Received April 19, 2020, accepted April 29, 2020, date of publication May 6, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992649

BDSS-FA: A Blockchain-Based Data Security Sharing Platform With Fine-Grained Access Control

HONG XU[1,2], QIAN HE[1,2], XUECONG LI[1,3], BINGCHENG JIANG[2,3], AND KUANGYU QIN[1,2]

[1]State and Local Joint Engineering Research Center for Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China
[2]Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[3]CETC Key Laboratory of Aerospace Information Applications, Shijiazhuang 050081, China

Corresponding author: Qian He (heqian@guet.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61661015 and Grant 61967005, in part by the Guangxi Innovation-Driven Development Project under Grant AA17202024, in part by the Guangxi Key Laboratory of Cryptography and Information Security Found under Grant GCIS201701, in part by the Guangxi Collaborative Innovation Center of Cloud Computing and Big Data Found under Grant YD1901, in part by the Innovation Project of GUET Graduate Education under Grant 2019YCXS046, in part by the CETC Key Laboratory of Aerospace Information Applications Found, in part by the Young and Middle-Aged Backbone Teacher of Guangxi Colleges and Universities Found, and in part by the Innovation Research Team of Guangxi Natural Science Foundation under Grant 2019GXNSFGA245004.

ABSTRACT Aiming at the problem of privacy leakage during data sharing in the Internet of Things, a blockchain based secure data sharing platform with fine-grained access control(BSDS-FA) is proposed. First, this paper proposes a new hierarchical attribute-based encryption algorithm, which uses hierarchical attribute structure and multi-level authorization center. The algorithm implements flexible and fine-grained access control by distributing different user attributes to different authorization centers. Then, it combined with the Fabric blockchain technology to solve the problem of huge decryption cost for users in the Internet of things. Smart contract in blockchain executes high-complexity partial decryption algorithm to reduce the users' decryption overhead. Blockchain can also realize the traceability of historical operations to meet the security requirements of data restriction open and transparent supervision. Finally, the hierarchical attribute-based encryption algorithm is proved to be CPA-safe. The theoretical analysis and experimental results show that BDSS-FA provides more secure and reliable data sharing services for users in the Internet of Things.

INDEX TERMS Attribute-based encryption, access control, blockchain, smart contract, Internet of Things (IoT).

I. INTRODUCTION

With the rapid development of Internet and sensor technology, more and more physical entities are connected to the Internet through sensors to realize information sharing, and the Internet of things(IoT) is born in this background [1], [2]. IoT can not only realize the connection between things and things, people and things, people and people, but also has been widely used in industry, agriculture, smart city, health care and other fields, which playing an important role in the development of national economy and human society [3], [4]. In the face of increasing data volumes and increasingly complex network topologies, how to establish an effective data sharing mechanism between different organizations for IoT has become a huge challenge [5].

Traditional data sharing mechanism generally upload IoT device data to the third-party service agency through sensors, and then the third-party service agency mines and analyzes these data through machine learning and statistical analysis to provide users with more convenient services. However, while people enjoy high-quality and personalized services, there is also the risk of personal privacy being revealed. Take wearable devices as examples, wearable devices include medical devices (blood glucose meters, sphygmomanometers, oximeter watches, etc.) and sports health devices (smart bracelets, etc.) [6]. Users constantly monitor their physical indicators through sensors, and send these sensing data to a third party based on their own health status, which providing

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Ning Dai.

users with more timely treatment. However, in the process of data transmission, user privacy is likely to be exposed, such as the user's location information and various physical function data. In order to eliminate third-party service organizations and quickly realize the secure sharing of data in peer-to-peer network, blockchain technology has attracted extensive attention, bringing opportunities to solve the challenges of IoT [7].

Blockchain is a peer-to-peer network with distributed connection, which can prevent the data collected by IoT from being transmitted through third-party service organizations, improve the transmission rate of data, and reduce transmission delay. In order to securely store and transmit this data to ensure its integrity, validity and authenticity, the access control has also become an important research content to ensure the secure sharing of data in the IoT. Therefore, many scholars have combined the blockchain technology with the existing access control model to carry out a series of research work. Zyskind and Nathan [8] combined the discretionary access control model (DAC) to manage sensitive data off-chain through the access control policy on the blockchain. Cruz *et al.* [9] used blockchain to solve cross-organizational access control issues in the role-based access control model (RBAC), which achieving cross-organizational authentication for user roles. Maesa *et al.* [10] extended the standard workflow of the attribute-based access control model (ABAC), replacing the traditional database with blockchain to store policies, and managing access policies in the form of transactions. However, the above method is only applicable to specific scenarios, and the access control is single, which is not suitable for one-to-many encryption scenarios in IoT.

At present, attribute-based encryption algorithms(ABE) [11] are considered as a solution to the problem of secure access control. ABE is developed on the basis of Identity-based Encryption(IBE), which is particularly suitable for one-to-many encryption scenarios. When the decryptor satisfies certain requirements specified by the encryptor, the decryptor can successfully decrypt the ciphertext that she/he wants to access. ABE algorithm not only guarantees data confidentiality but also provides fine-grained access control to data. Therefore, under the environment with high data sharing rate of IoT, ABE encryption algorithm can effectively solve the problem of secure access control for outsourced data.

Therefore, based on literature [12], this paper proposes a blockchain-based secure data sharing platform and fine-grained access control (BSDS-FA) combined with ABE. The main contributions of this paper are summarized as follows:

(1) Aiming at the problem of access control in data sharing, a new hierarchical attribute-based encryption algorithm (HABE) was proposed. By assigning different users to different authorization centers for management, the system performance of a single authorization agency is improved, and BSDS-FA can provide fine-grained access control while ensuring the security of shared data.

(2) Based on the Fabric blockchain, designing two smart contracts – Validation Contract and Decryption Contract. The Validation Contract is responsible for detecting the validity of user access right, while the Decryption Contract is responsible for performing partial decryption for the ciphertext of HABE, which reducing the computational overhead of data consumer and improving decryption performance.

(3) Based on the data sharing model, BSDS-FA is implemented for the actual data distribution system, and the security of HABE is proved. The experimental results show that BSDS-FA is practical and effective.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III introduces the system model. In section IV, the relevant algorithms of HABE and the main functional modules of BSDS-FA are introduced in detail. Section V describes the specific design of two smart contracts. Section VI proves the security of HABE algorithm. Section VII tests the performance of BSDS-FA, and the work is summarized in section VIII.

