Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7579961, 10 pages
https://doi.org/10.1155/2022/7579961

Research Article

A Metaheuristic Approach for Encrypting Blockchain Data Attributes Using Ciphertext Policy Technique

Nabamita Deb,[1] **Mohamed A. Elashiri**,[2] **T. Veeramakali**,[3] **Abdul Wahab Rahmani**,[4] **and Sheshang Degadwala**[5]

[1]*Department of Information Technology, Gauhati University, Gauhati, India*
[2]*Computer Science Department, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt*
[3]*Department of Data Science and Business Systems, School of Computing, SRM Institute of Science and Technology, Kattankulathur, India*
[4]*Isteqlal Institute of Higher Education, Kabul, Afghanistan*
[5]*Department of Computer Engineering, Sigma Institute of Engineering, Vadodara, Gujarat, India*

Correspondence should be addressed to Nabamita Deb; deb.nabamita@gmail.com and Abdul Wahab Rahmani; ab.wahab.professor@isteqlal.edu.af

Received 4 January 2022; Accepted 21 January 2022; Published 10 February 2022

Academic Editor: Vijay Kumar

Unlike public chains, the Alliance Blockchain Hyperledger Fabric has a member management service mechanism that may provide data isolation security at the channel level. However, because this data isolation security technique synchronizes plaintext data inside the channel, data leakage is still a possibility. Furthermore, in some fine-grained privacy protection circumstances, channel-based data access restriction is ineffective. In order to solve the data privacy security problems in the above-mentioned consortium chain superledger, a blockchain data attribute encryption scheme based on ciphertext policy is proposed. Combining the original Fabric Certificate Authority module in the Hyperledger, the proposed scheme can realize the user-level fine-grained security access to control blockchain data while also realizing the secure distribution of user attribute keys in the blockchain data attribute encryption scheme based on the ciphertext policy scheme. The security analysis of the scheme shows that the scheme achieves the security goals of attribute-based encryption user attribute private key secure distribution and data privacy protection. The scope of this research is that this study confirms that the solution's architecture achieves fine-grained access control of private data on the Hyperledger Blockchain network and also the security objectives of secure transmission of user characteristic secret keys and data privacy protection. The performance analysis part also shows that the proposed scheme has good usability.

1. Introduction

The Linux Foundation launched the Alliance Blockchain Hyperledger [1] blockchain technology project to develop cross-industry commercial blockchain platform technology. Hyperledger technology is a worldwide corporate blockchain initiative that provides the structure, rules, norms, and tools required for constructing open-source blockchains and related applications for usage in a variety of sectors. Blockchain technology is a method of storing information in such a way that it is difficult or impossible to edit, hack, or trick the system. Unlike well-known public chains like Bitcoin [2] and Ethereum [3, 4], this one is not well-known. The Hyperledger technology additionally integrates the member management service mechanism [5] and realizes identity management and network privacy that are more suitable for commercial use, confidentiality, review, and other functions.

By default, in a typical blockchain network, all data in the network are available to every node and user, posing data privacy and security problems. Particular sensitive data cannot be synchronized over the entire network in cleartext in some application circumstances. The alliance blockchain superledger has introduced support for multiple channels [6] to overcome the problem of security and privacy created by the transparency of network node data so that nodes in the same channel keep a log together and data in separate media is mutually exclusive and isolated. Through this multichannel data isolation mechanism, the Hyperledger technology greatly enhances the intensity of data privacy protection. But by default, the data in the channel is still evident to the nodes in the same track, so this mechanism still has the following problems: (1) Data leakage risk: once a node is compromised by an attacker, the plaintext data in it will be fully grasped by the attacker. (2) Data privacy protection granularity being too coarse: this channel-based coarse-grained data privacy protection method is not applicable in some fine-grained data access control scenarios. Therefore, a more fine-grained data security access control mechanism is needed.

Based on the coarse-grained data access and data encryption issues mentioned above, a corresponding solution was also proposed in the official version update of Hyperledger, that is, symmetric data encryption on the chain [7]. By symmetrically encrypting the plaintext data and then on the chain, in this way, only users with a symmetric decryption key can obtain the actual plaintext data. This solves the problem of fine-grained data privacy protection to a certain extent. However, in real applications, to achieve the requirements of fine-grained security access control, the scheme requires that each piece of data on the chain needs to maintain an independent key, and the key needs to be distributed to all recipients included in the access control policy. This process involves a large number of key generations, distribution, and management operations, making this scheme inefficient. Another solution officially proposed by Hyperledger is the privacy data mechanism [8]: transmitting and synchronizing real plaintext data among authorized organizational nodes and transmitting and synchronizing data hash values between unauthorized organizational nodes. The mechanism can refine the data privacy protection to the organizational level in the channel; that is, it can realize the private data sharing between certain organizations in the channel, which solves the problem of this paper to a certain extent. Since the data synchronized between authorized nodes is still plaintext data, this mechanism does not fundamentally solve the security problems caused by data leakage. In addition, as mentioned above, the privacy data mechanism can only refine the access control of data to the organizational level. Therefore, in some scenarios that hope to achieve access control based on users in the organization, this mechanism does not meet the needs of this paper.

This paper focuses on the current fine-graininess of Hyperledger Fabric data privacy protection requirements; a BES-CP-based algorithm is designed. The main contributions of the blockchain data access control scheme are as follows:

(1) Based on the BES-CP algorithm, a method suitable for the data access control mechanism of the Fabric network of the ledgers guarantees data is not leaked while achieving the finest user-based granularity degree of data access control.

