all raw data;
- Authoritative organizations can obtain some data related to specific projects with high accuracy and timeliness (i.e., they cannot be changed without authorization);
- Data statistic agencies can access supply chain and management data but cannot change them;
- Other relevant departments and data technology companies can only access and obtain shipping-related data, but data accuracy and timeliness are not guaranteed (e.g., there may be a competitive relationship);
- Screening opponents and other companies in the same industry cannot obtain any type of data;
- Other company departments on the chain can share data according to priority, and sensitive data can also be shared depending on the situation and level;
- Neither private nor innovative data can be shared.

The data acquisition process after the FGAC authorization is shown in Figure 3. When the property matching step is passed, the DO downloads the data ciphertext and key together, decrypts and compares them, and downloads the required data from the distributed storage system.

The whole process of cargo transportation in the supply chain is illustrated in Figure 3. The data generated in this process are saved on the blockchain Hyperledger Fabric. When users send sharing requests for the data, the data owner can set the relevant attributes and trigger the FGAC mechanism to control the access of the requester, which makes the data in the entire supply chain effectively protected.

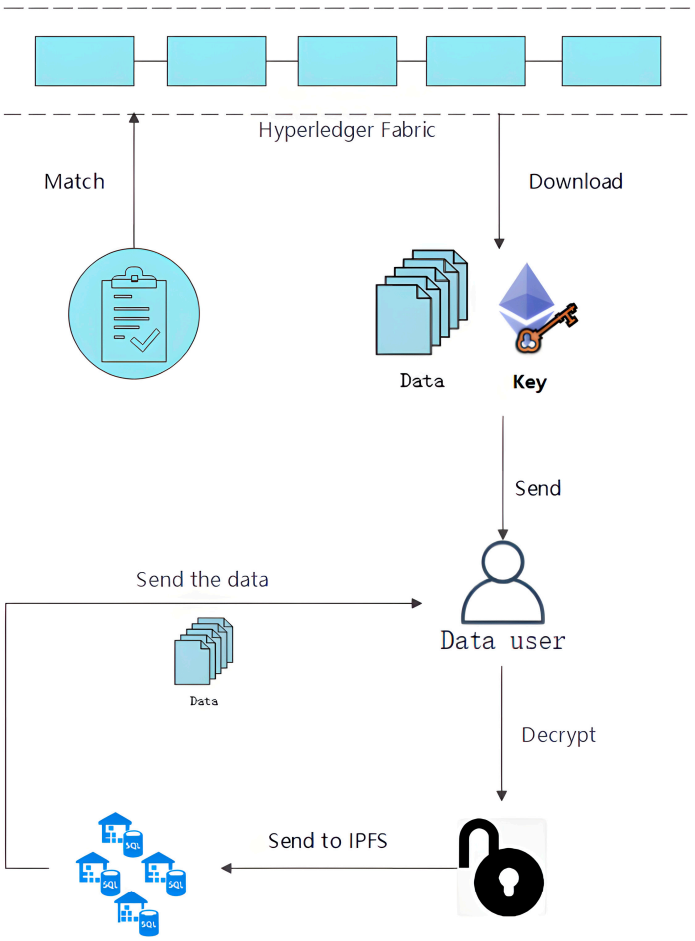

Figure 3. Validated supply chain data process.

5. Experimental Analysis and Verification

5.1. Comparative Analysis

At present, the combination of blockchain technology and access control has become one of the main applications of blockchain systems in data sharing. Table 1 outlines the research on the combination of blockchain and different access control models, which fully reflects the performance advantages of this work. In Table 2, a comparison of the advantages and disadvantages of integrating blockchain into the access control model is provided.

Table 1. Comparison of access control policies.

Scheme	Distributed	Flexibility	Dynamics	Fine-Grained
RBAC [6]	√ [1]	√	√	× [2]
ABAC [7]	√	√	√	×
CapBAC [8]	√	√	×	×
FGAC	√	√	√	√

[1] √ Means the performance is available in this framework. [2] × Means the performance is discrepant in this framework.

From the above two tables, we can see that the FGAC proposed in this paper has both flexibility and fine-grained access control. Compared with several traditional access control methods, it effectively improves the efficiency and security of the system.

Table 2. Integrating blockchain into an access control model.