II. RELATED WORK

With the large-scale deployment of sensors, the amount of data in IoT is exploding, but these data will produce different values for different organizations. In order to make these data play a greater role, the shared data, data providers and data consumers are generally integrated into a platform to achieve valuable interconnection of data. However, many of the shared data in the data sharing platform contain the private information of data providers. If the necessary security measures are not provided for the shared data, it is easy to cause the privacy leakage of users and threaten the personal safety or property security of the data provider. Therefore, the security of data sharing in IoT has attracted widespread attention.

In the data sharing platform, Balamurugan *et al.* [13] transfers data by issuing tokens, and only data consumers with tokens can access the shared data in the data sharing platform. However, with the increase of data consumers, the load of data providers will also increase dramatically, resulting in instability or even collapse of data sharing platform. In view of this, Sun and Ji [14] introduced a third-party service organization in the data sharing platform, and the interaction between data providers and data consumers was conducted through third-party service organizations, which greatly improved the execution efficiency of the platform. However, the construction and maintenance cost of the data sharing platform is high, and it is vulnerable to malicious attacks by illegal users, resulting in data leakage or tampering. Therefore, in order to quickly realize the safe sharing of data in the data sharing platform, blockchain technology is introduced.

Blockchain is a distributed database system with multiple independent nodes, which can also be understood as distributed ledger [15]. It makes comprehensive use of cryptography, consensus mechanism, distributed network and other technologies to realize an interaction mode based on decentralized credit, which has practical significance for promoting the application of the IoT. In 2017, the ministry of industry and information technology released the "Blockchain-Reference Architecture" and "Blockchain-Data format specification" [16], [17], aiming to provide a standard for the practical application of blockchain and provide guidance to open the IoT distributed platform by using blockchain. Shafagh *et al.* [18] proposed a data-centric storage system for IoT based on blockchain, but the scheme is limited to the theoretical level and has not been verified by experiments. Ge *et al.* [19] proposed a lightweight information sharing security framework for IoT based on blockchain, and the security, feasibility and effectiveness of the framework were proved by simulation experiments. However, when the framework is applied to specific industries, there are shortcomings in both privacy protection capabilities and concurrency capabilities.

In addition, none of the above schemes consider the problem of user access control, which makes the shared data possible to be stolen by illegal users, resulting in the disclosure of privacy. With the rapid development of blockchain technology, platforms such as Ethereum and Hyperledger began to support various types of smart contracts [20]. Azaria *et al.* [21], Ekblaw *et al.* [22], and Dagher *et al.* [23] combines smart contracts with access control to achieve automatic permission management of medical data, and realizes the integration and permission management of distributed medical data of different organizations based on the Ethereum platform. However, the scheme adopts the PoW consensus mechanism, which makes the calculation overhead too large when maintaining the consistency of the blockchain. Therefore, Xue *et al.* [24] improved the consensus mechanism, using the DPoS consensus mechanism to reduce the computational burden of nodes, but it brought extraordinary data storage overhead, which was not practical. A secure FaBric blockchain-based data transmission technique for industrial IoT [25] was proposed to improve the security of data in the transmission process, which reducing the communication overhead.

Although the above schemes all provide access control for shared data, their access control is relatively single, which cannot provide good privacy protection for the IoT with complex attributes. Attribute-based encryption algorithm [11] can realize fine-grained access control based on user attributes, which is one of the key technologies to solve the current secure sharing of IoT data. Rahulamathavan *et al.* [26] introduced the attribute-based encryption algorithm into the IoT ecosystem based on blockchain, improving the security of shared data in the system. However, the system stores all the data directly in the blockchain, which greatly increases the storage burden of the blockchain. Wang *et al.* [27] designed a data security sharing model based on blockchain, which can provide fine-grained access control while data is securely shared. However, the decryption overhead of the data consumers in this model is huge, which is not suitable for the scalable data sharing in IoT.

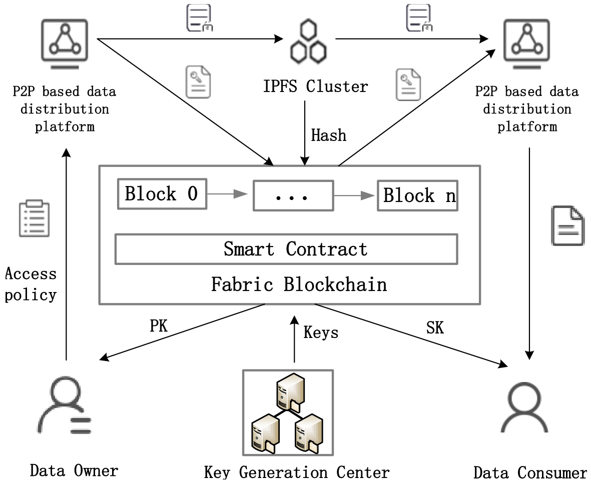

FIGURE 1. Model framework of BSDS-FA.

III. MODEL AND FUNCTION

A. SYSTEM INITIALIZATION

BSDS-FA is designed based on blockchain and HABE, which can provide a secure data sharing mechanism with traceability and fine-grained access control. The system framework is shown in Fig. 1, including Key Generation Center(KGC), Data Owner(DO), P2P based data distribution platform, IPFS Cluster, Fabric Blockchain and Data Consumer(DC).

The functions of the six entities are as follows.

1) KEY GENERATION CENTER

In BSDS-FA, KGC consists of a central authorization center(AC) and multiple domain authorization centers(DAC). For AC, it is responsible for the generation and distribution of system parameters, and management of the first-level DAC. For DAC, it is mainly responsible for generating corresponding keys for subordinate domain authorization centers or data consumers. Note that the function of each DAC is roughly the same. The only difference is that the attribute set managed by each DAC is different, and the lower-level DAC is managed by the higher-level DAC.

2) DATA OWNER

DO is a provider of shared data. DO collects raw data in IoT devices and shares remotely sensitive data with others through P2P-based data distribution platform. When DO wants to publish data, DO formulates HABE access control for the shared data, and only users who meet the access control have access to get the data in BSDS-FA.