(2) Based on the existing Fabric-CA module in Hyperledger, realize the dynamic generation and security analysis of BES-CP user attribute keys sent and other operations. Without affecting the original structure and operating machine of Hyperledger under the premise of the system, the user's attributes are encrypted through an asymmetric encryption algorithm key for encrypted transmission, which solves the widespread storage in traditional encryption schemes in the key distribution problem.

The next section discusses the background techniques that have been implemented, followed by the design that has been introduced in this research. After that, there is a discussion on the specific plan and its evaluation for this research. Finally, the paper has been concluded.

2. Background Technique

2.1. Blockchain and Hyperledger. The concept of a blockchain originated from Bitcoin [2], which effectively solved the problem of trust between nodes in decentralized distributed scenarios [9]. In essence, the blockchain is a special data structure. The blockchain organizes data blocks into a chain by combining multiple cryptographic technologies and distributed network technologies. The work of the Cryptographic Technology (CT) Unit in cryptographic mechanisms includes hash algorithms, symmetric and asymmetric cryptographic approaches, key management, authentication, and random number generation. Distributed networking, as used in dispersed computing, is a network system in which computer programming, software, and data are decentralized over several computers yet convey complex messages through their nodes (computers) and are reliant on one another. At the same time, the blockchain uses point-to-point transmission technology and consensus mechanisms (such as PoW [10] and PoS [11, 12]) to maintain the same data content among multiple nodes and uses this redundancy mechanism to achieve data immutability and durability. The blockchain network strictly follows the mechanism of "less number obeys the majority" to ensure the consistency of data between nodes, which means that the attacker must control more than half of the blockchain nodes (51% attack [13]) to achieve illegal modification or deletion. The existing records are on the chain. It is almost impossible for this type of attack to be deployed in a distributed blockchain network on a large scale. Therefore, it is generally believed that the data stored in the blockchain is safe. At the same time, with Ethereum, the integration of smart contracts for the representative blockchain network makes it possible to realize more complex and advanced distributed applications [14]. Because blockchain technology is decentralized, the data on the chain cannot be tampered with, and the transaction cannot be tampered with. Academia and the industry have carried out a lot of research work on the characteristics of content traceability and smart contract implementation [15]. At present, blockchain technology is widely used in financial services, credit investigation and ownership management, resource sharing, supply chain management, privacy protection, and public network services [16, 17].

According to the system control rights and whether the transaction information is disclosed or not [18, 19], blockchain can be divided into three categories: public chains, private chains, and consortium chains. A public blockchain is a decentralized platform that anybody may access. Private blockchains are frequently referred to as "permissioned" blockchains. Private blockchains are frequently maintained and operated by an entity. A consortium blockchain, also called a federated blockchain, is similar to a hybrid blockchain in that it has both private and public blockchain capabilities. The public chain does not have any access mechanism. Any node can join the network, and the information is open to the entire system. It is an entirely decentralized peer-to-peer system, such as Bitcoin and Ethereum. Private chains are currently primarily used in test scenarios. They are not essentially different from databases in the ordinary sense and are only suitable for limited institutions. The consortium chain refers to the particular access mechanism designed in the blockchain, transaction information is only disclosed locally, and the blockchain nodes usually need to pass legal certificates to initiate transactions or access content on the blockchain. This design is particularly suitable for commercial application scenarios. While protecting business privacy, it can solve the problem of mutual trust between business partners across institutions.

As the representative technology of the alliance blockchain, the hyperledger blockchain is not entirely decentralized but has several organizations to participate and manage, and a dedicated certificate authority Fabric-CA is configured to handle the nodes or users.

Compared with traditional public chain technology, Hyperledger has greatly improved in terms of the scalability and functional integrity of smart contracts. For example, Hyperledger supports the Byzantine consensus protocol and the Kafka-based crash and fault-tolerant consensus protocol, achieving fast and effective transaction consensus. In addition, Hyperledger supports intelligent contracts in multiple conventional programming languages, such as Go, Node.js, and Java. This also means that the smart contracts deployed on the Hyperledger blockchain can theoretically support any function. As a result, Hyperledger will not be limited by programming implementation issues when supporting commercial applications.

To decouple the function and improve scalability, two types of nodes are designed in Fabric: Ordered and Peer. Sorting nodes are responsible for sorting transactions according to rules and generating blocks. Multiple sorting nodes can form a sorting cluster. The consensus among ordering nodes can use the PBFT (Practical Byzantine Fault Tolerance) algorithm [20] or the crash fault tolerance algorithm based on Kafka [21]. Peer nodes are responsible for verifying and synchronizing data and performing functions such as smart contracts (also known as chain codes in Fabric). According to different node functions, peer nodes can be divided into endorsement nodes, accounting nodes, master nodes, anchor nodes, and so on, and node identities can be switched based on specific rules. Loose coupling of functions and realization of modularization ensure that the Fabric network has good scalability.

2.2. Introduction to Fabric-CA. The Hyperledger Blockchain network belongs to the permission chain type. When new users need to access the network, they need to obtain a legal certificate from a special organization. Fabric-CA [22] is used to implement all the functions of this organization. Fabric-CA is composed of server (Fabric-CA server) and client (Fabric-CA client) components. The Fabric-CA client command helps in managing the identities, while the Fabric-CA server helps in developing the connection between the identities. It provides three functions: one is to register a new user identity, and the registration result will be used as a credential for the user certificate application; the second is to issue a certificate, which is to generate a certificate for a legal registration ticket; the third is to renew or revoke a certificate, as a user in the Fabric network. When the information is updated, this function will update the certificate information of the corresponding user.

