

Received June 12, 2021, accepted July 13, 2021, date of publication July 27, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100707

A Reliability Guaranteed Solution for Data Storing and Sharing

TAM T. HUYNH¹, THUC D. NGUYEN², THANG HOANG³, LAM TRAN⁴, AND DEOKJAI CHOI⁴

¹Posts and Telecommunications Institute, Ho Chi Minh City 70000, Vietnam

²Department of Computer Science, Vietnam National University Ho Chi Minh City-University of Science, Ho Chi Minh City 70000, Vietnam

³Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33612, USA

⁴Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, South Korea

Corresponding author: Tam T. Huynh (tamht@ptithcm.edu.vn)

This work was supported by Vietnam National University Ho Chi Minh City (VNU-HCM) under Grant NCM2019-18-01.

ABSTRACT Digital data certified by a reputable organization are valuable digital data that can be stored or shared on the internet. However, the problems are: (1) How to ensure the anonymity of organizations on issued certificates? (2) How to ensure that valuable digital data are securely stored in the system? and (3) How could people verify the reliability of shared data while still ensuring the confidentiality of its content, and how to ensure that the data sharing process is safe, transparent, and fair? Therefore, we propose data producing, data storing, and data sharing schemas. In the data producing schema, we deploy a group signature scheme for a group of reputable organizations that provide the same type of service, an organization in the group generates a valuable digital data from raw data sent from a data owner and then issues a certificate on the ciphertext of this digital data. In the data storing schema, the data owner uploads his/her data to the public Inter-Planetary File System network and then stores the access address of the stored data and the corresponding certificate on the blockchain ledger. In the data sharing schema, everyone on the system could verify the reliability of shared data before sending a data sharing request to the data owner. The data sharing process is performed via a smart contract, and involved parties have to escrow to encourage honesty. The schemas of data storing and sharing guarantee the security properties including confidentiality, integrity, privacy, non-repudiation, and anonymity.

INDEX TERMS Blockchain, IPFS, data storing, data sharing.

I. INTRODUCTION

There has been exponential data growth in the world, and trusted data are considered one of the most valuable assets of individuals and organizations. The amount of data created and stored globally are predicted to create about 175 zettabytes by 2025. It is also estimated that by 2025 the global consumers interacting with data everyday will reach 5 billion [1]. Consequently, the demand for valuable data storing and sharing is tremendous, which also poses challenges related to data security in the processes of data storing and sharing. Currently, there are two main architectures used for data storing and sharing, centralized and decentralized architectures.

For the centralized architecture, organizations can store data on their datacenter system. However, these systems have high operating costs and are limited in scalability [2]. Using cloud storage services can reduce costs and can be

flexible in system expansion, and more suitable for IoT systems. The combination of IoT and cloud storage services is a matter of studies in [3]–[7]. To protect the security and privacy of data storing and sharing, encryption algorithms and access control models are proposed in [8]–[11], Murat Kantarcioglu *et al.* [12] proposed SECURED L for protecting the sensitive data stored in databases. However, the centralized architecture has two limitations including [13]: (1) *data security*, stored data could be accessed, modified, or removed illegally by system administrators or attackers who compromised the system; (2) *availability*, when the centralized systems are crashed due to system overload, denial-of-service or distributed denial-of-service (DoS/DDoS) attacks, or system errors, the services are not available for users.

For the decentralized architecture, most solutions use blockchain (BC) technology as the main component in the systems because of its properties such as *anonymity*, *transparency*, *decentralization*, and *auditability* [14]–[16]. However, current solutions do not provide features for verifying

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Verticale¹.

the accuracy and the reliability of the shared data on the BC network. Specifically, data verified and certified by a reputable organization (RO) are considered as meaningful data (MD). For instance, in the medical field, a diagnostic result of an electronic medical record is published by a reputable medical organization with highly skilled doctors, which is MD. In the education field, a lecture that is assessed and certified by a professional board of a reputable university is MD. MD needs to be securely stored on the system, besides a data owner (DO) can completely share or commercialize his/her MD to other people or organizations on the network. Data sharing methods must ensure that requesters can verify the reliability and accuracy of shared data before deciding to perform a data-sharing contract.

With the traditional data sharing method, the integrity of shared data is based on trust between the two partners participating in the exchange process. For example, doctors/hospitals absolutely believe that medical records received from their patients are integrity. In some cases, RO needs to ensure anonymity in MD generated by themselves. And the privacy of DO also needs to be protected as they don't want anyone to know which RO's service they used. In addition, the identities of those involved in the sharing process also need to be anonymous; and shared data need to be verified the reliability while still ensuring the privacy of its content.

Data storing and sharing for certified digital data are very necessary, which requires data storage and sharing solutions that need to meet all of the following requirements:

- For data storing: The anonymity of certificate authorities and the privacy of DO on stored data must be protected; Stored data in the system must be guaranteed confidentiality and integrity.
- For data sharing: Everyone on the system can verify the reliability of shared data before submitting a sharing request to DO. Note that everyone can only verify the reliability of the shared data but cannot read its contents; The data sharing process is done directly between DO and DU without depending on any intermediaries.
- The system serving data storage and sharing must ensure availability, integrity, and scalability.

However, current solutions do not meet all of the above requirements. In this paper, we propose data producing, data storing, and data sharing schemes. We consider RO as a data provider (DP), and DPs providing the same type of service join in a group. In the data producing scheme, a group manager sets up a group of DPs that provide the same type of service. A raw data of DO is produced into MD by a particular DP in the group. Then, DP encrypts MD using a symmetric algorithm along with a secret key. Later, DP generates a certificate on the MD ciphertext (denoted by EMD). Finally, EMD, the certificate, and DP's information will be sent to DO through a secure channel. In the data storing scheme, DO stores EMD on Inter-Planetary File System (IPFS), the access address of EMD on IPFS and related information are stored in a transaction on the blockchain system. In the

data sharing scheme, the group manager deploys a smart contract for data sharing, which ensures that the data purchasing/selling process is safe, transparent, and fair between participants. Data User (DU) may lookup and verify the accuracy and reliability of shared data on the system before performing the data-sharing smart contract. Our contribution in this paper can be summarized as follows:

- We propose a data producing scheme, which guarantees MD received from DP is completely accurate and reliability. This scheme could protect the anonymity of DP and could guarantee the privacy of DO.
- We propose a data storing scheme, in which we use BC technology and IPFS to form a secure storage system.
- We design a data sharing scheme, in which we propose the Purchase and Resolve algorithms, and also build the rules for the data sharing scheme to prevent fraud.
- We evaluate the proposed system and schemes in terms of the advantages, security features, and performance features.