3) P2P-BASED DATA DISTRIBUTION PLATFORM

When DO wants to share data, the platform will generate the corresponding seed information description file, and then perform two tasks: 1) The platform publishes the encrypted file sent by the data owner to the IPFS cluster; 2) The platform uses the DO's PK to encrypt the random key with

HABE algorithm and publishes the encrypted results to Fabric blockchain.

4) IPFS CLUSTER

It is a distributed file system that combines distributed hash tables, incentive block exchanges, and self-certified namespaces. It is mainly responsible for storing encrypted files uploaded by DO through P2P-based data distribution platform, and storing the hash value of the encrypted files on the block chain, which alleviating the storage pressure of blockchain.

5) FABRIC BLOCKCHAIN

Smart contracts are deployed on blockchain to realize the interaction of on-chain and off-chain data. When data consumer wants to access data in BSDS-FA, Validation Contract is mainly responsible for verifying whether the user has access to get the shared data, and Decryption Contract is mainly responsible for providing partial decryption to users with access right, which improving the decryption performance of data consumer.

6) DATA CONSUMER

It is a user who is interested in data in BSDS-FA and wants to access them. DC accesses shared data through the Internet, and only users with access right can decrypt the shared data to obtain the plain text.

B. SYSTEM FUNCTION

BSDS-FA mainly includes four functional modules: system initialization, user registration, data upload, data download. Now, the main functions of these four modules will be described as a whole.

1) SYSTEM INITIALIZATION

System initialization is mainly performed by AC of KGC. When BSDS-FA is started, AC initializes system parameters to ensure the normal operation of BSDS-FA.

2) USER REGISTRATION

User registration is mainly performed by AC and DAC of KGC. When a user wants to access the shared data in BSDS-FA, he/she will submit his/her own set of attributes to BSDS-FA, and BSDS-FA will verify the user's identity and complete the user registration.

3) DATA UPLOAD

Data upload is mainly performed by DO. The shared data of the data owner is stored in the IPFS cluster after symmetric encryption, and the encryption key is uploaded to the smart contract on the blockchain after being encrypted by HABE (HABE is an asymmetric encryption algorithm). Therefore, through the combination of symmetric encryption and asymmetric encryption, BSDS-FA can improve the encryption and decryption performance while ensuring the security of shared data.

4) DATA DOWNLOAD

Data download is mainly performed by DC. DC completes the process of data access through smart contracts, IPFS cluster and P2P-based distribution mechanisms, requiring that only users with access rights to access shared data in BSDS-FA.

IV. REALIZATION AND APPLICATION

In Section III, we briefly introduced the functions of the four functional modules. Next, we will give a detailed description of the specific execution process of this function module.

A. SYSTEM INITIALIZATION

System initialization is generally run when BSDS-FA platform is started, and the specific process is shown in Algorithm 1. The main flow of the algorithm is as follows: First, BSDS-FA platform determines the depth d of the user's key structure (Line 1 of Algorithm 1). Then, the *Setup* algorithm of HABE is called to get the system public key PK and the system main key MK, and the PK and MK are uploaded to the smart contract(Line 2-3 of Algorithm 1).

Algorithm 1 The process of system initialization

```
Input: $d$
Output: PK,MK
1. $d \leftarrow$ CA initializes the depth of user's key structure
2. $PK, MK \leftarrow \mathrm{HABE}_{Setup}(d)$
3. CAupload the$PK, MK$to the smart contract
4. return PK, MK
```

The *Setup* algorithm of HABE is as follows:

$Setup(d) \Rightarrow (PK, MK)$. The algorithm selects a cyclic group G with order p and generator g. At the same time, AC picks random parameters $\alpha, \beta_i \in Z_p, \forall i\{1, 2 \ldots d\}$, where d is the depth of user key structure. Then it outputs PK $= \{G, g, h_1 = g^{\beta_1}, y_1 = g^{1/\beta_1}, h_2 = g^{\beta_2}, y_2 = g^{1/\beta_2}, e(g, g)^{\alpha}\}$ and MK $= \{\beta_1, \beta_2, g^{\alpha}\}$. Here we assume that $d = 2$.

B. USER REGISTRATION

User registration generally creates new users for BSDS-FA platform and the specific process is shown in Algorithm 2. When a user wants to join BSDS-FA platform, he /she will send a registration request to BSDS-FA, and then submit his/her identity information according to the requirements of BSDS-FA (Line 1 of Algorithm 2). When BSDS-FA receives the user's registration request, it will verify that the user's identity is legal (Line 2 of Algorithm 2). If the user is a legal user, the user will be registered, otherwise BSDS-FA rejects the user's registration request (Lines 3-4 of Algorithm 2). When BSDS-FA performs a registration operation for the user, the AC of KGC first calls the *Keygen* algorithm of HABE to generate corresponding transformation key TK_i and private key SK_i for the top-level domain authorization center (Line 6-7 of Algorithm 2). Then, the top-level domain authorization center calls the *Delegate* algorithm of HABE to generate corresponding TK_{i+1} and SK_{i+1} for the lower-level domain authorization center, and uploads the TK_{i+1} and SK_{i+1} to the smart contract (Line 8-10 of Algorithm 2). Note that TK_i, SK_i and TK_{i+1}, SK_{i+1} are used to represent the upper and lower relationship of the authorization center, so the conversion key and private key corresponding to a certain user is still represented by TK and SK respectively.