Scheme	Characteristics
LBAC [32]	It proposes lightweight access control and uses smart contracts to ensure the correctness of outsourced decryption without additional verification on the user side but does not define role attributes.
BHEAC [33]	The blockchain-based token request mechanism allows users to request resources in batches and map the obtained tokens to multiple resources; it avoids repeated requests by users but has a broad division of permissions.
AI applications [15]	Only access control policies and data sharing protocols (DSAs) were designed to explain research strategies and research decisions, and no experiments and validation were performed.
FGAC of this work	It enhances role-based access control by providing different attribute keywords for different types of users. It is implemented in the form of smart contract Chaincodes and evaluated through quantitative verification.

5.2. Verification Results and Analysis

We used the model-checking tool UPPAAL [34] to verify and analyze the FGAC framework. The FGAC framework was modeled as the timed automata, which are shown in Figures 4 and 5, and the corresponding properties were specified as TCTL (timed computation tree logic), which are shown in Table 3. Our experiments were performed on a computer with an Intel (R) Core (TM) i5-9300HF CPU processor at 2.40 GHz, 2667 MHz, 4 cores, and 8 logical processors with 16 GB of RAM, running 64-bit Windows 10. The academic version 4.1.26 of the UPPAAL tool was used.

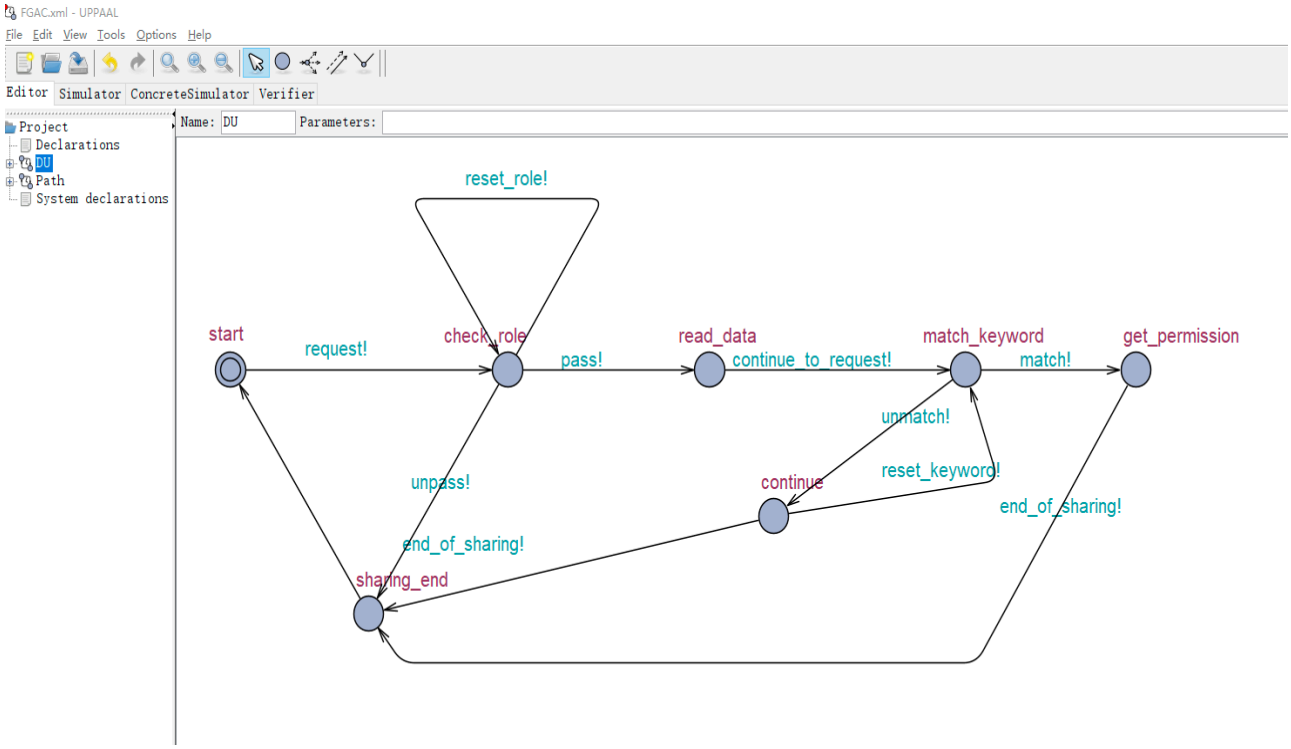

Figure 4. FGAC timed automata model.