II. RELATED WORK

The general characteristic of the current BC-based data storing and sharing solutions is that using BC technology to perform transactions and to store small data such as management information, access control policies, access addresses of shared data Meanwhile shared data are stored on centralized or decentralized systems.

For centralized storage solutions presented in [17]–[21], data are encrypted by a cryptographic algorithm before sending them to the data storage service provider. In this way, the privacy of data can be preserved, however, the availability of data depends on the service provider. To overcome the disadvantages of the centralized storage model, a decentralized storage platform named IPFS is proposed in [22]–[24].

As far as data sharing is concerned, Qi Xia *et al.* [17] proposed a BC-based data sharing framework for electronic medical records (EMRs). In this solution, only users who are authorized to join the system can use their membership private key allocated by the verifier to create data retrieval requests. The consensus nodes will receive these requests, perform the database queries, and return the results to users. Similarly, Zheng Xiaochen *et al.* [18] presented a personal health data sharing system based on BC, cloud storage and machine learning techniques. Where personal health information is compressed and encrypted before storing them in a cloud storage, and customers can search interested data to buy. Secret keys are transferred from users to key keepers, or from the key keepers to customers via an authenticated communication channel. However, the above-mentioned solutions are still manual because the sharing processes depend on an intermediary such as key keepers, verifiers and consensus nodes.

In another work, to protect the privacy in EMRs sharing, researchers proposed a BC-based privacy-preserving data sharing solution [19], in which access control policies to medical data are pre-set by owners through smart contracts.

However, this solution does not provide a method for people on the BC network to verify the accuracy and reliability of medical data before requesting patients' EMRs. In [20], data sharing policies from users to healthcare providers and to insurance companies are implemented by a hyperledger fabric membership service component and a channel scheme. But, data do not guarantee reliability because they have not been verified by any reputable health organizations. Additionally, the authors in [21] proposed a solution of sharing patient data among hospitals called MedBlock. It aims to allow patients to easily access their EMRs from different hospitals. However, the proposed solution only provides methods for storing data synchronously and controlling access from users in different hospitals.

For the peer-to-peer data sharing solutions mentioned in [22]–[24], data also encrypted before uploading them to IPFS, while smart contracts are deployed to distribute access information to participants. Nonetheless, a buyer can not verify the reliability of data before submitting a data access request or making an escrow payment to a smart contract.

Recently, researchers [25] presented a BC-based secure and privacy-preserving data sharing mechanism for smart cities. The framework called PrivySharing focuses on ensuring confidentiality for shared data of users and also aims to control the permissions of stakeholders through access control list rules embedded in smart contracts. Likewise, another team of researchers proposed a BC-based mechanism to protect privacy data sharing for decentralized storage systems [26]. The researchers recommend that a ring signature scheme can be used to hide users' identities. To access a file on IPFS, a user needs to satisfy two conditions. Firstly, the user has to own a private hidden key corresponding to a hidden public key included in the associated hidden access control list of the file. Secondly, the user has a valid private key to decrypt the file. However, these solutions also do not support a mechanism for verifying the reliability of shared data.

III. PRELIMINARIES

A. BLOCKCHAIN

BC technology is a decentralized ledger recording all confirmed transactions and is also considered as a linked list of blocks, in which each block is pointed to its previous block via a hash pointer containing a hash value of the predecessor at a fixed time. The first block of the chain is called the genesis block which has no parent block, so the previous hash value field of the block is initiated by the BC network builder. The general structure of each block consists of a block header and a block body, where a block header contains management information of block as well as chain such as *block id*, *version*, *previous hash*, *timestamp*, while a block body stores a list of transactions [14].

There are two types of nodes on a BC network. User nodes which only perform transactions and do not hold the ledger. And miner nodes which are responsible for verifying

transactions, creating new blocks, and holding a ledger. The ledger contains blocks that reached the consensus from honest miners, and data in the ledger is immutable. A miner node can also perform transactions as a user node. Each node owns a public/private key pair where the private key is used to sign transactions, while the public key is as the BC address of the node. In general, there are three types of BC networks: Public BC, private BC, and consortium BC [27]. Nodes in a BC network communicate directly with each other through a peer-to-peer network. Therefore, to synchronize data on the ledger of miners, one of the consensus protocols can be deployed in a BC system. Some consensus protocols are as follows:

- *Proof-of-Work (PoW)* [28]: Each miner has to find a *nonce* value such that the hash value of the combination of the new block and the *nonce* must be equal to or smaller than a target hash value $H(\text{nonce} \parallel \text{prev_hash} \parallel \text{tx} \parallel \dots \parallel \text{tx}) < \text{target}$ [29].
- *Proof-of-Stake (PoS)* [15], [30]: At each mining round, a miner owning a certain amount of the network's total value has a high probability to propose a new block. Depending on the particular applications the stake value will be indicated.
- *Proof-of-Authentication (PoAh)* [31]: The basic idea of PoAh is that a normal node records transactions into a new block. Then, the node signs on the block before transmitting it to trusted nodes for verifying. After successful verification, a trusted node broadcasts the verified block together with its PoAh identification to the network. Other nodes verify the PoAh information to add this block into their local chain.
- *Proof-of-Activity (PoA)* [32]: PoA is a hybrid consensus protocol between PoS and PoW, where each miner tries to generate an empty block header that satisfies PoS requirement, and then switching to PoS, this block needs to be signed by a certain amount of stakeholders to be a valid block.

Ethereum BC, proposed by Vitalik Buterin [33], is a decentralized turing-complete platform called Ethereum Virtual Machine (EVM). The programs that run on EVM are called smart contracts. A contract is a set of functions defined by a sequence of bytecode instructions and is executed automatically when specific conditions are met [34]. Solidity is the most popular programming language for writing smart contracts.

B. IPFS

IPFS, proposed by Juan Benet [35], is a peer-to-peer distributed file system. Each IPFS node is initialized with a key pair (private key and corresponding public key) and is identified by NodeID generated by its public key. There are three types of nodes on an IPFS network described as follows [36]:

- *Client node*: This type of nodes uses the network to store or distribute data.

- *Retrieval miner node*: A retrieval miner node is responsible for distributing objects to other nodes on the network. However, objects are temporarily cached on its local storage and are removed periodically by the garbage collection process.
- *Storage miner node*: This type of nodes provides a large storage space and high-speed processing capacity to the network. The cluster and pinning services can be used in these nodes for replicating data on cluster nodes and keep objects available to the network.