Algorithm 2 The process of user registration

```
Input: PK, MK, $\Lambda$
Output: $TK_{i+1}, SK_{i+1}$
1. DC send registration request
2. KGC verifies the identity of user
3. if verification result != true then
4.   return NULL
5. Else
6.   AC of KGC executes:
7.     $TK_i, SK_i \leftarrow HABE_{KeyGen}(PK, MK, \Lambda)$
8.   Top-level domain authorization center(DAC)
     executes:
9.     $TK_{i+1}, SK_{i+1} \leftarrow HABE_{Delegate}(TK_i, SK_i, u, \tilde{\Lambda})$
10.  DAC uploads$TK_{i+1}, SK_{i+1}$to the smart contract
11.  return $TK_{i+1}, SK_{i+1}$
12. end if
```

The *KeyGen* algorithm and *Delegate* algorithm of HABE are as follows:

(1)$KeyGen(PK, MK, \Lambda) \Rightarrow (TK_i, SK_i)$. After DC submits his own attribute set $\Lambda = \{A_0, A_1, \cdots A_m\}$, where $A_i = \{a_{i,1}, a_{i,2}, \ldots a_{i,n_i}\}$ with $a_{i,j}$ being the j th attribute of A_i and n_i being the number of attributes in A_i, the algorithm will generate a random number $r, z \in Z_p$ for DAC, preventing illegal users from conspiring to obtain user privacy. For the same reason, random numbers $r_i \in Z_p$ is generated for each set A_i of Λ, and random numbers $r_{i,j} \in Z_p$ is generated for each element $a_{i,j}$ of A_i.Then it outputs $\mathrm{TK}_i = \{\Lambda, D = y_1^{(\alpha+r)/z}, D_{i,j} = g^{r_i/z} \cdot H(a_{i,j})^{r_{i,j}/z}, D'_{i,j} = g^{r_{i,j}/z}, for 0 \leq i \leq m, 1 \leq j \leq n_i, E_i = y_2^{(r+r_i)/z}$, for $1 \leq i \leq m\}$ and $\mathrm{SK}_i = z$.

The element E_i of TK_i is used to decrypt of transform node, which can implement cross-set query of attributes. When the transform node is converted, $r_{i'}$ can be converted to r_i through $E_i/E_{i'}$.

(2)$Delegate(TK_i, SK_i, u, \tilde{\Lambda}) \Rightarrow (TK_{i+1}, SK_{i+1})$: When a new subordinate DAC_{i+1} or DC wants to join the system, the superordinate DAC_i will verify its identity and generate a subset $\tilde{\Lambda}(\tilde{\Lambda} \subset \Lambda)$ of attributes for it. As in KeyGen algorithm, this algorithm selects a random number $\hat{r} \in Z_p$ for DAC or DC, random numbers $\hat{r}_i \in Z_p$ for each set A_i of $\tilde{\Lambda}$ and random numbers $\hat{r}_{i,j} \in Z_p$ for each element $a_{i,j}$ of A_i. Then it outputs TK_{i+1} *or* $\mathrm{TK}_{user} = \{\tilde{\Lambda}, \tilde{D} = D \cdot g^{\hat{r}/(z \cdot \beta_1)}, \tilde{D}_{i,j} = D_{i,j} \cdot g^{\hat{r}_i/z} \cdot H(a_{i,j})^{\hat{r}_{i,j}/z}, \tilde{D}'_{i,j} = D'_{i,j} \cdot g^{\hat{r}_{i,j}/z}$, for $a_{i,j} \in \tilde{\Lambda}, \tilde{E}_i = E_i \cdot g^{(\hat{r}+\hat{r}_i)/(z \cdot \beta_2)}$, for $A_i \in \tilde{\Lambda}\}$ and $\mathrm{SK}_{i+1} or \mathrm{SK}_{user} = z$. Note that TK_{i+1} and SK_{i+1} are generated for the new subordinate DAC_{i+1}, while TK_{user} and SK_{user} are generated for the new users DC.

C. DATA UPLOAD

The data upload process is that the data owner provides shared data to BSDS-FA platform. In order to ensure the security of the shared data, the data owner will formulate access control policies and perform encryption operations for the shared data. The specific process is shown in Algorithm 3. When the data owner wants to upload the shared data, a random key ψ is generated for the shared data (Line 1 of Algorithm 3), and then the shared data is symmetric encrypted using the random key to obtain EncryptedFile (Line 2 of Algorithm 3). In addition, the data owner formulates an access policy Γ (Line 3 of Algorithm 3) for the shared data, and obtains the PK from the smart contract, and then sends EncryptedFile, Γ, ψ, PK to P2P based data distribution platform. When P2P platform receives this data, it stores the EncryptedFile on the IPFS cluster (Line 4 of Algorithm 3). At the same time, the seed resource server also generates the corresponding P2P seed file for the random key ψ, and then P2P platform uses *encryption* algorithm of HABE to encrypt the seed file, and store the encrypted result *CT* in the smart contract (Line 5-6 of Algorithm 3).

Algorithm 3 The process of data upload

```
Input: $file$, $\psi$,PK, $\Gamma$
Output: CT
1. $\psi$ $\leftarrow$ DO generate random symmetric key
2. EncryptedFile $\leftarrow AES_{encrypt}(file, \psi)$
3. DO formulates access control $\Gamma$
4. $IPFS_{Addr}$ $\leftarrow$ send EncryptedFile to IPFS Cluster
5. $CT$ $\leftarrow HABE_{Encrypt}$(PK,$\psi$, $\Gamma$)
6. The encrypted result $CT$ stored in the smart contract
7. return $CT$
```

The *encryption* algorithm of HABE is as follows:

$Encrypt(PK, \psi, \Gamma) \Rightarrow (CT)$: DO establishes access control Γ for data visitors. Γ is a tree structure, and each node x in Γ corresponds to a polynomial q_x of order d_x,where $d_x = k_x - 1$ and k_x is the threshold of node x. Note that $d_x = 0$ when x is leaf node. Starting from the root node r in Γ, DO picks a random value $s \in Z_p$ and makes $q_x(0) = s$. Next, DO randomly picks some numbers representing other values of q_x to compute polynomial completely. For any other nonroot node x, $q_x(0) = q_{\mathrm{parent}(x)}(index(x))$, and then selects other values of the polynomial. In addition, q_x of leaf node is constant. Then it outputs CT $= \{\Gamma, C = \psi \cdot e(g, g)^{\alpha \cdot s}, \tilde{C} = h_1^s, \bar{C} = h_2^s, \forall y \in Y : C_y = g^{q_{y(0)}}, C'_y = H(\mathrm{attr}(y))^{q_y(0)}, \forall x \in X : \hat{C}_x = h_2^{q_x(0)}\}$, where Y represents the set of leaf nodes and X represents the set of transform nodes.