Figures 5 and 6 above show the timed automata model and shared path of FGAC, respectively, which were built in UPPAAL. The FGAC model included the status of role checking, data reading, keyword matching, obtaining permission, etc. These statuses start from initiating a sharing request by the DU, after which the DU obtains the responding data through the assessment of the nodes in different statuses, and finally, the sharing process is completed.

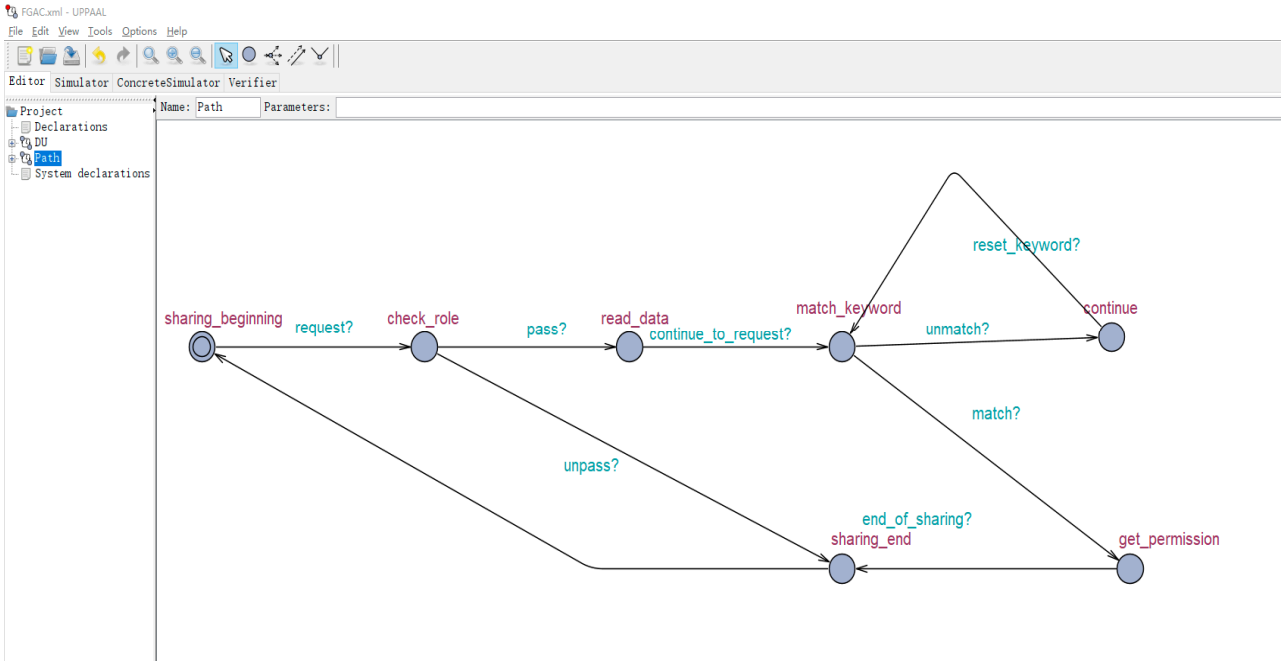

Figure 5. Shared pathway model.

Table 3. Property to be verified in UPPAAL.

Name	Property	Equivalent Property
Possibly	E<>η	
Invariant	A[]η	Not E<>notηη
Potentially always	E[]η	
Eventually	A<>η	Not E[]notηη
Leads to	η→Ψ	A[](ηimply A<>Ψ)

In order to verify the correctness of the model, after using UPPAAL to build the model, it was necessary to further extract the key attributes and verify the key attributes of the FGAC time automaton model. The main properties and corresponding expressions of validation were as follows:

1. The built model has no deadlock; Expression: A[] not deadlock
2. The data users in the above access control framework can normally access the status of read data; Expression: E<>DU1. read data
3. The data users in the above access control framework can normally access the status of obtaining permission; Expression: E<>DU1.get_permission

After triggering events for data sharing, UPPAAL displays the generated tracking trajectory, as shown in Figure 6.

After many experiments, the average value of the verification results we obtained is shown in Table 4.

Table 4 shows the verification results of the above properties in UPPAAL. From the verification results, it can be seen that the FGAC framework satisfies the aforementioned requirements and attributes, such as safety, no deadlock, etc. The FGAC framework could achieve data sharing normally, and access control was performed when user keywords did not match. In addition, it can also be determined whether the system model meets the requirements according to the verification time of the property formula and peak memory usage and can meet the usage time limit and resource constraints.