New users can interact with the Fabric-CA server through an independent Fabric-CA client or an SDK program integrated with the client, and all communication is carried out through the REST API.

2.3. Hyperledger Transaction Process. A typical Hyperledger Fabric transaction process is discussed. Before joining the Hyperledger network, users can register with Fabric-CA, obtain a legal certificate, and then use the certificate to interact with the blockchain network through the command line or Fabric-SDK. Hyperledger supports a multichannel mechanism. Each channel maintains an independent blockchain ledger. Blocks are distributed according to the channel ID, and data between channels is completely isolated. The user needs to specify the channel ID (Channel 1 or Channel 2) when initiating a transaction and send the transaction proposal to the endorsing node. After the endorsement node processes the request, the client returns a signed endorsement response. The client then combines the endorsement responses from different endorsement nodes and sends the final transaction content to the ranking service cluster for processing. After the sorting is completed, the sorting node will distribute it to all master nodes in the channel according to the transaction channel. The master node will synchronize the blocks in the organization. After receiving the transaction, each node verifies the transaction content and signature and adds the legal transaction to the blockchain ledger.

2.4. Attribute-Based Encryption. Attribute-based encryption (ABE) was first proposed by Goyal et al. [23] based on FIBE (fuzzy identity-based encryption) [24]. The purpose is to solve the problem of fine-grained access control of data in the cloud storage environment, the problem of large-scale user dynamic expansion [25]. ABE is essentially an asymmetric encryption technology, but it uses one-to-many encryption. ABE is separated into key policy attribute-based encryption (KP-ABE) [23] and ciphertext policy attribute-based encryption (BES-CP) [26] based on the location of the decryption method. The ciphertext decryption strategy is embedded in the user's private key, and the relevant attributes are embedded in the ciphertext during encryption, so the access strategy is associated with the key; in BES-CP, the decryption strategy is embedded in the ciphertext, and the user's attributes are embedded in the private key when the key is generated, so the access policy is associated with the key. Users can only decrypt the ciphertext when the private key and the attribute set encoded in the ciphertext completely meet the access control policy [27], regardless of whether they utilize KP-ABE or BES-CP. Because access control policies and attribute sets may have a one-to-many relationship, ABE technology provides encrypted access control functionalities by default. At the same time, the ABE scheme can determine the granularity of the ciphertext access control mechanism based on the strictness of the encryption or key generation approach. Consider the BES-CP scheme for the above-said problem.

3. Design

3.1. Overall Program Framework. Aiming at the problem of blockchain data access control proposed in Section 1, this paper proposes a scheme based on the BES-CP algorithm to implement blockchain data access control. Through the Client, the user interacts with the Fabric-CA and the Fabric network, which primarily consists of registration with the Fabric-CA to obtain the appropriate certificate and the user attribute secret key of the BES-CP scheme, and then uses the obtained private key and user-specified access control. (1) Modify the original user certificate management organization Fabric-CA to realize the initialization of the BES-CP scheme and the generation and distribution of the user attribute private key. (2) The private data is encrypted on the chain through the BES-CP scheme to achieve private data encrypted access control mechanism. The scheme is mainly composed of the certificate authority Fabric-CA, the blockchain network Fabric, and the client. The overall scheme framework is shown in Figure 1.

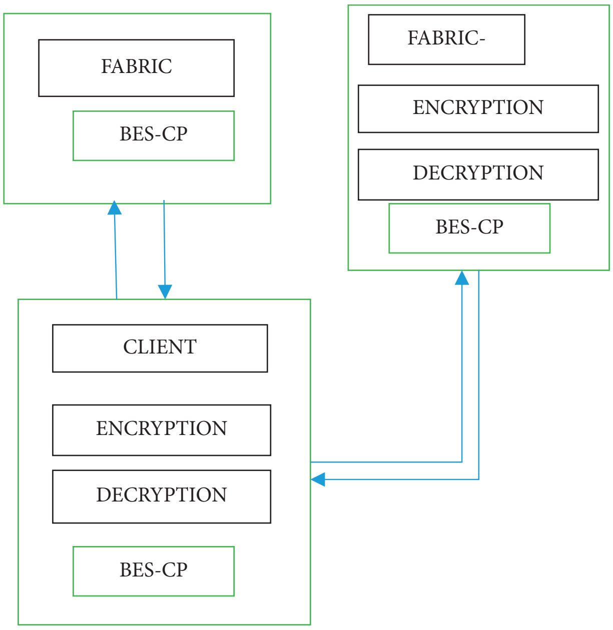

Figure 1: The framework of the scheme.

3.1.1. Fabric Part. This part mainly implements the original functions in the Hyperledger network, including transaction endorsement, transaction sequencing, transaction verification, and chaining. The division is mainly to pass the ciphertext to be stored through endorsement, sorting, and nodes to verify the chain operation and divide the ciphertext Fabric storage in the entire channel.

Client part: in the original Hyperledger network, this part is mainly used to implement transaction initiation and user certificate access. The solution in this paper adds user attribute key SK reception and BES-CP data encryption and decryption functions on top of its original functions. Users can interact with the Fabric network and Fabric-CA through the Client.