D. DATA DOWNLOAD

The data download process is that the data consumer wants to access the shared data in BSDS-FA platform and the specific process is shown in Algorithm 4. When a data consumer wants to access shared data, he/she will send a query request to BSDS-FA (Line 1 of Algorithm 4). Then Validation

Contract calls *Verify* algorithm to verify whether the user's attribute set meets the access control (Lines 2-3 of Algorithm 4). If the user does not have permission to access the data, BSDS-FA will reject the user's request (Lines 4-5 of Algorithm 4). Otherwise, Decryption Contract executes the *PartDec* algorithm to obtain the semi-decryption ciphertext CT' for the user, and sends the CT' to P2P based data distribution platform (Line 6-7 of Algorithm 4). P2P platform decrypts the CT' using the *Decrypt* algorithm of HABE to obtain a random key ψ, and send it to the data consumer (Lines 9-10 of Algorithm 4). Finally, the data consumer obtains the EncryptedFile on the IPFS cluster and uses the random key ψ to symmetric decrypt the EncryptedFile to obtain shared data (Lines 11-13 of Algorithm 4).

Algorithm 4 The process of data download

```
Input: $\Gamma$, $\Lambda_{User}$
Output: $file$
1.        DC send data access request to smart contract
2.        Smart Contract executes:
3.          $\Gamma(\Lambda) \leftarrow Verify(\Gamma, \Lambda_{User})$
4.          if $\Gamma(\Lambda) ==$ NULL then:
5.            return NULL
6.          else
7.            $CT' \leftarrow PartDec(\text{CT}, TK_{User})$
8.          end if
9.     P2P based data distribution platform executes:
10.         $\psi \leftarrow HABE_{Decrypt}(CT', SK_{User})$
11.    DC executes:
12.         $EncryptedFile \leftarrow$ get file on IPFS according to
              $IPFS_{Addr}$
13.         $file \leftarrow AES_{Decrpt}(EncryptedFile, \psi)$
14.    return $file$
```

The *Verify* algorithm and *PartDec* algorithm will be introduced in Section V. While the *Decrypt* algorithm of HABE is as follows:

$Decrypt(CT', SK)$: When CT' is not empty, it indicates that DC has the right to access the ciphertext, and then DC performs a decryption operation on the CT' to obtain the plaintext information $\psi = C/E^{\text{SK}} = \psi \cdot e(g, g)^{\alpha \cdot s}/e(g, g)^{(\alpha s/z) \cdot z}$.

V. SMART CONTRACT DESIGN

Smart contract allows both parties to conduct trusted transactions that are traceable and irreversible without the supervision of a third-party manager. This section will introduce *Verify* algorithm of Validation Contract and *PartDec* algorithm of Decrption Contract.

A. VALIDATION CONTRACT

Validation Contract mainly verifies the access right of DC. The verification algorithm is a recursive algorithm and is implemented with Go language. It will judge whether user's attributes of DC meet access control Γ of HABE and stores the subset labels with $\Gamma(\Lambda)$. The specific execution process is shown in Algorithm 5, where attr, children, satisfiable and label are member variables of access control Γ.

Algorithm 5 The process of verification algorithm

```
Input: $\Lambda$, $\Gamma$
Output: $\Gamma(\Lambda)$
    convert set $\Lambda$ to an array $\boldsymbol{user\_attr}$ of string type
    len = len($\boldsymbol{user\_attr}$)
    if $\Gamma$ is leaf node then:
      for $i \in [0, len)$ do:
        if $\Gamma.attr == \boldsymbol{user\_attr}\ [i]$ then:
          $\Gamma.satisfiable =$ true
          $\Gamma.label = \boldsymbol{user\_attr}\ [i]$.label
        end if
      end for
     else:
       for $j \in [0$, len($\Gamma.children$) ) do:
           $Verify(\Gamma.\ children[j], \Lambda)$
     end for
    var parentLabel string
     for $k \in [0$,len($\Gamma.children$) ) do:
       if $\Gamma.children[k].satisfiable =$ true:
         parentLabel $+ = \Gamma.label$
      end if
    end for
    $\Gamma.label =$ parentLabel;
  end if
  $\Gamma(\Lambda) = \Gamma.label$
  return $\Gamma(\Lambda)$
```

In Algorithm 5,if $\Gamma(\Lambda)$ is NULL, it indicates that the authentication of DC has failed, that is, DC does not have permission to access the shared data in BSDS-FA. Otherwise, DC can perform normal decryption on the shared data.

B. DECRYPTION CONTRACT

Decryption Contract mainly performs partial decryption for the data requested by DC. When the result $\Gamma(\Lambda)$ of *Verify* algorithm is empty, returns $\perp$; otherwise, *PartDec* algorithm randomly picks a i from $\Gamma(\Lambda)$, and then calls the function $PartDecNode(CT, TK, node, i)$ recursively from the root node r. The specific execution process of $PartDecNode(CT, TK, node, i)$ is shown in Algorithm 6.

In algorithm 6, F_{node} is obtained by Lagrange interpolation, where $k = \text{index(child)}$, $B'_{\text{child}} = \{\text{index(child)}:$

child $\in$ $\text{B}_{\text{node}}\}$.The specific calculation process is as follows:

$$\begin{aligned} F_{\text{node}} &= \prod_{\text{child} \in B_{\text{node}}} F_{\text{child}}^{\Delta k, B'_{\text{child}}(0)} \\ &= \prod_{\text{child} \in B_{\text{node}}} (e(g, g)^{\frac{r_i \cdot q_{\text{child}}(0)}{z}})^{\Delta k, B'_{\text{child}}(0)} \\ &= \prod_{\text{child} \in B_{\text{node}}} (e(g, g)^{\frac{r_i \cdot q_{\text{parent(child)}}(\text{index(child)})}{z}})^{\Delta k, B'_{\text{child}}(0)} \\ &= e(g, g)^{\frac{r_i \cdot q_{\text{node}}(0)}{z}} \end{aligned}$$