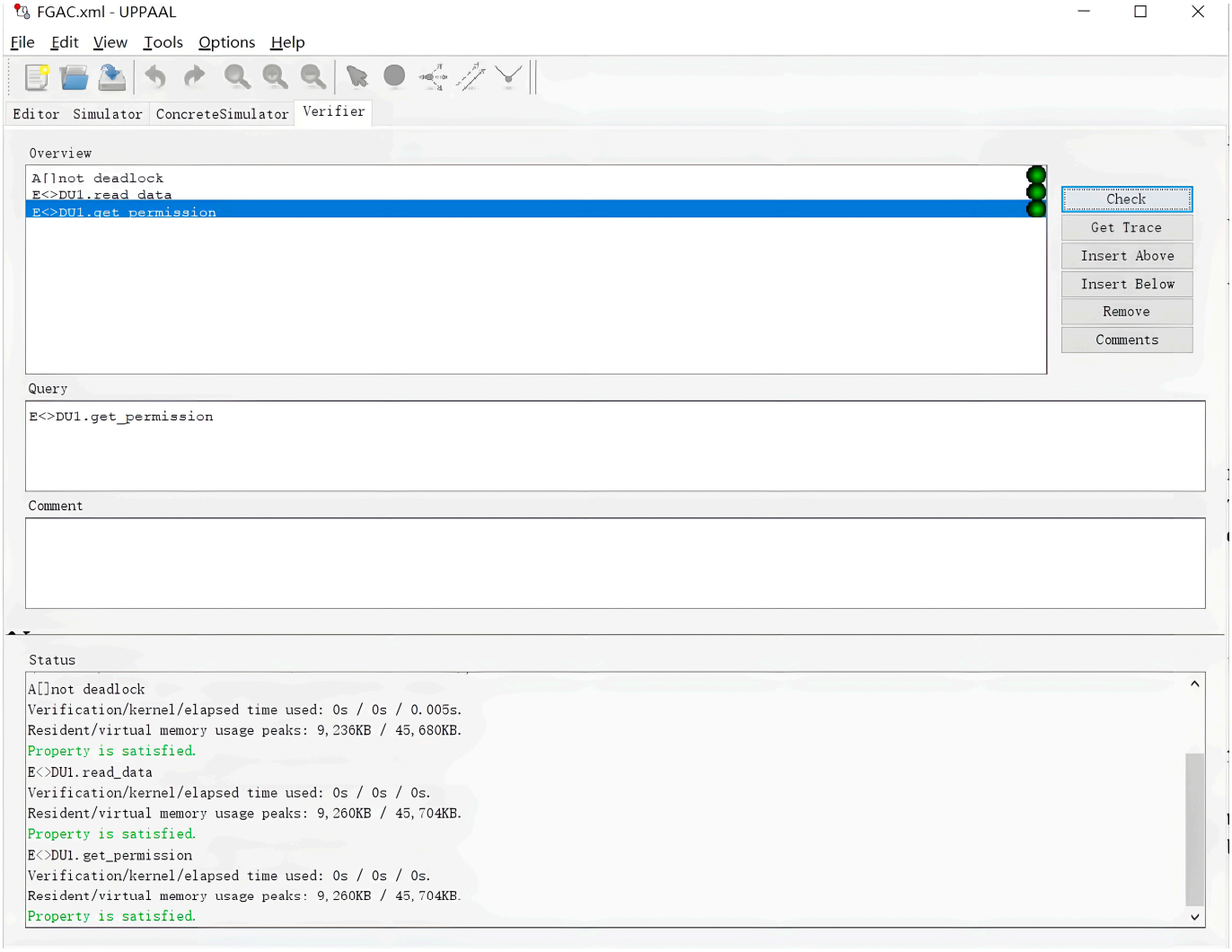

Figure 6. System verification results.

Table 4. System property validation results.

Property Formula	Validation Results	Verification Time	Peak Memory Usage
A[] not deadlock	Pass	0.001 s	9.306 KB
E<>DU1.read_data	Pass	0.001 s	9.236 KB
E<>DU1.get_permission	Pass	0.001 s	9.330 KB

The above comparative analysis and quantitative verification show that the proposed FGAC can effectively implement fine-grained access control in data sharing. It also has the characteristics of flexibility, high efficiency, and non-single-factor control, which effectively improves the security of data in the process of data sharing.