Table 1: Definition of user attribution.

Attribute	Attribute value
Channel ID	Channel 1; Channel 2
Organization ID	Org1; Org2
User ID	User1; User2

3.2. BES-CP Attribute Set and Policy Definition

3.2.1. Attribute Definition. Only the key that meets the policy attributes in the BES-CP scheme can decrypt the ciphertext to obtain the plaintext data. As can be seen from Section 2.1, the Hyperledger network can include multiple channels (such as Channel 1, Channel 2, and Channel 3). DAO will contain multiple organizations (such as Org1, Org2, and Org3); each organization will have multiple users (such as User1, User2, and User3), dividable as a Hyperledger Blockchain network. The smallest unit, the user, is the smallest in the BES-CP solution attribute set granularity and three inherent: channel ID, organization ID, and user ID, attributes; therefore, this paper will define the attributes in the data access control. The optional range is shown in Table 1.

3.2.2. Access Control Policy Definition. A strategy is an access structure composed of attributes. As can be seen from Section 3.2.1, this paper defines three attributes: channel ID, organization ID, and user ID. In principle, these three attributes can be combined as well when forming an access control policy. For example, in the actual ciphertext, when generated, policy P1 can be defined as User1 in Channel 1 Organization 1, and all users in Channel 2 Organization 2 can access, and P2 is defined as User1 in Channel 1 Organization 1, and all users in Channel 1 Organization 2 can access. By combining the access control strategy and ciphertext, the purpose of data access control can be achieved.

The block distribution of the Hyperledger network is distributed according to the channel ID. Because the channels are isolated from each other, the data block can only be sent to one of the channels, and there is usually no data on the chain. It can be accessed by multiple channels at the same time. Therefore, the aforementioned policy P1 does not exist in actual access control; that is, the attribute of the channel ID cannot be included in the access control policy. However, when the channel ID attribute is not considered as the choice of the access policy, this paper defines the policy as only including one or a combination of the organization ID attribute or the user ID attribute. For example, a certain ciphertext in Channel 1 specifies the policy P as (organization ID = Org2), once the user User1 in Channel 1 Organization 2 will leak the obtained ciphertext to User2 in Channel 2 Organization 2 because user User2 also contains the attribute Org2, so User2 can successfully decrypt the ciphertext, causing data leakage. Then, when the data owner wants to make the encrypted ciphertext decrypt able to all users in Channel 1, without considering the channel ID attribute as the access strategy choice, it is necessary to define multiple organizations to perform and operations, that is, the logic of the strategy. It is more complicated than directly defining the strategy P as (channel ID = Channel 1). Therefore, in order to more accurately implement data access control for encrypted data, the designation of user policies needs to be considered at the channel level.

In order to make the available access control strategy more comprehensive and specific comprehensive display, this paper defines three types of overall strategies:

(a) All users in a specific channel can decrypt

(b) All users of a specific organization in a specific channel can resolve secrets

(c) A specific user in a specific organization in a specific channel can decrypt it

Among them, the symbol * in Channel*, Org*, and User* means $\{*| * \in(1,2,3,4)\}$. Channel* refers to the collection of all channels in the Hyperledger network {Channel 1, Channel 2...}, Org* refers to the collection of all organizations under a specific channel {Org1, Org2...}, and User* refers to users in a specific organization under a specific channel set {User1, User2...}.

(i) General Strategy 1. Channel ID = Channel*; that is, the channel ID can be equal to any channel in the {Channel*} set. In other words, all users under this channel can decrypt the ciphertext encrypted based on this strategy to access the corresponding plaintext data.

(ii) General Strategy 2. (Channel ID = Channel 1) and (Organization ID = Org*); that is, the channel ID is specified. This can only be Channel 1 (but not limited to Channel 1, only a specific example), and the organization ID can be in Channel 1. Select one of the organization set {Org*} arbitrarily. At this time, all users under the organization can decrypt the ciphertext encrypted based on the policy to access the corresponding plaintext data.

(iii) General Strategy 3. (Channel ID = Channel 1) and (Organization ID = Org1) and (User ID = User*); that is, the channel ID is specified here, which can only be Channel 1, and the organization ID here only corresponds to Org1 in Channel 1 (it is not limited to Org1, only an example), and the user ID can be selected arbitrarily in the user set {user*} formed in Org1 in Channel 1. At this time, a user under Org1 in Channel 1 can decrypt the ciphertext encrypted based on this policy to access the corresponding plaintext.

When the user encrypts the data on the chain, the policy designation can directly use the above three general policies and only need to change the corresponding field in the module to the field that the user wants to specify. In addition, users can also combine general strategy 2 and general strategy 3 to achieve more fine-grained data access control.

4. Specific Plan and Evaluation

4.1. Specific Plan. The existing Fabric and Fabric-CA can be realized by embedding BES-CP in Fabric-CA, which is equivalent to Fabric-CA as a trusted third party in the BES-CP solution. At this time, Fabric-CA manages the original. There are certificates required by users in the Fabric network, the initialization of the BES-CP scheme, and the generation and distribution of the user attribute private key SK. The user realizes the interaction with the Fabric-CA and Fabric network through the Client, which mainly includes registering with the Fabric-CA to obtain the corresponding certificate and the user attribute private key of the BES-CP scheme and using the obtained private key and user-specified access control. The strategy encrypts the plaintext data and then sends the ciphertext to the Fabric network in the form of transactions for on-chain storage of the encrypted data.