Algorithm 6 The process of *PartDecNode(CT,TK,node,i)*

```
Input: CT,TK,node,$i$
Output: $F$
    if node is leaf node then:
        if node.$attr$ == $a_{i,j}(a_{i,j} \in A_i, A_i \in \Lambda)$ then:
            $F = e(C_{\text{node}}, D_{i,j})/e(C'_{\text{node}}, D'_{i,j})$
              $= e(g,g)^{r_i \cdot q_{\text{node}}(0)/z}$
        else:
            return NULL
        end if
    else:
        for $j$ ∈[0, len(node.$children$) ) do:
        child = node. $children[j]$
        childLabel = child.$label$
        if $i \in$ childLabel then:
            $F_{child} = PartDecNode(CT, TK, child, i)$
        else :
            pick $i' \in S_{\text{node}}$
            if $i' \neq i$ and child is $translating\ node$ then:
              $F'_{child} = PartDecNode(CT, TK, child, i')$
            else:
              return NULL
             end if
            if $i = 0$ then:
              $F_{\text{child}} = e(E_{i'}, \hat{C}_{\text{child}})/F'_{\text{child}}$
              $= e(g,g)^{r \cdot q_{\text{child}}(0)/z}$
            else:
              $F_{\text{child}} = e(E_i/E_{i'}, \hat{C}_{\text{child}}) \cdot F'_{\text{child}} =$
              $e(g,g)^{r_i \cdot q_{\text{child}}(0)/z}$
            end if
          end if
    end for
    $F_{\text{node}} = \prod_{child \in B_{\text{node}}} F_{child}^{\Delta k, B'_{child}(0)}$
    if $i = 0$ then:
    $F_{\text{node}} = e(g,g)^{r \cdot q_{\text{node}}(0)/z}$
else:
    $F_{\text{node}} = e(g,g)^{r_i \cdot q_{\text{node}}(0)/z}$
$F = F_{\text{node}}$
return $F$
```

When algorithm 6 is finished, the result F corresponding to the root node r will be obtained, that is $F = F_r$. So when $i = 0, F = e(g,g)^{r \cdot s/z}$, otherwise $F = e(g,g)^{r_i \cdot s/z}$.Then calculate R, when $i = 0, R = F$, otherwise $R = e(E_i, \hat{C}_r)$/F=$e(g,g)^{r \cdot s/z}$.

Finally, calculate $E = e(\tilde{C}, D)/R = e(g,g)^{\alpha s/z}$. Then it outputs CT$' = (C, E)$.

VI. SCHEME ANALYSIS

Theorem 1: If adversary cannot break CP-ABE with a non-negligible advantage ε in any polynomial time, then there is no polynomial time to make that the adversary can solve the DBDH problem with a non-negligible advantage $\varepsilon/2$.

Definition 1: If adversary A wins the attack game with a negligible advantage in any polynomial time, then the scheme in this paper reaches CPA security.

Proof: First, simulator B is constructed to play the role of challenger in the attack game. Next, the bilinear group G and bilinear mapping $e : G \times G \to G_T$ are defined. Note that p is a prime and p is the order of G, and g is the generator of G.

Initialization: Given a DBDH problem for simulator B, then adversary A chooses an access policy Γ and sends it to simulator B.

System establishment: Simulator B generates random numbers $\alpha, \beta_i \in Z_p, \forall i\{1, 2 \ldots d\}$ according to the depth of key structure, then PK $= \{G, g, h_1 = g^{\beta_1}, y_1 = g^{1/\beta_1}, h_2 = g^{\beta_2}, y_2 = g^{1/\beta_2}, e(g,g)^{\alpha}\}$ and MK $= \{\beta_1, \beta_2, g^{\alpha}\}$, and sends PK to adversary A, while simulator B reserves MK.

Phase 1: Adversary A selects the attribute set $\Lambda = \{A_0, A_1, \cdots A_m\}$ and submits it to simulator B for private key SK and conversion key TK query, and all the attribute sets $A_i (0 \leq i \leq m)$ used by adversary A for key query do not satisfy the access policy Γ. Next, simulator B selects random values $r, r', z \in Z_p$ to calculate TK_{user}=$(D = g^{(\alpha+r)/\beta z}, \forall a_i \in S : D_i = g^{r/z} \cdot H(a_i)^{r_i/z}, D'_i = g^{r_i/z})$ and SK$_{user} = z$. Finally, simulator B sends SK to adversary A.

Challenge phase: Adversary A randomly selects two equal-length ciphertexts M_0, M_1 and sends them to the simulator. The simulator chooses one of them M_b, where $b \in \{0, 1\}$, calculates the ciphertext CT = $\{\Gamma, C = \psi \cdot$
$e(g,g)^{\alpha \cdot s}, \tilde{C} = h_1^s, \bar{C} = h_2^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(\text{attr}(y))^{q_y(0)}, \forall x \in X : \hat{C}_x = h_2^{q_x(0)}\}$ and sends it to opponent A.

Phase 2: Repeat **Phase 1**.

Guessing phase: Adversary A outputs guess $b' \in \{0, 1\}$. If $b' = b$, adversary A guesses $Z = e(g,g)^{abc}$, and the advantage of simulator to solve DBDH guess problem is $\Pr[b' = b | Z = e(g,g)^{abc}] = 1/2 + \varepsilon$. If $b' \neq b$, the z guessed by adversary A is only a random number on G, that is, the adversary A cannot obtain any information related to the plaintext M_b, while the advantage of simulator to solve DBDH guess problem is $\Pr[b' \neq b | Z = e(g,g)^z] = 1/2$.

From the above conjecture process, it can be seen that the advantage of simulator to solve DBDH conjecture problem is $\varepsilon/2$. Therefore, in this scheme, if the advantage ε of adversary A to win the game cannot be ignored, then the simulator can solve DBDH hypothesis problem with advantage $\varepsilon/2$, that is, simulator can break DBDH hypothesis problem. However, there is no non-negligible advantage to solve DBDH problem in polynomial time, so adversary A does not break the scheme proposed in this paper with non-negligible advantage, that is, the scheme proposed in this paper is CPA security.

VII. EXPERIMENTS ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

Based on Java and Go language, this paper implements BSDS-FA platform. In this experiment, the experimental

TABLE 1. The configuration of blockchain.

Server	Configuration	Function	Number
Zookeeper	4Core 8G 500G	Consensus mechanism	3
Kafka	4 Core 8G 500G	Consensus mechanism	4
Orderer	4 Core 8G 500G	Sorting service	3
Peer	4 Core 8G 500G	Maintain blockchain ledger	3
IPFS	4 Core 8G 500G	Storage	3

environment mainly includes P2P-based data distribution platform and Fabric blockchain.