Each file on IPFS is identified by the hash value of its content, this hash value is also the access address of the file on IPFS. When uploading a file to IPFS, it will be put into objects. Each IPFS object includes two fields, the data field which stores binary data, and the links field which contains an array of links that point to other related objects. Each link composed of three components: *name*, *hash*, and *size*. The first is as an alias of the link, the second is the hash value of the object pointed, and the last is the size of the pointed object. Each object can store up to 256 kilobytes (KB) of data, hence, if the size of a file is less than 256 KB, which is stored in one object with the empty link field. Otherwise, the file is split into chunks of 256 KB and using the Merkle DAG (Merkle directed acyclic graph) data structure for managing these chunks [37].

C. GROUP SIGNATURE

Group signatures, introduced by Chaum and van Heyst [38], allow a group member to anonymously sign on a message on behalf of the group. Verifiers only know whether the signature is valid, but they don't know exactly which group member signed. The components of a group signature scheme include: Group members, a group manager, and a revocation manager. The group manager is responsible for setting up and managing the group, while the revocation manager is responsible to revoke the anonymity of the group members who signed a signature. A group member, after registered to join the group and approved by the group manager, can sign on digital data on behalf of the group.

Definition 1 (Group Signature): A group signature scheme $GS = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Tracing}, \text{Vertracing})$ consists of the following five algorithms [39]:

- $(gpk, gmk, grk, gsk) \leftarrow \text{KeyGen}(1^\lambda, 1^n)$: On input 1^λ and 1^n , output the group's public key gpk , the membership manager's secret key gmk , the revocation manager's secret key grk , and gsk is an n -element vector of keys with $gsk[i]$ being the secret signing key of element i , $1 \leq i \leq n$.
- $\sigma \leftarrow \text{Sign}(gpk, gsk[i], M)$: Given the group's public key gpk , a secret key $gsk[i]$ of the i -th group member, and a message M , issue a signature σ .
- $0, 1 \leftarrow \text{Verify}(gpk, M, \sigma)$: Given the group's public key gpk , a message M , and a signature σ , output 1 if the signature is valid, otherwise 0.
- $(i, arg) \leftarrow \text{Tracing}(\sigma, M, grk, gpk)$: Given a signature σ , a message M , the revocation manager's secret key

TABLE 1. Notations.

Notation	Description
λ	Security parameter
n	The number of group members
$[n]$	The set of $1, \dots, n$
H	The cryptographic hash function
$\parallel$	String concatenation
$make_proc(x)$	The function produces a raw data x to digital data
$:\equiv$	The procedure can be performed by human intervention
$Rand_Key(\cdot)$	The random key generation algorithm
$E_k(x)$	Symmetric encryption of x using key k
$D_k(x)$	Symmetric decryption of x using key k
$PCS(x, k)$	The public-key cryptosystem with a message x and a key k

grk , and the group's public key gpk , outputs an identity $i \in [n]$ and an argument arg .

- $0, 1 \leftarrow \text{Vertracing}(\sigma, M, gpk, i, arg)$: Given a signature σ , a message M , a group public key gpk , an identity i of a group member, and an argument arg , output 1 if and only if arg was generated by tracing or 0 indicating failure.

A group signature scheme must satisfy the security requirements [39]: *Unforgeability*, *anonymity*, *unlinkability*, *no framing*, and *unforgeability of tracing verification*.

IV. THE PROPOSED SCHEMES

In this section, we will propose the system model, threat model, security features, system setup, and proposed schemes. The notations used in this paper are given in Table 1.

A. SYSTEM MODEL

Fig. 1 presents our system model. There are five parties in our schemes.

(i) Data owner: DO is a person who owns raw data (RD), DO provides RD to a particular DP for generating MD. DO has the right to store and share MD, unaltered or modified, for those in need.

(ii) A group of DPs: Which is created by the group manager, each DP is an organization that has the function and means for generating MD from the DO's RD. DP is not the owner of MD, therefore has no right to provide or use MD without the consent of DO. DPs in the group provide the same type of service.

(iii) Data user: DU is a person or organization who would like to use MD created by DP.

(iv) Decentralized storage (DS): DS mainly stores EMD and returns the address of EMD to DU. We use a public IPFS as DS.

(v) Blockchain system: We use BC to reserve information of MD and achieve data sharing. The group manager predefined policies in the smart contracts to ensure data sharing securely.

Our system offers Data Producing, Data Storing, and Data Sharing defined as follows:

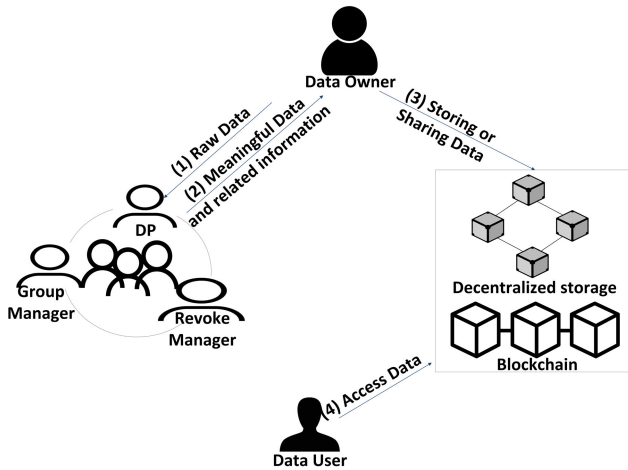


FIGURE 1. Our general system model.

- **Data Producing:** Which is a manual procedure, where RD from DO as input, it outputs MD and some related information (generated by DP) to DO.
- **Data Storing:** Given EMD and the related information from DO as input, it stores EMD on IPFS and initializes a store blockchain transaction containing the access address of EMD (on IPFS) and the related information.
- **Data Sharing:** Given a store transaction on BC as input, it verifies the reliability of MD and executes the smart contract to let DU obtain MD.

B. THREAT MODEL

We consider the following threat model in each of our schemes.

- **Data Producing:** There are DO, DP involved. We assume both DO and DP are trusted.
- **Data Storing:** There are DO, IPFS and BC systems involved. We assume DO is trusted, IPFS nodes and BC nodes honestly perform the pre-defined protocol but these nodes may access contents of data stored on the systems. Such nodes' compromise the confidentiality of stored data.
- **Data Sharing:** There are DO, DU, IPFS and BC systems involved. We assume DO and DU are untrusted, IPFS and BC systems are similar to Data Storing scheme. Specifically, DO may provide the invalid decryption key of EMD to DU, and DU may submit a dispute resolution request while it has received the valid decryption key of EMD.