6. Conclusions and Future Work

Based on the blockchain Hyperledger Fabric, in this paper, we proposed a fine-grained access control (FGAC) framework for supply chain data sharing. It enhances role-based access control (RBAC) by providing different attribute keywords for different types of users. It was implemented in the form of smart contract Chaincodes of the blockchain Hyperledger Fabric and evaluated by the quantitative system verification tool UPPAAL. Moreover, it was applied to the supply chain of Shanghai Port to enhance data sharing security. In the future, we will apply the FGAC framework to more scenarios of supply chain data sharing and optimize its performance through system quantitative verification techniques.

Author Contributions: Conceptualization, Y.L. and X.L.; methodology, Y.L. and Y.M.; software, X.L.; validation, Y.L., X.L. and Y.M.; formal analysis, Y.L.; investigation, Y.L.; resources, X.L.; data curation, X.L.; writing—original draft preparation, Y.L. and X.L.; writing—review and editing, Y.L.; visualization, X.L.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MOE Humanities and the Social Sciences Foundation of China under Grant No. 20YJCZH102, and Singapore–UK Cyber Security of EPSRC under Grant No. EP/N020170/1.

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. Ghorbel, A.; Ghorbel, M.; Jmaiel, M. Accountable Privacy Preserving Attribute-Based Access Control for Cloud Services Enforced Using Blockchain. *Int. J. Inf. Secur.* **2021**, *21*, 489–508. [CrossRef]
2. Saini, A.; Zhu, Q.; Singh, N.; Xiang, Y.; Gao, L.; Zhang, Y. A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System. *IEEE Internet Things J.* **2021**, *8*, 5914–5925. [CrossRef]
3. Zhu, Y.; Ahn, G.-J.; Hu, H.; Ma, D.; Wang, S. Role-Based Cryptosystem: A New Cryptographic RBAC System Based on Role-Key Hierarchy. *IEEE Trans. Inf. Forensic Secur.* **2013**, *8*, 2138–2153. [CrossRef]
4. Hu, V.C.; Ferraiolo, D.; Kuhn, R.; Schnitzer, A.; Sandlin, K.; Miller, R.; Scarfone, K. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2014; p. NIST SP 800-162.
5. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, *4*, 2.
6. Matsumoto, S.; Reischuk, R.M. IKP: Turning a PKI Around with Decentralized Automated Incentives. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 25 May 2017; IEEE: Manhattan, NY, USA, 2017; pp. 410–426.
7. Das, D.; Banerjee, S.; Biswas, U. A Secure Vehicle Theft Detection Framework Using Blockchain and Smart Contract. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 672–686. [CrossRef]
8. Wood, D.G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
9. Ouaddah, A.; Elkalam, A.A.; Ouahman, A.A. Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT. In *Europe and MENA Cooperation Advances in Information and Communication Technologies*; Rocha, Á., Serrhini, M., Felgueiras, C., Eds.; Advances in Intelligent Systems and Computing; Springer International Publishing: Cham, Switzerland, 2017; Volume 520, pp. 523–533. ISBN 978-3-319-46567-8.
10. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 1594–1605. [CrossRef]
11. Miao, S.; Zhang, X.; Liu, Z. Fine-Grained Access Control Mechanism of Energy Internet. *Wuhan Univ. J. Nat. Sci.* **2022**, *27*, 231–239. [CrossRef]
12. Zikratov, I.; Kuzmin, A.; Akimenko, V.; Niculichev, V.; Yalansky, L. Ensuring Data Integrity Using Blockchain Technology. In Proceedings of the 2017 20th Conference of Open Innovations Association (FRUCT), St-Petersburg, Russia, 3–8 April 2017; IEEE: Manhattan, NY, USA, 2017; pp. 534–539.
13. Shah, M.; Li, C.; Sheng, M.; Zhang, Y.; Xing, C. CrowdMed: A Blockchain-Based Approach to Consent Management for Health Data Sharing. In *Smart Health*; Chen, H., Zeng, D., Yan, X., Xing, C., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11924, pp. 345–356. ISBN 978-3-030-34481-8.
14. Shamshad, S.; Minahil; Mahmood, K.; Kumari, S.; Chen, C.-M. A Secure Blockchain-Based e-Health Records Storage and Sharing Scheme. *J. Inf. Secur. Appl.* **2020**, *55*, 102590. [CrossRef]
15. Spanaki, K.; Karafili, E.; Despoudi, S. AI Applications of Data Sharing in Agriculture 4.0: A Framework for Role-Based Data Access Control. *Int. J. Inf. Manag.* **2021**, *59*, 102350. [CrossRef]
16. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
17. Shen, B.; Guo, J.; Yang, Y. MedChain: Efficient Healthcare Data Sharing via Blockchain. *Appl. Sci.* **2019**, *9*, 1207. [CrossRef]
18. Zheng, B.K.; Zhu, L.-H.; Shen, M.; Gao, F.; Zhang, C.; Li, Y.-D.; Yang, J. Scalable and Privacy-Preserving Data Sharing Based on Blockchain. *J. Comput. Sci. Technol.* **2018**, *33*, 557–567. [CrossRef]
19. Qin, X.; Huang, Y.; Yang, Z.; Li, X. A Blockchain-Based Access Control Scheme with Multiple Attribute Authorities for Secure Cloud Data Sharing. *J. Syst. Archit.* **2021**, *112*, 101854. [CrossRef]
20. Chi, J.; Li, Y.; Huang, J.; Liu, J.; Jin, Y.; Chen, C.; Qiu, T. A Secure and Efficient Data Sharing Scheme Based on Blockchain in Industrial Internet of Things. *J. Netw. Comput. Appl.* **2020**, *167*, 102710. [CrossRef]