P2P-based data distribution platform mainly consists of a notebook computer (CPU: Intel i7-4710MQ, OS: Windows 10, Memory: 16G) and four workstations (CPU: Intel Xeon E3 (4 core), Memory: 16G). The data management system is deployed on a workstation, and DO in BSDS-FA can upload shared data through the system. The laptop and three other workstations simulate DC in BSDS-FA.

The deployed Fabric blockchain consists of three organizations, each of which has one blockchain node (3 Peer nodes), three IPFS nodes, three Zookeeper service nodes, four Kafka service nodes and three ordering sort service nodes. These components are respectively deployed on one workstation (CPU: Intel Xeon i7-7700(4 core, 3.6GHz), Memory:8G, Hard Disk Memory:500G) according to the server type serial number. The configuration used is shown in Table 1.

In addition, the experiments in this paper were carried on the real data set Kosarak. The Kosarak data set is the number of clicks on the Hungarian news website provided by Ferenc Bodon. Each record represents the track of the news page accessed by user within a certain day. It contains 41270 news pages and 990002 records. However, the Kosarak data set is about 12.2MB, and the experiments in this paper were conducted with 10MB, so the records in the Kosarak data set have been partially deleted.

B. ENCRYPTION AND DECRYPTION PERFORMANCE

In HABE, the number of exponential operations and bilinear mapping operations performed by the user during decryption is often proportional to the complexity of the access control policy defined in the ciphertext. Therefore, when the access control policy is more complicated, the decryptor's computational overhead is greater. However, in the era of IoT, the overly complex operation of attribute-based decryption not only improves the performance requirements of users' decryption devices, but also makes users wait too long in the process of decryption, affecting user's experience. In this scheme, the design of the Decryption Contract enables the user to perform only a small number of operations, which reduces the user's computing overhead. At the same time, by using HABE to provide fine-grained access control, which improves the user experience.

The experiments test local encryption time, local decryption time and the decryption time of Decrypt Contract when the number of attributes contained in CT is 10-50. The results are shown in Table 2.

TABLE 2. The cost time of different attribute.

Number of attributes	Local Encryption Time (s)	Local Decryption Time(s)	Decryption Time of Decrypt Contract(s)
10	0.445	0.05	0.367
20	0.834	0.05	0.711
30	1.237	0.05	1.044
40	1.664	0.05	1.362
50	2.049	0.05	1.706

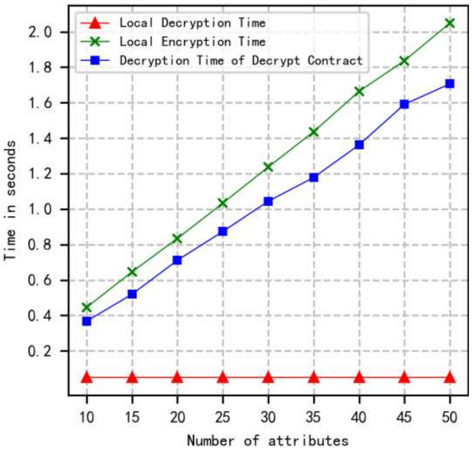

FIGURE 2. The cost time of different attribute.

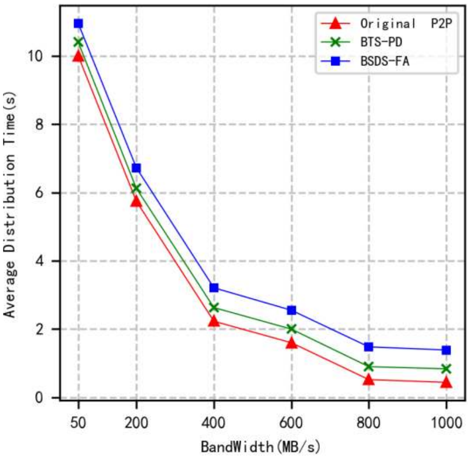

FIGURE 3. Distribution time of 10MB data in different bandwidth.

The comparison of cost time under the different attributes is shown in Fig.2.

The experimental comparison results show that by outsourcing a large amount of calculations in the decryption process to the Decryption Contract, the user's decryption calculation overhead is greatly reduced, the user's waiting time is shortened, and the user experience is improved.

C. CFILE DISTRIBUTION PERFORMANCE

File distribution performance is different for different systems. This paper compares the Original P2P distribution system, BTS-PD [12] and BSDS-FA. The experiment result is shown in Fig.3.

Fig.3 shows that the file distribution time of BSDS-FA is slightly higher than Original P2P and BTS-PD, but it is not much higher. Although the file distribution time of

Original P2P and BTS-PD is lower, but there is no access for shared data, resulting in loss of data security, while BSDS-FA can provide fine-grained access control while ensuring data security, so the time consumption is worth it. In addition, this experiment is also conducted with a large file of 1GB, and the result is similar to this experiment, which indicates that the time cost of encryption and decryption has little effect on the running process of the whole system. Therefore, BSDS-FA is more suitable for the practical application scenarios in IoT.

VIII. CONCLUSION

In order to protect the privacy of users in the data sharing process of IoT, this paper proposes a blockchain based secure data sharing platform with fine-grained access control(BSDS-FA). This paper first proposes a new hierarchical attribute-based encryption algorithm(HABE), which introduces multiple authorization centers and hierarchizes the authorization centers on the basis of the traditional attribute-based encryption algorithm. So that in the case of massive data and massive users, HABE can also provide users with fine-grained access control while ensuring the security of user data. Then, the HABE algorithm combined with smart contract technology is applied to BSDS-FA, so that BSDS-FA can not only prevent illegal users from accessing shared data, but also reduce the user's decryption overhead. Among them, Validation Contract to review the user permissions, so that only users whose attribute set meets the access control have the right to access shared data; Decryption Contract was used to perform partial decryption operation on ciphertext of HABE to improve user's decryption performance. Finally, the safety proof of BDSS-FA was carried out, and relevant experiments were carried out. Experimental results show that BDSS-FA can provide users with more secure and reliable data sharing services while providing fine-grained access control without affecting download performance.

REFERENCES

[1] A. N. Peng, W. Zhou, Y. Jia, and Y. Q. Zhang, "Review of security research on Internet of Things operating system," *J. Commun.*, vol. 39, no. 3, pp. 22–34, 2018.