The operation process can be roughly divided into 3 stages: key generation stage, data encryption on-chain stage, and access control stage, including the 6 steps of Setup, KeyGen, Encrypt, Update, Download, and Decrypt. In order to be able to show the specific process of the scheme more objectively, this paper uses the interaction between UserA and UserB as (UserA wants to encrypt private data on the chain; UserB wants to be able to access the plaintext corresponding to the ciphertext), assuming that UserA passes ClientA and UserB interacts with Fabric network and Fabric-CA through ClientB. The program symbol description is shown in Table 2.

Key generation phase: this phase is mainly the communication phase between the user and Fabric-CA. This phase generates the master key MK and public parameter PK in the BES-CP scheme and generates it according to the user registration request sent in the user identity authentication phase. The ciphertext SK CT corresponds to the user certificate Ucert and the user attribute private key SK.

4.1.1. Setup(1λ)$\longrightarrow$(MK, PK). This phase is consistent with the initialization phase in the BES-CP scheme. By entering the system security parameter λ, the master key MK and the public parameter PK in the BES-CP scheme are generated.

Table 2: The description of symbols.

Symbol	Symbol definition
SK	The attribute private key in the BES-CP scheme corresponding to the user
MK; PK	The master key and public parameters in the MK, PK BES-CP scheme
$\mathrm{U_{pk}}$; $\mathrm{U_{sk}}$	The public and private keys of users in the original Hyperledger network of USK
$\mathrm{U_{cert}}$	User certificate in the original Hyperledger network
M; CT	Plaintext data and encrypted ciphertext

4.1.2. SK KeyGen(MK, UCR)⟶(CP, Ucert). UCR (user certificate request) is a request for a user certificate, which contains the public key PK U corresponding to the certificate to be generated by the user and the attribute S of the user. In the original Hyperledger network, users send UCR to Fabric-CA through Client to apply for a certificate register; Fabric-CA signs the user's certificate request, generates a user certificate cert U, and returns the certificate to the user. Different from the original Fabric-CA, in the scheme of this paper, when Fabric-CA receives the request, it not only needs to generate the corresponding user certificate cert U but also needs to generate the user decryption key (i.e., SK). In order to ensure the security of the user attribute key SK, SK cannot be directly transmitted in the network. Therefore, based on the original Fabric-CA, this paper uses the public key PK U corresponding to the user certificate in the UCR to compare the newly generated user attributes.

The private key SK is encrypted to obtain the ciphertext SK CT corresponding to the key. Only the user who has the private key SK U corresponding to the public key PK U contained in the user certificate can successfully obtain the user attribute private key SK after receiving the SK CT. After SK CT is generated, Fabric-CA returns SK cert (CT, U) to the user, waiting for the user to continue subsequent operations.

Data encryption and chaining phase: this phase is mainly the phase of interaction between UserA and the Fabric network. UserA specifies the encryption strategy AP and the plaintext data AM to be chained and encrypts the data based on the strategy to obtain the ciphertext CTA, then pass and initiate a transaction, and send the ciphertext as a form of transaction load, that is, Tx(CT) A, to the blockchain to synchronize the ciphertext data with the blockchain network. The detailed steps are as follows.

4.1.3. Encrypt(PK,,) CT A AA M P ⟶. Similar to the data encryption steps in the original BES-CP scheme, UserA first uses the Encrypt algorithm to encrypt the specified plaintext message AM under the policy AP and uses the public parameter PK in the BES-CP scheme to encrypt the data to obtain the ciphertext CTA. The user initiates a transaction to the blockchain network and sends the ciphertext to the blockchain network as the transaction load, namely, transaction Tx(CT) A.

4.1.4. Update(Tx(CT)) Block A ⟶. After receiving the transaction containing ciphertext data submitted by UserA after Tx(CT) A, the Fabric network will first follow the endorsement policy to Tx(CT) A endorses, mainly on the transaction proposal format, transaction submit repeatability, transaction signature, and transaction submitter authority for verification and then simulate the execution of chain code to generate an endorsement response and return it to UserA; after collecting enough endorsement responses, Users can only decrypt the ciphertext when the private key and the attribute set encoded in the ciphertext completely meet the access control policy [28], regardless of whether they utilize KP-ABE or BES-CP. Because access control policies and attribute sets may have a one-to-many relationship, ABE technology provides encrypted access control functionalities by default. At the same time, the ABE scheme can determine the granularity of the ciphertext access control mechanism based on the strictness of the encryption or key generation approach. Consider the BES-CP scheme, which typically contains the following:

(a) Verification of transaction data, including verification of transaction format, transaction signature, and whether the transaction content has been tampered with.

(b) Chain code verification, including the chain code involved in the transaction, whether the information is empty and whether there is a verification of the illegal call chain code.

(c) The verification of the status data includes the verification of the consistency of the status data when the simulation is executed and the status data when the transaction is submitted. To complete the on-chain storage of encrypted data, the block BlockA is put on each node in the same channel of the blockchain network when the verification is successful. It should be emphasized that only the format, signature, and state consistency of on-chain transactions are validated by Hyperledger throughout the endorsement and verification process, and the legality of the exact data in the transaction is not verified. The chain data itself is transparent to the bottom layer of the Hyperledger, so the feasibility of the ciphertext data on the chain can be guaranteed.