C. SECURITY FEATURE

Our system provides security features as follows:

- **Confidentiality:** Only authorized persons are able to read the content of EMD stored on BC and IPFS.
- **Integrity:** DO is unable to tamper with the data received from DP.
- **Privacy:** Based on data stored on BC, everyone cannot know which DP DO cooperated.

- **Non-repudiation:** Parties cannot deny transactions they have submitted in the data sharing scheme.
- **Anonymity:** Everyone cannot know the real name of participants take part in the data storing and sharing schemes, and cannot distinguish which DP generated MD.

D. SYSTEM SETUP

1) THE GROUP OF DPs

The group manager chooses a security parameter λ and the group signature scheme GS to generate keys for n group member. Specifically, the group manager has a public/private key pair (PK_{GM}, SK_{GM}) ; the revocation manager owns a public key PK_{RM} and a private key SK_{RM} ; $gsk[i]$ and $IdDP[i]$ are a private key and an identifier of the i -th group member respectively, where $1 \leq i \leq n$; and the group public key gpk .

2) THE BLOCKCHAIN SYSTEM

DO, DU, and the group manager each initializes an account on BC system. Particularly, DO owns a public key PK_{DO} and a private key SK_{DO} ; DU also has a public key PK_{DU} and a private key SK_{DU} ; Similarly, the group manager has a public/private key pair (PK_{BCGM}, SK_{BCGM}) . On the BC network, users use their public key as a BC address, for instance PK_{DU} is as DU's BC address, each transaction must be signed by the transaction's initiator. The BC system provides the public BC address.

E. DATA PRODUCING

In the data producing scheme, DO transfers RD to a particular DP in the group, for instance the i -th DP. After receiving RD, DP performs the Produce algorithm to generate MD, CERT, and DPInfo. To ensure the confidentiality of MD in the data storing and data sharing schemes, DP encrypts MD to form EMD and then generates CERT on EMD. Later, DP sends these result data to DO via the security channel. After receiving the results data, DU verifies the accuracy of MD and DPInfo. In this scheme, DO and DP are considered to know each other, therefore, it is not necessary to secure the identity of each other. This means that DO knows the identifier of the DP and the group public key of the group of DPs. The data producing scheme is described in Fig. 2, which includes the following steps:

(1) DO transfers RD (in materials form or digital original data) to a particular DP of the group via a security channel.

(2) After receiving RD, DP produces MD, CERT, and DPInfo using the Produce algorithm which includes nine steps:

Step 1: DP uses the *make_proc* function and the procedure $\equiv$ to produce MD in digital form.

Step 2: DP creates the identifier of MD by using the cryptographic hash function provided by the system. The output is denoted by IdMD.

Step 3: DP performs the *Rand_Key* to generate a key K .

Step 4: DP encrypts MD using K and the encryption algorithm provided by the system. The output is EMD

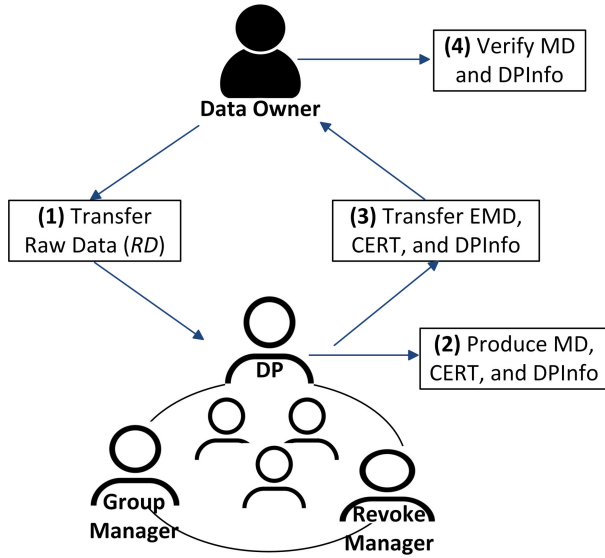


FIGURE 2. Our proposed data producing scheme.

Step 5: DP encrypts DP's $IdDP$ and K using PK_{DO} and PCS provided by the system. The output is denoted by $DPInfo$.

Step 6: DP encrypts $DPInfo$ and $IdMD$ using PK_{RM} and PCS. The output is denoted by EId .

Step 7: DP generates a signature on EMD by using the Sign algorithm of GS, the group public key gpk , and the group member secret key of DP $gsk[i]$. The output is denoted by SD .

Step 8: The certificate of MD (denoted by CERT) includes SD and EId .

Step 9: The algorithm outputs EMD, CERT, and $DPInfo$. The Produce algorithm is summarized in Algorithm 1.

Algorithm 1 Produce

Input: $RD, IdDP[i], gsk[i], gpk, PK_{DO}, PK_{RM}$

Output: EMD, CERT, $DPInfo$

- 1: Produce MD from RD
 $MD \leftarrow make_proc(RD)$
- 2: Create an identifier for MD
 $IdMD \leftarrow H(MD)$
- 3: Generate a random key K
 $K \leftarrow Rand_Key(\cdot)$
- 4: Encrypt MD using K and the encryption algorithm
 $EMD \leftarrow E_K(MD)$
- 5: Encrypt the DP's $IdDP$ and K using PK_{DO} and PCS
 $DPInfo \leftarrow PCS(IdDP[i] \parallel K, PK_{DO})$
- 6: Encrypt $DPInfo$ and $IdMD$ using PK_{RM} and PCS
 $EId \leftarrow PCS(DPInfo \parallel IdMD, PK_{RM})$
- 7: Sign on EMD using $gpk, gsk[i]$, and $GS.Sign$
 $SD \leftarrow GS.Sign(gpk, gsk[i], EMD)$
- 8: Certificate CERT includes (SD, EId)
 $CERT = (SD, EId)$
- 9: return $(EMD, CERT, DPInfo)$

```

#TX::Store_Data
From: DO's BC address
To: Public BC Address
Content:
- EMD_Link
- CERT
- DPInfo
- Paymentadd
- Price
- SC
  
```

FIGURE 3. The transaction in the data storing scheme.

(3) DP sends EMD, CERT, and $DPInfo$ to DO via a secure channel.

(4) After receiving data from DP, DO verifies the accuracy of MD and $DPInfo$ as follows:

Step 1: DO decrypts $DPInfo$ using SK_{DO} and PCS

$$IdDP[i] \parallel K \leftarrow PCS(DPInfo, SK_{DO})$$

Step 2: DO compares $IdDP[i]$ with the DP's information that DO knew before. If they are the same, go to the next step. Otherwise, stop verifying.

