TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
Tập 17, Số 9 (2020): 1713-1724
ISSN: 1859-3100

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
Vol. 17, No. 9 (2020): 1713-1724
Website: *http://journal.hcmue.edu.vn*

Research Article

A SOLUTION FOR PRIVACY-PRESERVING DATA SHARING ON PEER-TO-PEER NETWORKS

Huynh Thanh Tam[1*], Dang Hai Van[2], Nguyen Dinh Thuc[2]

[1]Faculty of Information Technology
Posts and Telecommunications Institute of Technology, HCM, Vietnam
[2]Faculty of Information Technology, University of Science, VNU-HCMC, Vietnam
[*]Corresponding author: Huynh Thanh Tam – Email: tamht@ptithcm.edu.vn
Received: May 23, 2020; Revised: September 22, 2020; Accepted: September 25, 2020

ABSTRACT

Digital data, in many fields, such as banking or medicine, must be protected when storing and sharing on the Internet. Moreover, in some cases, the integrity of shared data must also be guaranteed such that there is no modification on the shared data. In this paper, the following scenario is analysed: A pharmaceutical company which is doing research on a X cancer, broadcasting on the Internet that the company wants to buy DNA data of those having the X. The questions are: (1) How should a user share securely and anonymously her/his DNA data to the company? and (2) How could the company determine the integrity of receiving DNA data before paying for an online contract? These problems can be solved based on the blockchain technology. A protocol is proposed for guaranteeing the privacy, integrity, and authentication of data shared on the peer-to-peer network. The designed protocol is a combination of three components, including the blockchain (BC) technology, the interplanetary file system (IPFS), and cryptosystems. Data are transferred directly to the recipients without using any central system. The information of the data sharing process is published on a blockchain, and users can verify the integrity and authentication of the original data but cannot know the content of the data. The experimental results show that the protocol satisfies the security requirements and can be easily implemented in practice.

Keywords: IPFS; blockchain; data sharing; privacy

1. Introduction

Sharing data is a process of transferring data from a data owner (DO) to a shared person (user), which is a regular activity on the network. Privacy-preserving for data sharing is to guarantee that only the participants in the sharing process can understand the shared content. This work can be done by using traditional methods, such as sending an email with encrypted attachments combined with a key sharing protocol that is easily conducted based on the concept of public key cryptosystem. Data integrity is based on trust between the two partners participating in the exchange process. For instance, doctors/hospitals absolutely believe that medical records are received from their patients are integrity.

Cite this article as: Huynh Thanh Tam, Dang Hai Van, & Nguyen Dinh Thuc (2020). A solution for privacy-preserving data sharing on peer-to-peer networks. *Ho Chi Minh City University of Education Journal of Science, 17*(9), 1713-1724.

What happens when a person provides his/her medical records to a pharmaceutical research firm to receive income? In this case, the patient can completely change certain information, especially the information that is detrimental to himself/herself. The anonymous solution can be used, by adopting blockchain (BC) technology, to encourage people who sell information to provide truthful information after eliminating the inferred information for identification. Actually, this solution is more damaging for buyers because dishonest sellers can create many BC addresses to anonymize and many fake records, and each record is used for each BC address. Therefore, buyers have no way to eliminate this problem.

In this paper, another protocol is proposed that not only allows people to verify the integrity and authentication of the exchanged information in public transactions but also ensures the privacy of data. In this protocol, the content provider who creates the original content, such as a doctor or a research lab, provides the content along with a certificate for such content. After that, the original content in an encrypted form and the certificate are transmitted on the network to the data owner. The secret key is transmitted in a blockchain transaction to the data owner for decryption. The data owner can share the data with another partner. In order to verify the integrity and authentication of the received content, the partner must use the public key of the content provider who generated such content.

The contributions of the paper are two-fold as follows: (1) an introduction of a new peer-to-peer data sharing protocol that is used to share data securely between two partners and how to implement and evaluate its effectiveness. The rest of this paper is organized as follows. Section 2 presents the foundations of the proposed protocol. The protocol is described in section 3. Section 4 shows some experimental results. Section 5 compares the protocol with other similar solutions. Finally, the conclusions are given in Section 6.

2. Foundations

2.1. IPFS

IPFS, proposed by Juan Benet (2014), is a peer-to-peer distributed file system in which nodes communicate directly with each other without relying on any central system. Each node is initialized a public/private key pair by RSA-1024 cryptosystem, where the private key is used for signing data, and the public key is used for verifying the signature and generating NodeID. Before exchanging data, nodes must be initialized connections by exchanging their public key and NodeID with each other. If the NodeID matches with the public key being exchanged, the connection is established. Basically, there are three types of nodes in the IPFS network, namely: client node, retrieval miner node, and storage miner node (Huynh et al., 2019).