Access control phase: this is mostly the stage where the Fabric network and UserB communicate. After the block containing the ciphertext transaction is uploaded to the chain, it will be synchronized by all nodes in the channel. To retrieve the appropriate ciphertext, UserB uses the client to request the associated information of the transaction ciphertext contained in the blockchain network. CTA through text. Then using the previously obtained UserB attribute key ciphertext SK CT B, the user first decrypts SK CT B to obtain the user attribute key SKB of the BES-CP scheme, further decrypts the ciphertext CTA based on SKB to obtain the plaintext AM, and implements user-level data access control. The detailed implementation process is as follows:

(1) *Download(Tx(CT)) CT AA ⟶.* UserB interacts with the Fabric network through clientB, requesting that the blockchain network contains the corresponding information of the transaction ciphertext. Thereby, the corresponding data ciphertext A CT is obtained.

(2) *Decrypt(CT,SK,PK) A BA ⟶M.* In the key generation stage, in order to ensure the security of transmitting the user attribute private key SK, this paper uses the attribute private key SKB.

The public key PKB U in the original network of the user is encrypted to generate SK CT B for transmission. After receiving SK CT B, UserB will decrypt the ciphertext to generate a plaintext attribute private key SKB, and if the private key attribute satisfies when the strategy AP included in the CTA is used, SKB is used to decrypt the CTA to obtain the plain text AM corresponding to the encrypted data, which realizes the blockchain data access control between UserA and UserB based on the BES-CP scheme under normal circumstances.

4.2. Safety Analysis. The security analysis of this scheme mainly includes the secure distribution of the user attribute private key SK and the guarantee of data privacy as follows:

(1) The secure distribution of the user attributes private key SK. First of all, as a module in the original Hyperledger, Fabric-CA is completely trustworthy, thus ensuring the trustworthiness of the initial generation phase of the master key MK and PK and the user attribute information. Reliable audit and user attribute private key SK generation process are credible. However, since Fabric-CA does not have a native key distribution mechanism, the solution in this article uses the public key.

(2) The key PK U encrypts SK to generate SK CT and then distributes SK CT to the mechanism of the corresponding user, thereby ensuring the transmission security of the user attribute private key SK. Because when each user initiates a user certificate request (UCR), the user will put his public key PK U into the corresponding UCR, Fabric-CA will dynamically generate the user attribute private key SK after receiving the certificate request and then pass the UCR. The included PK U encrypts SK. At this time, only the user who has the private key SK U corresponding to the public key PK U can successfully decrypt SK CT and obtain SK. Other users, even if they get SK CT, cannot solve it. To achieve the purpose of stealing, the user attribute uses private key SK.

(3) Data privacy guarantees that the data owner encrypts the plaintext data with the BES-CP scheme and then uploads it to the chain, so only the ciphertext of the data is visible in the entire Fabric network. In addition, as mentioned earlier, Fabric-CA is completely credible, and the corresponding key generation stage is credible. The data owner directly restricts the access rights to the data by specifying the policy P. Only users who meet the attributes of the policy P can decrypt the data and then access it. The private key that does not meet the decryption attributes of the policy P cannot decrypt the ciphertext, thereby ensuring data privacy.

4.3. Performance Analysis. This section mainly analyzes the implementation performance of the scheme. The experimental environment uses the Ubuntu16.04 LTS virtual machine installed by Oracle VM VirtualBox and is allocated 4 GB of memory and 1 core processor. When selecting performance indicators, considering that the solution in this paper did not modify the transaction process of the original Hyperledger network, it just replaced the plaintext data on the chain in the original blockchain network with the BES-CP encrypted password. This update is transparent to the underlying transaction process and will not impair the Hyperledger network's original operational efficiency. Therefore, it is only necessary to measure the performance indicators of the BES-CP scheme. This section mainly measures the generation time and encryption and decryption time of the user attribute private key for the BES-CP scheme implemented in literature [26] as shown in Figure 2.

In the scheme designed in this paper, the attribute private key involved in the BES-CP scheme contains at most 3 attributes: channel ID, organization ID, and user ID. It can be seen from the literature [23] that the generation time of the private key increases linearly with the increase of the number of attributes included. In the scheme of this paper, considering that the number of attributes is at most 3, it is measured that the generation time of the private key is about 0.0315 s. At the same time, considering that, in the actual Hyperledger transaction process, a single block can hold up to 10 MB of data, so the maximum data size in the experiment is set to 10 MB. As shown in Figure 2, as the size of the data to be encrypted and decrypted increases, the corresponding encryption and decryption time increases in a linear trend. When the data size is 10 MB, the encryption time is about 0.08 s, and the decryption time is about 0.065 s.

The impact of the number of attributes versus User is shown in Figure 3. It is discovered that the time attained by these two operations is within an acceptable range of increments by measuring the impact of the user attribute private key generation time and the size of the data to be encrypted and decrypted on the encryption and decryption time. As a result, the original Hyperledger network implements a blockchain data access control mechanism based on the BES-CP algorithm, which is quite feasible.