Step 3: DO decrypts EMD using K and the decryption algorithm:

$$MD \leftarrow D_K(EMD)$$

Step 4: DO recalculates an identifier for MD:

$$IdMD \leftarrow H(MD)$$

Step 5: DO checks the accuracy of DP's information.

$$True/False \leftarrow (PCS(DPInfo \parallel IdMD, PK_{RM}) == CERT.EId)$$

If it returns *True*, go to the next step. Otherwise, stop verifying.

Step 6: DO checks the accuracy of MD using the Verify algorithm of GS

$$True/False \leftarrow GS.Verify(gpk, CERT.SD, EMD)$$

If it returns *True*, DO has received accurate and reliable data. Otherwise, ignore the transaction.

F. DATA STORING

In this scheme, DO stores EMD on IPFS, and the access address of EMD and the related information are stored in a BC transaction. The information stored on this transaction will also serve for the data sharing scheme.

Fig. 4 presents the data storing scheme, which includes the following steps:

(1) DO uploads EMD on IPFS.

(2) After successful upload, IPFS returns the access address of EMD (denoted by EMD_Link) to DO.

(3) DO submits the $TX :: Store_Data$ transaction to the BC system, as shown in Fig. 3, which includes the following information:

- *DO's BC address*: The BC address of DO.

- *Public BC address*: The public BC address of the BC system.

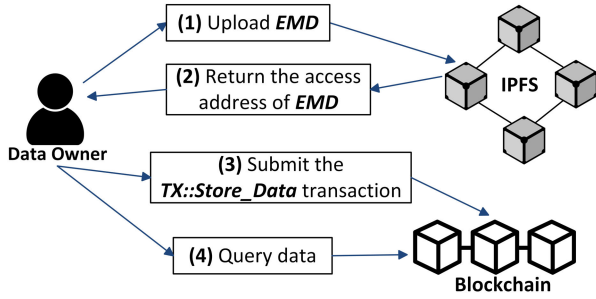


FIGURE 4. Our proposed data storing scheme.

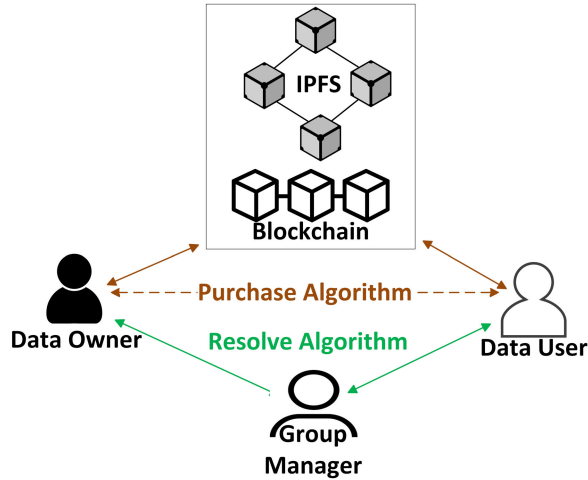


FIGURE 5. Our proposed data sharing scheme.

- *EMD_Link*, *CERT*, and *DPInfo*.
- *Paymentadd*: Payment wallet address of DO.
- *Prices*: The amount of money that a buyer has to pay to DO.
- *SC*: The smart contract address is used for the sales/purchase process.

If the DO's signature on this transaction is valid, BC miners will store this transaction on their ledger.

(4) DO lookups the *TX :: Store_Data* transaction on the BC ledger:

$$TX \leftarrow \text{Ledger}$$

Later, DO checks the query result. If *TX* is not null, DO has stored data successfully.

G. DATA SHARING

In this scheme, the data purchasing/selling process between DO and DU is conducted via the **Purchase** algorithm, and we use the **Resolve** algorithm for dispute resolution. We consider EMD as shared data and BC as data market. Everyone on the system may find and buy data that they need. In the data sharing scheme, participants have to make an escrow to the smart contract deployed by the group manager. If any party is detected cheating, his/her escrow will be lost.

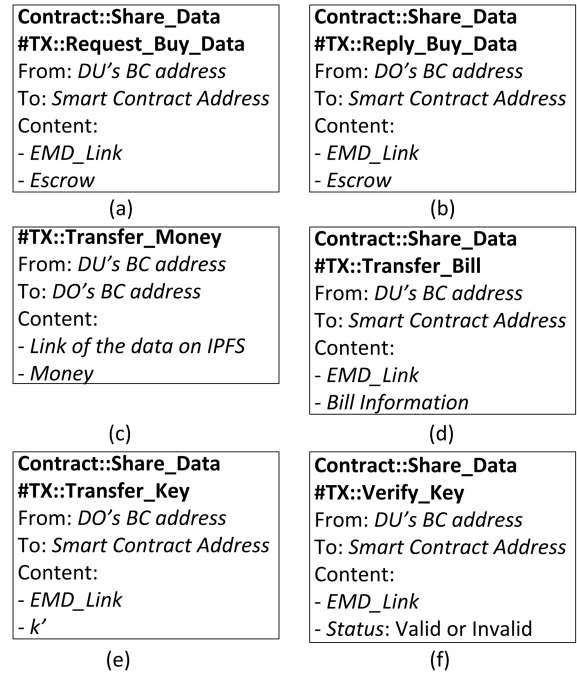


FIGURE 6. The transactions in the Share_Data contract.

Fig. 5 presents the data sharing scheme, which includes the following steps:

(1) DU performs the **Purchase** algorithm, summarized in *Algorithm 2*, to search and to buy data shared they need. Specifically, DU can verify the reliability of the shared data using the **Verify** algorithm of the group signature scheme *GS* (lines 1-3). Note that DU can only verify the validity of the shared data but cannot read its contents. If the shared data is valid, DO performs the *Contract :: Share_Data* smart contract indicated in the *TX :: Store_Data* transaction, with the *TX :: Request_Buy_Data* transaction (line 4). This operation is like submitting a data buying request to DO. In this transaction, DU also transfers an amount of money to the smart contract as an escrow asset. The contract information is notified to DO by the system application. If DO accepts the DU's request, DO makes the *TX :: Reply_Buy_Data* transaction of the *Contract :: Share_Data* smart contract and also transfers an escrow payment to the smart contract (line 6). After receiving the transaction from DO, DU makes the *TX :: Transfer_Money* transaction to DO (line 7), in which the *Money* field is the price of the shared data, and then sends the bill information to DO via the *TX :: Transfer_Bill* transaction of the *Contract :: Share_Data* smart contract (line 8). If DO has already received money according to the bill information sent from DU, DO makes the *TX :: Transfer_Key* transaction to send the secret key *K* to DU for decrypting the shared data (line 9), in which *K* is encrypted by the public-key cryptosystem and the DU's public key PK_{DU} its output denoted by k' .

$$k' \leftarrow PCS(K, PK_{DU})$$

After receiving the key information, DU uses the DU's private key to decrypt k' and get K , then using K to decrypt the shared data (EMD) (lines 10-11).

$$K \leftarrow PCS(k', SK_{DU})$$

$$MD \leftarrow D_K(EMD)$$

If K is valid, DU performs the $TX :: Verify_Key$ transaction of $Contract :: Share_Data$, in which the *Status* field is set to *Valid* (line 12-13), otherwise, both DU and DO will go to the **Resolve** algorithm (line 15). The transactions of the **Purchase** algorithm are shown in Fig. 6.