21. Qi, S.; Lu, Y.; Zheng, Y.; Li, Y.; Chen, X. Cpds: Enabling Compressed and Private Data Sharing for Industrial Internet of Things Over Blockchain. *IEEE Trans. Ind. Inf.* **2021**, *17*, 2376–2387. [CrossRef]
22. Li, T.; Ren, W.; Xiang, Y.; Zheng, X.; Zhu, T.; Choo, K.-K.R.; Srivastava, G. FAPS: A Fair, Autonomous and Privacy-Preserving Scheme for Big Data Exchange Based on Oblivious Transfer, Ether Cheque and Smart Contracts. *Inf. Sci.* **2021**, *544*, 469–484. [CrossRef]
23. Ma, X.; Wang, C.; Chen, X. Trusted Data Sharing with Flexible Access Control Based on Blockchain. *Comput. Stand. Interfaces* **2021**, *78*, 103543. [CrossRef]
24. 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; IEEE: Manhattan, NY, USA; pp. 1027–1034.
25. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-Based Access Control Using Smart Contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
26. Kamboj, P.; Khare, S.; Pal, S. User Authentication Using Blockchain Based Smart Contract in Role-Based Access Control. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2961–2976. [CrossRef]
27. Yavari, A.; Panah, A.S.; Georgakopoulos, D.; Jayaraman, P.P.; van Schyndel, R. Scalable Role-Based Data Disclosure Control for the Internet of Things. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; IEEE: Manhattan, NY, USA, 2017; pp. 2226–2233.
28. Wang, X.; Jiang, X.; Li, Y. Model for Data Access Control and Sharing Based on Blockchain. *J. Softw.* **2019**, *30*, 1661–1669. [CrossRef]
29. Li, J.; Chen, N.; Zhang, Y. Extended File Hierarchy Access Control Scheme with Attribute-Based Encryption in Cloud Computing. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 983–993. [CrossRef]
30. Su, M.; Li, F.; Shi, G.; Li, L. An Action Based Access Control Model for Multi-Level Security. *Int. J. Secur. Its Appl.* **2012**, *6*, 359–366.
31. Kumar, S.; Bharti, A.K.; Amin, R. Decentralized Secure Storage of Medical Records Using Blockchain and IPFS: A Comparative Analysis with Future Directions. *Secur. Priv.* **2021**, *4*, e162. [CrossRef]
32. Qin, X.; Huang, Y.; Yang, Z.; Li, X. LBAC: A Lightweight Blockchain-Based Access Control Scheme for the Internet of Things. *Inf. Sci.* **2021**, *554*, 222–235. [CrossRef]
33. Chai, B.; Yan, B.; Yu, J.; Wang, G. BHE-AC: A Blockchain-Based High-Efficiency Access Control Framework for Internet of Things. *Pers. Ubiquitous Comput.* **2021**, *26*, 971–982. [CrossRef]
34. Behrmann, G.; David, A.; Larsen, K.G. A Tutorial on Uppaal. In *Formal Methods for the Design of Real-Time Systems*; Bernardo, M., Corradini, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3185, pp. 200–236. ISBN 978-3-540-23068-7.