Data storage: The uploaded data are divided into objects having a data structure including two fields: the data field is used to store binary data while the links field contains an array of links that point to other related objects. Each link is composed of three components: Name, Hash, and Size, where Name is as an alias of the link, Hash contains the

hash value of the linked object, and the total size of the linked object is put in the Size field. Each object is identified by a unique hash value of its content, can store up to 256 kilobytes (KB) of data. Thus, if the size of uploaded files is less than the size of an object, the storage nodes only use one object with the empty link field. Otherwise, the file is split and put into chunks of 256 KB, and using the Merkle directed acyclic graph (DAG) data structure for managing its chunks (Steichen et al., 2018).

Routing: Each miner node owns a distributed hash table (DHT) for storing the routing information such as IP address, UDP port, and NodeID. In order to look up or store objects, nodes can use four remote procedure calls including PING, STORE, FIND_NODE, and FIND_VALUE. Currently, the S/Kademlia DHT, an extension of kademlia protocol, is used to build the routing table (Baumgart et al., 2007).

IPFS clustering: In order to improve the redundancy of data on the network, some storage miner nodes, called cluster nodes, are configured the cluster feature (Ipfs.io, 2019). Data can be selected to store long term and replicated between cluster nodes. Normally, cluster nodes have a large storage space and high-speed processing capacity.

Inter-Planetary Name System (IPNS): When modifying the content of an object to form a new version, a new hash value will also be generated to identify for such object. In order to access the same link to access a mutable object, the IPNS is used to map between a NodeId generated from a public key and a hash value of a certain object, and this hash value is signed by the corresponding private key. An IPNS link could be considered as a website address, and we can use the TXT record in Domain Name System to configure a domain to it (Huynh et al., 2019).

2.2. Blockchain

BC technology has attracted significant attentions from both researchers and government organizations because of the benefits that the technology brings about such as anonymity, transparency, decentralization, and auditability. It could be applied in many different fields, including finance, medicine, transportation, agriculture, and Internet of Things (Huynh et al., 2019; Zheng et al., 2018; Conti et al. 2018).

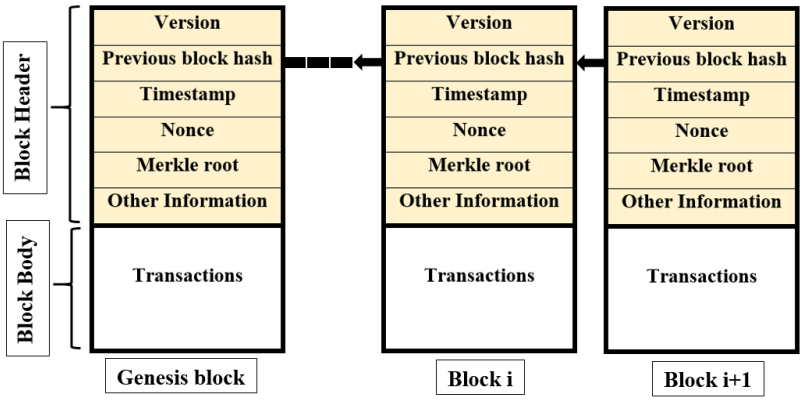

Fig. 1. *An example of a blockchain*

A BC can be viewed as a linked list of blocks, each block is linked to its predecessor via a hash pointer containing a hash value of the predecessor at a fixed time. This value is used to verify the integrity of the previous block. The first block of the chain is called the genesis block which has no parent block, so the hash value field of the block is initialed by the constructor of the BC network. Each block has two main components, the block header which contains management information of the block and chain and the block body which holds a list of transactions as shown in Figure 1.

There are two types of node on a BC network, the user node only performs transactions, and the miner node is responsible for validating transactions and creating new blocks (called the mining process). Each node has a key pair. The secret key is used to sign its transactions, and the public key is published to the network used to validate its transactions. All valid transactions are mined and securely recorded in the ledger which is stored at miner nodes. In order to synchronize data on the ledger, there are two popular consensus algorithms used: Proof of Work and Proof of Stake (Huynh et al., 2019).