Algorithm 2 Purchase

Input: TX'
Output: MD
DU:
 1: Search shared data
 2: Download EMD on IPFS
 3: **if** ($GS.Verify(gpk, CERT.SD, EMD) == 1$)
 then
 4: Submit $TX :: Request_Buy_Data$ transaction of
 the $Contract :: Share_Data$ smart contract
 5: **end**
DO:
 6: Submit $TX :: Reply_Buy_Data$ of $Contract ::$
 $Share_Data$
DU:
 7: Submit $TX :: Transfer_Money$ transaction to DO,
 in which the *Money* field is the price of the shared
 data.
 8: Submit $TX :: Transfer_Bill$ of $Contract ::$
 $Share_Data$
DO:
 9: Submit $TX :: Transfer_Key$ of $Contract ::$
 $Share_Data$
DU:
 10: Decrypt k' and get K
 $K \leftarrow PCS(k', SK_{DU})$
 11: Decrypt EMD
 $MD \leftarrow D_K(EMD)$
 12: **if** (K is valid) **then**
 13: Submit $TX :: Verify_Key$ with the *Status* field
 is set to *Valid*
 14: **else**
 15: Go to the *Resolve* protocol
 16: **end**
 17: return (MD)

(2) When receiving the request for dispute resolution from DU, the group manager perform the **Resolve** algorithm, summarized in *Algorithm 3*, to determine who is a scammer. Because the group manager acts as the referee in the smart contract, the group manager can track all transactions of this contract and can know the information of the shared data ($EMD_Link, CERT$). In the **Resolve** algorithm, the group manager submits the $TX :: Dispute_Key_Request$ transaction of the contract to DO to require DO to provide the decryption key of the shared data (line 1). DO submits the $TX :: Dispute_Key_Reply$ transaction to the group manager,

in which K is encrypted by $PKBC_{GM}$ and PCS (line 2):

$$EK' \leftarrow PCS(K, PKBC_{GM})$$

The transaction of the **Resolve** algorithm are shown in Fig. 7. Once receiving the response from DO, the group manager decrypts EK' using $SKBC_{GM}$ and PCS to obtain the secret key K (line 5), and then uses K to decrypt EMD (line 6). If K is invalid, the group manager concludes that DO is a scammer (lines 7-8). Otherwise, the group manager encrypts K using the DU's public key PK_{DU} and PCS (line 10), it output $EK1'$. Finally, the group manager gets k' in the $TX :: Transfer_Key$ transaction, and compares k' with $EK1'$, if they are the same, DU is a scammer (DU has received a valid key but still request for dispute resolution). On the contrary, DO has sent an invalid decryption key of the shared data to DU, hence, DO is a scammer, in this case, the group manager will also send the valid key K to DU (lines 11-17).

Algorithm 3 Resolve

Input: TX'
Output: DO, DU
Group manager:
 1: Submit $TX :: Dispute_Key_Request$ to DO
DO:
 2: Submit $TX :: Dispute_Key_Reply$ to the group
 manager
Group manager:
 3: Access EMD_Link and download EMD on IPFS
 4: Get related transactions of the smart contract
 5: Decrypt EK' using $SKBC_{GM}$ and PCS to get K
 $K \leftarrow PCS(EK', SKBC_{GM})$
 6: Decrypt EMD
 $MD \leftarrow D_K(EMD)$
 7: **if** (K is invalid) **then**
 8: $SCA \leftarrow DO$
 9: **else**
 10: Encrypt K using PK_{DU} and PCS
 $EK1' \leftarrow PCS(K, PK_{DU})$
 11: Get k' in $TX :: Transfer_Key$ of $Contract ::$
 $Share_Data$
 12: **if** ($EK1' == k'$) **then**
 13: $SCA \leftarrow DU$
 14: **else**
 15: $SCA \leftarrow DO$
 16: Send K to DU
 17: **end**
 18: **end**
 19: **return** (SCA)

The rules of the data sharing scheme are as follows:

- The escrow deposit must be 2 or 3 times the price of the shared data to encourage honesty of participants.
- If DU has performed escrow and submitted the $TX :: Request_Buy_Data$ transaction, however DO doesn't make the $TX :: Reply_Buy_Data$ transaction. After a

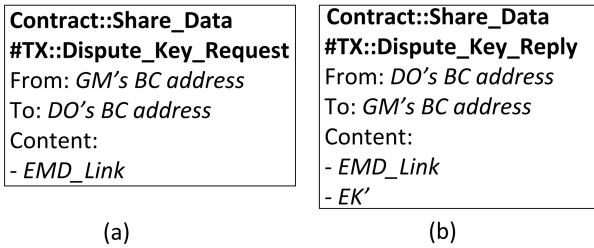


FIGURE 7. The transactions in the Resolve algorithm.

certain amount of time, the smart contract will automatically return the escrow deposit to DU.

- If both DO and DU have made escrow, however DU doesn't submit the $TX :: Transfer_Money$ transaction to DO. After a certain amount of time, the smart contract will also transfer the escrows back to DU and DO.
- If DU has already submitted the $TX :: Transfer_Bill$ transaction, but DO doesn't perform the $TX :: Transfer_Key$ of $Contract :: Share_Data$ within the allotted time, the DO's escrow will be lost, and the smart contract will also return DU's escrow to DU.
- If DU has already received a valid key from DO, but DU doesn't make the $TX :: Verify_Key$ transaction of the contract within a certain time, the secret key is considered a valid key and the contract is automatically completed.
- After the scammer has been detected by the *Resolve* algorithm, the escrow of the scammer will be lost, and the escrow in the smart contract will be returned to the honest party.