4.4. Comparative Analysis of Plans. This section compares the proposed solutions with the existing official Hyperledger solutions and compares them from user-level access control granularity, data privacy, key security distribution, and easy

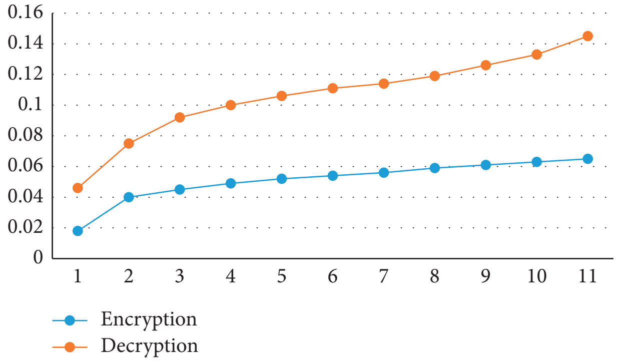

Figure 2: Impact of data size on encryption and decryption time.

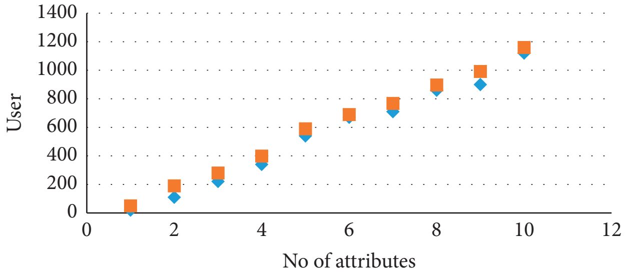

Figure 3: Impact of the number of attributes versus User.

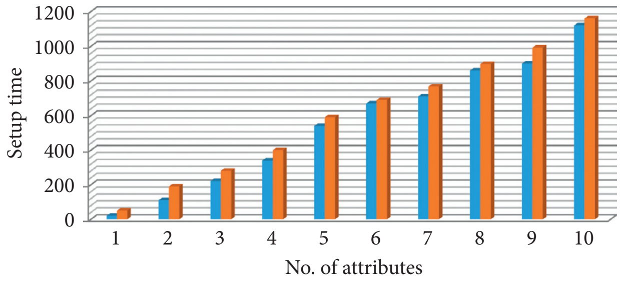

Figure 4: Comparative analysis.

Table 3: The comparison of safety and performance.

Type of solution	User-level access control	Granularity	Data privacy	Key security distribution	Encryption/decryption key management
Scheme [7]	Symmetric encryption	✓	✓	X	X
Scheme [8]	Hash synchronization	X	X	—	—
Proposed scheme	Attribute-based encryption	✓	✓	✓	✓

encryption/decryption key management. Among them, the user-level access control granularity indicates whether the scheme supports the finest access control granularity of the Hyperledger network, that is, the user-level access control function; data privacy represents whether the scheme guarantees the privacy of the data on the chain; key security distribution explains whether the scheme guarantees the secure distribution of keys; the ease of encryption/decryption key management represents whether users do not need to manage and maintain multiple encryption/decryption keys under the premise of ensuring the security of encrypted ciphertexts, that is, whether they can get rid of them. The complex operation of “one secret one key” is used.

Comparative analysis is also performed, as shown in Figure 4, for the proposed solutions with the existing official solutions and compares them for setup time with respect to the number of attributes.

As shown in Table 3, compared with similar solutions, this solution can not only guarantee the privacy of data on the chain and support the finest access control granularity based on the user level but also simplify the encryption/decryption key under the premise of ensuring the secure distribution of the key. Therefore, it is more suitable for the realization of on-chain data access control of the Hyperledger network.

5. Conclusion

This paper examines the current Hyperledger Blockchain network's clear text data storage and coarse-grained data access control mechanism and proposes a blockchain data access control scheme based on the BES-CP algorithm that implements a fine-grained data access control scheme based on user attributes. The control target should be asked for granular access. At the same time, this paper is based on the original Fabric-CA module in the Hyperledger blockchain, and by adding support for the generation of BES-CP algorithm keys (including system initialization and user attribute private keys), the secure distribution of user attribute private keys is realized. Finally, the security analysis of the solution proposed in this paper verifies that the design of the solution has achieved fine-grained access control of private data in the Hyperledger Blockchain network, as well as the security goals of secure distribution of user attribute private keys and data privacy protection. In this paper, the Performance Analysis Department also mentions the program's usefulness. At the same time, it demonstrates the relevance of the design scheme in this study to the realization of on-chain data access control of the Hyperledger network by comparing it to current equivalent schemes.

Data Availability

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

Conflicts of Interest

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

References

[1] D. Hu, H. Zhou, T. Ma, K. Yu, and N. Cheng, "An evolutionary game assisted spectrum sharing blockchain framework for internet of vehicles," in *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–6, IEEE, Victoria, BC, Canada, November–December 2020.

[2] S. Hong and H. Kim, "Analysis of Bitcoin exchange using relationship of transactions and addresses," in *Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT)*, pp. 67–70, IEEE, Pyeong-Chang, Korea (South), February 2019.

[3] Y. Huang, B. Wang, and Y. Wang, "MResearch on Ethereum private blockchain multi-nodes platform," in *Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 369–372, IEEE, Fuzhou, China, June 2020.

[4] D. Son, S. Al Zahr, and G. Memmi, "Performance analysis of an energy trading platform using the Ethereum blockchain," in *Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–3, IEEE, Sydney, Australia, May 2021.

[5] M. Ahmed, S. Reno, N. Akter, and F. Haque, "Securing medical forensic system using hyperledger based private blockchain," in *Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, Dhaka, Bangladesh, December 2020.