Ethereum is a new distributed BC network which provides a decentralized turing-complete platform called Ethereum Virtual Machine (EVM). The programs running on EVM are called smart contracts which reside on top of the blockchain network. Generally, a smart contract is a set of rules under which participants agree to use, which is written in bytecode and executed automatically when specific conditions are met. The solidity is one of the most popular languages for programming smart contracts (Buterin, 2014).

2.1 Cryptography

Cryptography is the science of using methods to protect data privacy, integrity, confidentiality, authentication, and non-repudiation. In cryptography, the process of transforming plaintext data to ciphertext is called encryption, and the reverse of the data encryption process is referred to decryption. Depending on the number of keys used, encryption algorithms are classified into two main categories: asymmetric and symmetric encryption algorithms (Katz et al., 2014).

Symmetric encryption uses a single key for both encryption and decryption, is used for encrypting large amounts of data efficiently as compared to asymmetric encryption. However, the weakness is the exchange of a shared secret key between two parties (Agrawal et al., 2012). Some symmetric algorithms include DES, TRIPLE DES, AES, RC4, etc. In asymmetric encryption algorithms, encryption and decryption are performed by two different keys that are related mathematically. In which a public key is published and a private key is kept secretly. One of the keys can be used for encryption, and the other key is used for decryption. Some asymmetric algorithms include RSA, DSA, Diffie-Hellman, etc. These algorithms are suitable for encrypting small plaintexts (Singh, 2013; Nie et al., 2009; Hossain et al, 2016).

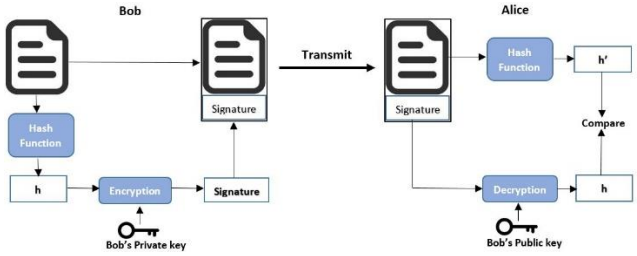

Fig. 2. *A general model of digital signature process*

Digital signature is a mechanism used to validate the authentication and integrity of the digital message. A hash function is usually used in the digital signature model which can be used to map a variable-length message into a fixed-size hash value. As shown in Figure 2, the integrity of the message is checked by comparing the hash values of h and h'. While asymmetric cryptography provides the authentication for digital signature (Merkle et al., 1989; Bellare et al., 1999).

However, data exchanged between Bob and Alice are in a plaintext form. To ensure confidentiality, data must be encrypted by a symmetric algorithm before transmitting on the network, and the secret key is sent to receivers for decryption through a secure channel.

3. Peer-to-peer data sharing protocol

Figure 3 illustrates the general diagram of the data sharing and creating protocol on the peer-to-peer network. That means the data generated by an authorized person or organization will be directly transferred to the data owner. The transactional information and related information are transparent on a BC network.