V. SECURITY ANALYSIS

This paper combines BC, IPFS, and the group signature scheme to design the data producing, data storing, and data sharing schemes, in which the data producing scheme is used to generate meaningful data for DO, and then DO can store or share this digital data on the system. Our data storing and data sharing schemes gain more benefits than the solutions surveyed in Section II. In this section, we discuss the advantages, security features, and performance features of these schemes.

A. ADVANTAGES

1) PROACTIVITY

- In the data storing scheme: DO is proactive in storing EDM on IPFS. Specifically, DO can use some of his/her devices to join the IPFS network, then upload EMD on these nodes. And DO can also remove EMD stored on his/her devices.
- In the data sharing scheme: DU can fully verify the reliability of shared data (EMD) he/she needs before performing the *Purchase* algorithm of the data sharing scheme. Particularly, DU accesses *EMD_Link* and downloads EMD. Then DU uses the *Verify* algorithm of the group signature scheme *GS* provided by the system, the group public key *gpk*, and *SD* in *CERT*

stored in the $TX :: Store_Data$ transaction to verify the reliability of EMD as follows: $True/False \leftarrow GS.Verify(gpk, CERT.SD, EMD)$. Besides, the sharing process is done directly between DU and DO without depending on any intermediaries. This process is performed by the *Purchase* algorithm of the data sharing scheme.

2) TRANSPARENCY AND FAIRNESS IN DATA SHARING

- All transactions in the data sharing scheme are recorded on the BC ledger, which means that they are publicly traceable to the associated addresses.
- Parties involved in the data sharing process have to transfer escrow to the smart contract to encourage honesty. In the event of a dispute, the group manager will resolve this dispute through the *Resolve* algorithm of the data sharing scheme. As a result, the scammer will lose his/her deposit.

B. SECURITY FEATURES

1) CONFIDENTIALITY

- In the data storing scheme, MD is encrypted by the secret key generated by DP for uploading to IPFS. To retrieve MD's content, any requestors need to know the secret key to decrypt such data. Note that the secret key is encrypted by PK_{DO} before storing on BC. Therefore, it is very challenging for the attacker to guess the secret key to decrypt and obtain MD on IPFS.
- In the data sharing scheme, the decryption key of EMD will also be encrypted by DU's public key in the smart contract. Therefore, only DU can decrypt and obtain the secret key for decrypting EMD. In dispute resolution, the decryption key of EMD is also encrypted before sending to the group manager.

2) INTEGRITY

In the data sharing scheme, *CERT* certified by DP is used to verify the validity of EMD. Therefore, DO may modify MD to form a new version MD but cannot generate a valid certificate for this new MD because DO does not have a group member key.

3) PRIVACY

From the stored data on the BC system, everyone can verify the accuracy and reliability of MD but cannot understand its content and cannot know exactly which DP in the group the DO used the service. This property is ensured by the group signature scheme.

4) NON-REPUDIATION

In the data storing and data sharing schemes, each party has to use his/her private key to sign on their BC transactions. All transactions with invalid signatures will be dropped by the miners of the BC network. Therefore, adversaries cannot impersonate anyone to perform transactions. In addition, valid BC transactions are stored in the public ledger, and data

in the BC ledger is immutable, therefore, adversaries cannot repudiate their transactions.

5) ANONYMITY

In the BC network, each user is identified by the user's public key without any additional personal information required, and the corresponding private key is used to generate signatures on transactions. Therefore, all BC nodes, DUs, and DOs could not know the real name of each other in transactions. Besides, the group signature also ensures the anonymity of DPs. Particularly, any party even the revocation manager and the group manager cannot know any identity of group members who generated and issued certificates for MD. The anonymity of the DP is revealed only if the group manager and the revocation manager cooperate together.

C. PERFORMANCE FEATURES

In the proposed system, the BC transaction system and the IPFS storage system achieve the properties as follows:

1) AVAILABILITY

In our system, both the IPFS storage system and the BC system are peer-to-peer networks with a lot of nodes in the system, therefore adversaries will be very difficult to crash these system. In the IPFS system, the availability of stored data will also be guaranteed. Specifically, when the adversaries remove EMD stored on the compromised IPFS node, EMD may be cached within a certain time at some other nodes on the IPFS network, moreover, the IPFS network is an open network, DU can completely use some of DU's devices to join the network and can activate the pinning and clustering services on these nodes to improve the availability of stored data. For the BC transaction system, because the data in the BC ledger is synchronized between miners, therefore if some miner nodes do not operate because of DoS/DDoS attacks or hardware errors, the BC system will still be maintained by other miner nodes.

2) INTEGRITY

For the IPFS storage system, data stored on IPFS is identified by the hash value of its content. This hash value is used to verify the integrity of the stored data and is also the access address of the data on IPFS. Hence, modified data will have a new access address. For the BC transaction system, data stored on the ledger is immutable and synchronized between miners. Adversaries may modify the ledger data if and only if they compromise the majority of miners or all miners, however this is very hard.

3) SCALABILITY

Both the IPFS storage system and the BC transaction system are the peer-to-peer networks, hence, the expansion is simply adding some nodes to the networks.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose three schemes: data producing, data storing, and data sharing. In the data producing scheme,

we consider RO as DP, a group manager sets up a group of DPs providing the same type of services. DP can generate MD from RD sent from DO, and then issues a certificate on EMD.

In the data storing scheme, we provide not only the confidentiality and integrity of the stored data but also the anonymity of DP and the privacy of DO which have not been fulfilled in the existing solutions. In the data sharing scheme, everyone on the system can verify the reliability of shared data before submitting a sharing request to DO. Note that everyone can only verify the reliability of the shared data but cannot read its contents. This property could not be fulfilled by existing solutions. In addition, the data sharing process is done directly between DO and DU without depending on any intermediaries.

The results of the security analysis show that the proposed schemes meet the security properties including confidentiality, integrity, privacy, non-repudiation, and anonymity.

In our future work, we will apply the proposed system to specific applications such as IoT, electronic medical records. We will then evaluate and optimize the schemes.