[2] Y. Q. Gao, X. Y. Li, and B. X. Fang, "A survey of Internet of Things searching techniques," *J. Commun.*, vol. 36, no. 12, pp. 57–76, 2015.

[3] J. R. Li, X. Y. Li, L. L. Gao, and B. X. Fang, "Research on data forwarding model in the Internet of Things environment," *J. Softw.*, vol. 29, no. 1, pp. 196–224, 2018.

[4] H. M. Chen, H. L. Shi, Y. Li, and L. Cui, "Middleware for Internet of Things services: Challenges and research progress," *Chin. J. Comput.*, vol. 40, no. 08, pp. 1725–1749, 2017.

[5] J. G. Yu, H. Zhang, Y. Li, L. S. Mao, and P. X. Ji, "Blockchain-based IoT data sharing model," *Small Micro Comput. Syst.*, vol. 40, no. 11, pp. 2324–2329, 2019.

[6] F. F. Mag, S. B. Liu, X. X. Xiong, and G. X. Niu, "Local differential privacy protection of wearable devices' numerical sensitive data," *J. Comput. Appl.*, vol. 39, no. 7, pp. 1985–1990, 2019.

[7] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[8] G. Zyskind and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, Oct. 2015, pp. 180–184.

[9] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[10] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, in Lecture Notes in Computer Science, vol. 10320. Cham, Switzerland: Springer, 2017, pp. 206–220.

[11] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[12] X. C. Li, Q. He, B. C. Jiang, X. Qin, and K. Y. Qin, "BTS-PD: A blockchain based traceability system for P2P distribution," in *Blockchain and Trustworthy Systems* (Communications in Computer and Information Science), vol. 1156. Singapore: Springer, 2019, pp. 607–620.

[13] B. Balamurugan, P. V. Krishna, M. Ninnala Devi, R. Meenakshi, and V. Ahinaya, "Enhanced framework for verifying user authorization and data correctness using token management system in the cloud," in *Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT)*, Mar. 2014, pp. 1443–1447.

[14] A. B. Sun and T. K. Ji, "Big data open sharing platform and industrial ecological construction for smart cities," *Big Data*, vol. 2, no. 4, pp. 69–82, 2016.

[15] W. D. Cai, L. Yu, R. Wang, N. Liu, and E. Y. Deng, "Research on the development method of application system based on blockchain," *J. Softw.*, vol. 28, no. 6, pp. 1474–1487, 2017.

[16] China Electronics Standardization Institute. *Blockchain-Reference Architecture [EB/OL]*. Accessed: Dec. 27, 2017. [Online]. Available: http://www.cesi.ac.cn/201705/2478.html.

[17] China Electronics Standardization Institute. *Blockchain-Data format specification [EB/OL]*. Accessed: Dec. 27, 2017. [Online]. Available: http://www.cesi.ac.cn/images/editor/

[18] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. Cloud Comput. Secur. Workshop (CCSW)*, Dallas, TX, USA, Nov. 2017, pp. 45–50.

[19] L. Ge, X. S. Ji, T. Jiang, and Y. M. Jiang, "Security mechanism of IoT information sharing based on blockchain technology," *J. Comput. Appl.*, vol. 39, no. 02, pp. 458–463, 2019.

[20] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020.

[21] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, Aug. 2016, pp. 25–30.

[22] A. Ekblaw, A. A. Azaria, J. D. Halamka, and A. Lippman, "A case study for blockchain in healthcare: 'MedRec' prototype for electronic health records and medical research data," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 5–56-ONC, 2016.

[23] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018.

[24] T. F. Xue, Q. C. Fu, Z. Wang, and X. Y. Wang, "A medical data sharing model via blockchain," *J. Automat.*, vol. 43, no. 9, pp. 1555–1562, 2017.

[25] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K.-C. Li, "A secure FaBric blockchain-based data transmission technique for industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3582–3592, Jun. 2019.

[26] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017, pp. 1–6.

[27] X. L. Wang, X. Z. Jiang, and Y. Li, "Model for data access control and Sharing based on blockchain," *J. Softw.*, vol. 6, pp. 1661–1669, Oct. 2019.

HONG XU was born in Meishan, Sichuan, China, in 1995. She received the B.S. degree from Leshan Normal University, in 2017. She is currently pursuing the M.S. degree with the Guilin University of Electronic Technology. Her research interests include cloud computing and information security.

QIAN HE was born in Hunan, China, in 1979. He received the bachelor's degree in engineering from Hunan University, in 2001, the master's degree in engineering from the Guilin University of Electronic Technology, in 2004, and the Ph.D. degree in engineering from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, in January 2011. After graduating, he stayed in the school to work at the Network Center. In 2004, he was selected to be sent to Tsinghua University to participate in the training of key software teachers in western universities of the Ministry of Education. He is currently a Postdoctoral Fellow with the School of Computer Science, National University of Defense Technology, and a Visiting Scholar with The University of Manchester. He is also a Full Professor with the Guilin University of Electronic Technology. He chairs the National Natural Science Foundation of China, Research on Web Services Organization and Automatic Construction Methods Based on Active Peer-to-Peer Architecture. In addition, as a main member, he participated in one National Natural Science Foundation and two national defense pre-research projects. He has published more than 20 articles and SCI/EI included more than 10 articles. He is a Senior Member of CCF. He won the Second Prize of Guangxi Science and Technology Progress Award.

XUECONG LI was born in Luoyang, Henan, China, in 1995. She received the B.S. degree from Henan Agricultural University. She is currently pursuing the M.S. degree with the Guilin University of Electronic Technology. Her research interests include blockchain and information security.

BINGCHENG JIANG was born in Shanwei, Guangdong, China, in 1990. He received the B.S. degree from the Dongguan University of Technology and the M.S. degree from the Guilin University of Electronic Technology, where he is currently pursuing the Ph.D. degree. His research interests include information security and service computing.

KUANGYU QIN received the B.E. degree in mechanical engineering from the University of Science and Technology Beijing, China, in 1995, the M.C.A. degree in computer applications from Bangalore University, India, in 2008, and the Ph.D. degree from Wuhan University. He is currently a Senior Engineer with the Guilin University of Electronic Technology. His research interests include software defined networking, network management, and network security.