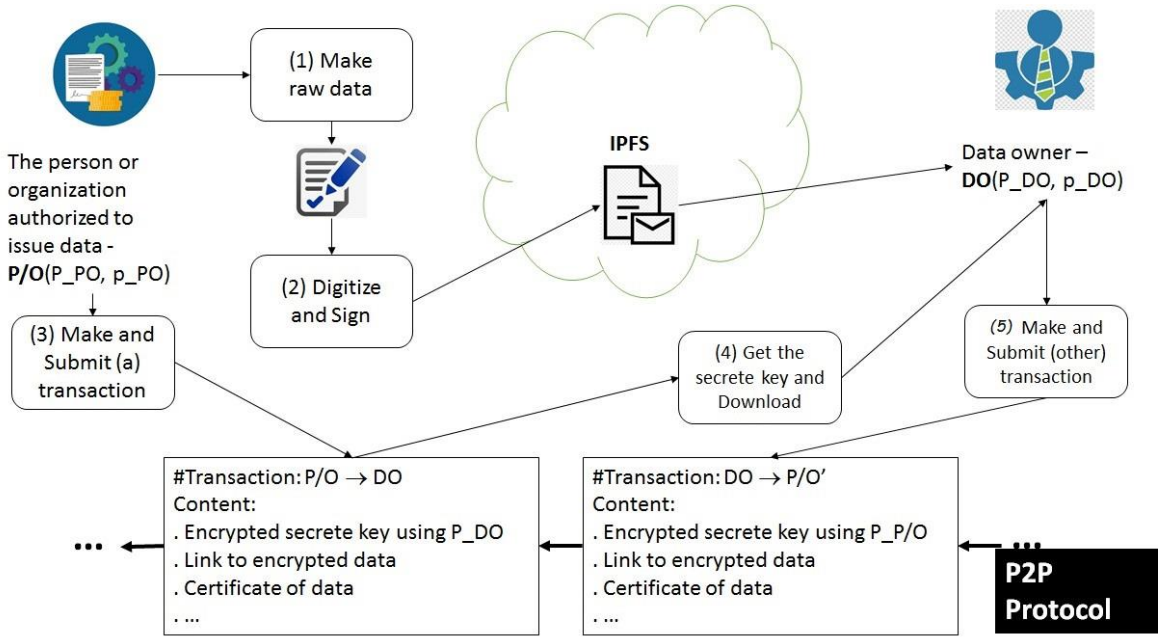

Fig. 3. *The diagram of the peer-to-peer data sharing protocol*

In Figure 3, P/O represents for an authorized person or an organization (such as laboratory, doctor, hospital, or deoxyribonucleic acid (DNA) sequencing center), each P/O owns a public key P_PO and a private key p_PO; Data owner, such as patient, is denoted by DO, has a public key P_DO and a private key p_DO. In this protocol, sensitive data (such as DNA or health information) in encrypted forms and a certificate after being created by the P/O will be delivered to the DO where the certificate is used to prove the security and integrity of the data.

The process of creating and getting data of the protocol as follows:

(1) The P/O creates raw data 5.

(2) Digitize and sign: This step includes the following activities:

(i) Convert the raw data into the digital format.

(ii) Encrypt the digitized data with a secret key (k1) and the selected symmetric cryptography. The P/O can use the symmetric cryptosystem provided by the system. In case of using a private cryptosystem, the P/O must provide this cryptosystem or the source code of the cryptosystem to the DO.

(iii) Issue a certificate to mask the integrity of the data: The certificate must be created in a regular mechanism so that people in the BC system can check the integrity of the original data without understanding such content.

(iv) The ciphertext and the certificate (CA) are uploaded to the cloud (in this paper, we use IPFS as the cloud).

(3) Create and submit a transaction to the BC network: In a transaction, besides the information about the sender and receiver addresses, it also has the following necessary information:

(i) The secret key, for decrypting the original data, is encrypted by the DO's public key.

(ii) The path of the encrypted data and the certificate on the cloud.

(iii) The certificate of the data.

(4) Get the secret key and download the data: When receiving a transaction from the P/O, the DO verifies the validity of the certification and then using the p_DO to decrypt and get a secret key (k1). Then downloading and decrypting data.

(5) In case the DO wants to share the data with a partner, the DO uses the partner's public key to encrypt the k1. Then executing a transaction (*encrypted key, link to encrypted data, certificate*) to the partner.

4. Experiment results

To demonstrate the effectiveness of the protocol, the peer-to-peer network is built and implemented using IPFS and ethereum BC as shown in Figure 4. The IPFS network is used to store and distribute data to patients. Three virtual machines are used (OS CentOS, RAM 1GB, CPU 2.6 Ghz) to build the private IPFS network. All nodes are configured the clustering feature, and one node acts as the IPFS gateway. For the BC network, Ganache is

run to create a virtual ethereum BC. By default, Ganache generates 10 different accounts. Each account wallet contains 100 ETH. In order to build the decentralized application (DApp), reaction is used with the libraries web3, crypto, and ipfs-api. The smart contract is written by the solidity programming language and using truffle to deploy them to the BC network.

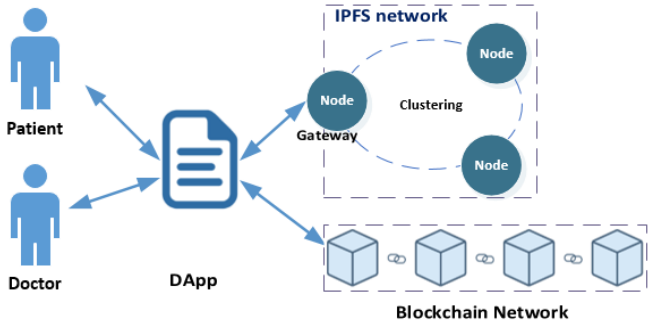

Fig. 4. *The peer-to-peer data sharing model*

In the crypto library, the AES-256 algorithm in CBC mode is used to encrypt data at step 1 of the protocol. The RSA-1024 is used to generate key pairs for P/Os and DOs. The SHA-256 function is used in the certificate generating process.