[6] H. Yusuf, I. Surjandari, and A. M. M. Rus, "Multiple Channel with crash fault tolerant consensus blockchain network: A case study of vegetables supplier supply chain," in *Proceedings of the 2019 16th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–4, IEEE, Shenzhen, China, July 2019.

[7] H. Guo, W. Li, E. Meamari, C.-C. Shen, and M. Nejad, "Attribute-based multi-signature and encryption for EHR management: A blockchain-based solution," in *Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–5, IEEE, Toronto, ON, Canada, May 2020.

[8] Y. Cui, S. Li, Y. Wang, and B. Gao, "The data protection of intelligent connected vehicles cloud control framework using fully homomorphic encryption," in *Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pp. 19–24, IEEE, Hangzhou, China, December 2020.

[9] A. Demir, B. N. Akilotu, Z. Kadiroğlu, and A. Şengür, "Bitcoin price prediction using machine learning methods," in *Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–4, IEEE, Ankara, Turkey, November 2019.

[10] P. Alemany, R. Vilalta, R. Munoz, R. Casellas, and R. Martinez, "Blockchain-based connectivity provisioning in multiple transport SDN domains," in *Proceedings of the 2021 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–3, IEEE, Gothenburg, Sweden, 28 June-1 July 2021.

[11] L. Wan, D. Eyers, and H. Zhang, "Evaluating the impact of network latency on the safety of blockchain transactions," in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 194–201, IEEE, Atlanta, GA, USA, July 2019.

[12] F. Chen, Z. Li, B. Li et al., "Blockchain-based optical network slice rental approach for IoT," in *Proceedings of the 2020 IEEE Computing, Communications and IoT Applications (ComComAp)*, pp. 1–4, IEEE, Beijing, China, December 2020.

[13] G. Wang, Z. Shi, M. Nixon, and S. Han, "ChainSplitter: towards blockchain-based industrial IoT architecture for supporting hierarchical storage," in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 166–175, Atlanta, GA, USA, 2019.

[14] X. Huang, V. Jagota, E. Espinoza-Muñoz, and J. Flores-Albornoz, "Tourist hot spots prediction model based on optimized neural network algorithm," *International Journal of System Assurance Engineering and Management*, 2021.

[15] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of blockchain and Internet of Things in healthcare and medical sector: applications, challenges, and future perspectives," *Journal of Food Quality*, vol. 2021, pp. 1–20, 2021.

[16] S. Deshmukh, K. T. Rao, and M. Shabaz, "Collaborative learning based straggler prevention in large-scale distributed computing framework," *Security and Communication Networks*, vol. 2021, Article ID 8340925, 9 pages, 2021.

[17] B. Wang, X. Yao, Y. Jiang, C. Sun, and M. Shabaz, "Design of a real-time monitoring system for smoke and dust in thermal power plants based on improved genetic algorithm," *Journal of Healthcare Engineering*, vol. 2021, Article ID 7212567, 10 pages, 2021.

[18] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: a comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020.

[19] J. Bhola, M. Shabaz, G. Dhiman, S. Vimal, P. Subbulakshmi, and S. Kumar Soni, "Performance evaluation of multilayer clustering network using distributed energy efficient clustering with enhanced threshold protocol," *Wireless Personal Communications*, 2021.

[20] N. R. Nayak, S. Kumar, D. Gupta, A. Suri, M. Naved, and M. Soni, "Network mining techniques to analyze the risk of the occupational accident via bayesian network," *International Journal of System Assurance Engineering and Management*, vol. 1, no. 1, pp. 01–09, 2022.

[21] V. De Florio, G. Deconinck, and R. Lauwereins, "An algorithm for tolerating crash failures in distributed systems," in *Proceedings of the Seventh IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2000)*, pp. 9–17, IEEE, Edinburgh, UK, April 2000.

[22] S. Kakei, Y. Shiraishi, M. Mohri, T. Nakamura, M. Hashimoto, and S. Saito, "Cross-certification towards distributed authentication infrastructure: a case of hyperledger fabric," *IEEE Access*, vol. 8, pp. 135742–135757, 2020.

[23] S. Kumar, P. K. Baag, and K. V. Shaji, "Impact of ESG integration on equity performance between developed and developing economy: Evidence from S and P 500 and NIFTY 50," vol. 20, no. 4, pp. 01–16, 2021.

[24] M. Rakhra, R. Singh, T. K. Lohani, and M. Shabaz, "Metaheuristic and machine learning-based smart engine for renting and sharing of agriculture equipment," in *Mathematical Problems in Engineering*, D. Singh, Ed., vol. 2021, , pp. 1–13, Hindawi Limited, 2021.

[25] G. Fimiani, "Supporting privacy in a cloud-based health information system by means of fuzzy conditional identity-based proxy Re-encryption (FCI-pre)," in *Proceedings of the 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 569–572, IEEE, Krakow, Poland, May 2018.

[26] H. Deng, L. Huang, H. Xu, X. Liu, P. Wang, and X. Fang, "Revenue maximization for dynamic expansion of geo-distributed cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 83, pp. 899–913, July-September 2018.

[27] S. Porwal and S. Mittal, "Design of concurrent ciphertext policy-attribute based encryption library for multilevel access of encrypted data," in *Proceedings of the 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 42–47, IEEE, Solan, India, December 2018.

[28] M. N. Ghuge and P. N. Chatur, "Collaborative key management in ciphertext policy attribute based encryption for cloud," in *Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 156–158, IEEE, Coimbatore, India, April 2018.