REFERENCES

- [1] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world from edge to core," IDC White Paper, Nov. 2018.
- [2] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.
- [3] L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1443–1451, May 2014.
- [4] T. A. Phan, J. K. Nurminen, and M. Di Francesco, "Cloud databases for Internet-of-Things data," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom)*, Sep. 2014, pp. 117–124.
- [5] K. Yasumoto, H. Yamaguchi, and H. Shigeno, "Survey of real-time processing technologies of IoT data streams," *J. Inf. Process.*, vol. 24, no. 2, pp. 195–202, 2016.
- [6] A. Kumar, N. C. Narendra, and U. Bellur, "Uploading and replicating Internet of Things (IoT) data on distributed cloud storage," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2016, pp. 670–677.
- [7] K. Hossain, M. Rahman, and S. Roy, "IoT data compression and optimization techniques in cloud storage: Current prospects and future directions," *Int. J. Cloud Appl. Comput.*, vol. 9, no. 2, pp. 43–59, Apr. 2019.
- [8] J. D. Bokefode, A. S. Bhise, P. A. Satarkar, and D. G. Modani, "Developing a secure cloud storage system for storing IoT data by applying role based encryption," *Procedia Comput. Sci.*, vol. 89, no. 1, pp. 43–50, 2016.
- [9] W. Wang, P. Xu, and L. T. Yang, "Secure data collection, storage and access in cloud-assisted IoT," *IEEE Cloud Comput.*, vol. 5, no. 4, pp. 77–88, Jul. 2018.
- [10] M. Rashid, S. A. Parah, A. R. Wani, and S. K. Gupta, "Securing E-health IoT data on cloud systems using novel extended role based access control model," in *Internet Things (IoT)*. Cham, Switzerland: Springer, 2020, pp. 473–489.
- [11] R. Arora, A. Parashar, and C. C. I. Transforming, "Secure user data in cloud computing using encryption algorithms," *Int. J. Eng. Res. Appl.*, vol. 3, no. 4, pp. 1922–1926, 2013.
- [12] M. Kantarcioglu and F. Shaon, "Securing big data in the age of AI," in *Proc. 1st IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Dec. 2019, pp. 218–220.
- [13] C. Zhang, J. Sun, X. Zhu, and Y. Fang, "Privacy and security for online social networks: Challenges and opportunities," *IEEE Netw.*, vol. 24, no. 4, pp. 13–18, Jul./Aug. 2010.
- [14] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol, CA, USA: O'Reilly Media, 2014.

- [15] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
- [16] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018.
- [17] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.
- [18] X. Zheng, R. R. Mukkamala, R. Vatrapi, and J. Ordieres-Mere, "Blockchain-based personal health data sharing system using cloud storage," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Sep. 2018, pp. 1–6.
- [19] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.
- [20] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.
- [21] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 146, Aug. 2018.
- [22] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.
- [23] M. Naz, F. A. Al-zahrani, R. Khalid, N. Javaid, A. M. Qamar, M. K. Afzal, and M. Shafiq, "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, no. 24, p. 7054, 2019.
- [24] X. Wu, Y. Han, M. Zhang, and S. Zhu, "Secure personal health records sharing based on blockchain and IPFS," in *Proc. Chin. Conf. Trusted Comput. Inf. Secur.*, Singapore: Springer, 2019, pp. 340–354.
- [25] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101653.
- [26] V. H. Hoang, E. Lehtihet, and Y. Ghamri-Doudane, "Privacy-preserving blockchain-based data sharing platform for decentralized storage systems," in *Proc. IFIP Netw. Conf. (Netw.)*, Jun. 2020, pp. 280–288.
- [27] V. Buterin. (Jul. 2015). *On Public and Private Blockchains*. Ethereum Blog. [Online]. Available: <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>
- [28] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [29] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, "Bitcoin and cryptocurrency technologies," *Curso Elaborado Pela*, to be published.
- [30] S. King and S. Nadal. (Aug. 2012). *PPCoin: Peer-to-Peer Cryptocurrency with Proof-of-Stake*. [Online]. Available: <https://decred.org/research/king2012.pdf>
- [31] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–5.
- [32] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending Bitcoin's proof of work via proof of stake [extended abstract]," in *Proc. 9th Workshop Econ. Netw., Syst. Comput.*, Austin, TX, USA, Jun. 2014, pp. 34–37.
- [33] V. Buterin, "A next-generation smart contract and decentralized application platform," White Paper 37, 2014, vol. 3.
- [34] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, no. 2014, pp. 1–32, Apr. 2014.
- [35] J. Benet, "IPFS—content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*. [Online]. Available: <http://arxiv.org/abs/1407.3561>
- [36] J. Benet and N. Greco, "Filecoin: A decentralized storage network," Protocol Labs, San Francisco, CA, USA, Tech. Rep., 2018.
- [37] M. Steichen, R. Norvill, B. F. Pontiveros, and W. Shbair, "Blockchain-based, decentralized access control for IPFS," in *Proc. IEEE Blockchain*, Jul. 2018, pp. 1499–1506.
- [38] D. Chaum and E. van Heyst, "Group signatures," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 547. Berlin, Germany: Springer, 1991, pp. 257–265.

- [39] J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant," *BRICS Rep. Ser.*, vol. 5, no. 27, pp. 11–21, Jan. 1998.



TAM T. HUYNH is currently a Lecturer with the Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Ho Chi Minh City Campus. His research interests include blockchain, the IoT, and decentralized storage.



THUC D. NGUYEN is currently a Lecturer with the Faculty of Information Technology, Vietnam National University Ho Chi Minh City—University of Science, Vietnam. His research interests include cryptography, information security, and machine learning.



THANG HOANG received the B.S. degree in computer science from Vietnam National University Ho Chi Minh City—University of Science, Ho Chi Minh City, Vietnam, in 2010, the M.E. degree in computer science from Chonnam National University, Gwangju, South Korea, in 2014, and the Ph.D. degree from the University of South Florida, in August 2020. He is currently a Postdoctoral Fellow hosted by Prof. Elaine Shi at Carnegie Mellon University (CMU) and a Research Associate hosted by Prof. Attila A. Yavuz at the University of South Florida (USF). His research interests include applied cryptography and machine learning, with special interests in privacy-enhancing technologies. He is also interested in zero-knowledge proofs, multi-party computation, oblivious RAM, and searchable encryption.



LAM TRAN received the B.S. degree in computer science from Vietnam National University Ho Chi Minh City—University of Science, Ho Chi Minh City, Vietnam, in September 2012, and the M.E. degree in computer science from Chonnam National University, Gwangju, South Korea, in August 2017, where he is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence Convergence. His research interests include inertial sensor-based gait biometrics, privacy enhancing for biometric systems, and machine learning.



DEOKJAI CHOI received the B.S. degree from the Department of Computer Engineering, Seoul National University, in 1982, the M.S. degree from the Department of Computer Science, KAIST, South Korea, in 1984, and the Ph.D. degree from the Department of Computer Science and Telecommunications, University of Missouri—Kansas City, Kansas City, MO, USA, in 1995. He is currently a Full Professor with the Department of Artificial Intelligence Convergence, Chonnam National University, South Korea. His research interests include mobile security, human activity profile, biometric authentication, and network security.

...