In order to generate a certificate for a health record of a patient, firstly the health record is encrypted by the AES-256 algorithm. Then, the output is hashed by the SHA-256 to form a unique value. Finally, the P/O uses the p_PO to sign on the hash value. The general formula is expressed as follows: $CA = sign(p\_PO, H(E(k, health\ record)))$, where H is a hash function, E is an encryption algorithm, and k is a secret key. The CA is concatenated with the ciphertext of the health record, the output result is uploaded to the IPFS. Besides, CA is also submitted to the BC network.

To verify the integrity of data, from a transaction on the BC network, people perform the following steps:

(1) Download the data on the IPFS from the given link in the transaction. Then separating the certificate field (denoted CA') and the ciphertext field (denoted c') from the data.

(2) Calculate: $h = D(P\_PO, CA)$ and $h' = H(c')$, where D is the decryption function, CA is extracted from the transaction. If $((CA = CA')$ and $(h = h'))$ then data is valid.

In this way, people in the BC system only check the validity of the data without understanding such content on the IPFS.

The solidity language is used to build smart contract for storing data from P/Os as follows:

```
1   pragma solidity >=0.4.21 <0.6.0;
2   import "github.com/Arachnid/solidity-stringutils/strings.sol";
3   contract DataSharing {
4       mapping (address => string) message;
```

```
5       using strings for *;
6        event AddData(string,string,string);
7       function SubmitData(address recipient, string  Key, string  link, string Cert) public
8   {
9           string memory data;
10          data=Key.toSlice().concat("|".toSlice());
11          data=data.toSlice().concat(link.toSlice());
12          data=data.toSlice().concat("|".toSlice());
13          data=data.toSlice().concat(Cert.toSlice());
14           message[recipient] = data;
15           emit AddData(Key, link, Cert);
16      }
17      function GetData() public view returns (string memory) {
18          return (message[msg.sender]);
19      }
20  }
```

Some images of the Dapp:

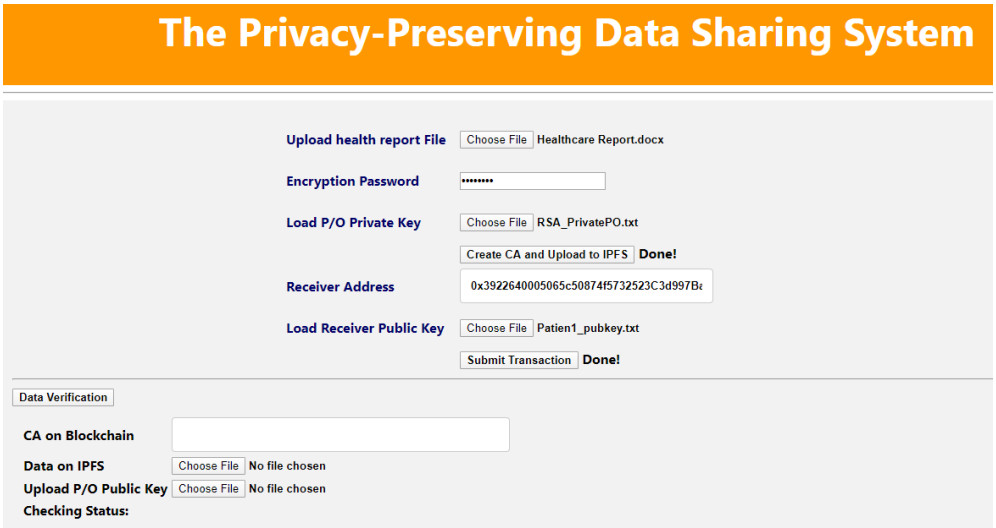

Fig. 5. *The interface of the DApp*

In the Encryption Password field, the doctor has to set a password to generate a secret key, as shown in Figure 5. The information of the transaction is shown in Figures 6 and 7.

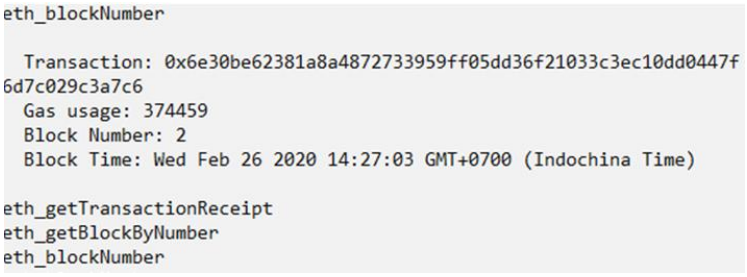

Fig. 6. *The hash value of the transaction on the BC network*

Fig. 7. *The detail information of the transaction*

Fig. 8. *The decoded result of the transaction*

People can access the data in the IPFS system based on the hash value '*QmREDpNEjyoLuaMsYw7mgHFYSGEzfJF6eBbMfBjiCpUYXW*'. However, they cannot understand the content.

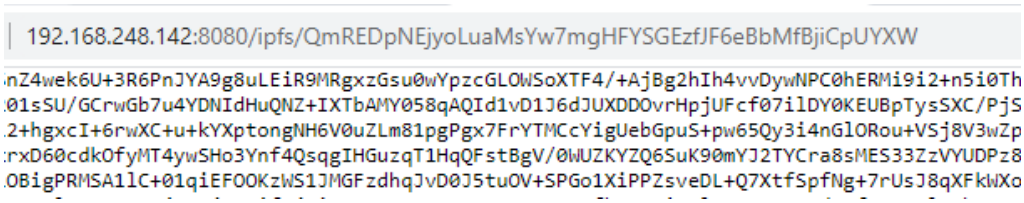

Fig. 9. *The data on the IPFS system*

The DApp also provides the "**Data Verification**" feature used to check the CA and data in the IPFS system.

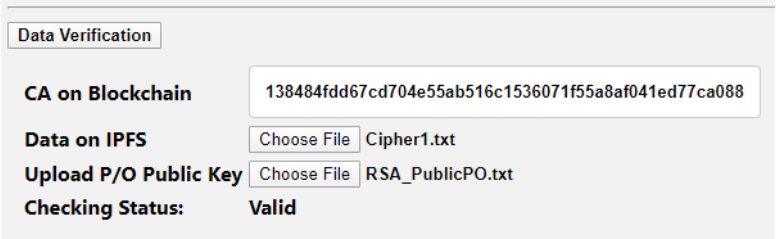

Fig. 10. *The Data Verification feature of DApp*

The size of the medical records is different, however, when uploading records to IPFS, their access addresses are fixed-length hash strings. Blockchain transactions only contain data access addresses on IPFS, thus the size of the transactions does not depend on the size of the medical records. In this experiment result, the size of a transaction is 2 kilobytes. The content of a transaction is shown in Figure 11.


```
> eth.getTransaction('0x8bedf08278f343c6f3fac5df46f8b2dde1eeb5cde6dfd47e172577cbafb63f82')
{
  blockHash: "0xf66ef4b016d84624870649cb990d398c49af7401893838f6640db01b9ae2b480",
  blockNumber: 9,
  from: "0xa545a5d4ec27fdd9061b0da84efaea48f69ed80b",
  gas: 1000000,
  gasPrice: 20000000000,
  hash: "0x8bedf08278f343c6f3fac5df46f8b2dde1eeb5cde6dfd47e172577cbafb63f82",
  input: "0x16aae90e0000000000000000000000000a9986c3d49c1ed8e3d140f5d611fc2d75a1d979f0000000000000000000000000000000000000000000000000000000000000000
0000000000000000800000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000001c0000000000000000000000000000000000000000000000000000000000000ac466241376e4e376236784859723064646333584c3150666
d6549344c42324346734b6c5a4e415a4d7635386e4d5272496331452b706e4c367475793736534b67626d346e7253785761506a586c3172394e5162777948755a
4c3146557068394c3066594a594e6d324f587248457a4c5339503356525a58573947737342624773656736426879336263 5a3631673061355245 2b6e737257736
471344c686f4d6b654f6634677a716b2f6b6f3d0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000002e516d516f72526e32314c3637673262794c367147557750316469395350795 46c766754577443547142315867 6b5000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
643037343366366613738383736653832303761623562306337656430393633393936643235333364323330363337326635333434326634616533303261633066 3
766373435326263333266656238393838633383356132653261313533343264303037633232336534613634313664383739633631646366623533613661386463
613561653736323934363563646633393465653037363165313736316630663335346464626465396662393333393961353430663963626534633764663165643
334663866343463336531653635343665356530663939623138323136386334613762663 0",
  nonce: 6,
  r: "0xf443f4734095abe0197cbdd3f3e74a43627602e644e4accdcd5112b640d6b98e",
  s: "0x56723e579f7e33efc6582d7b434efdbccb6e400440dbe9f7bb295ee1f1d11ccc",
  to: "0xa9986c3d49c1ed8e3d140f5d611fc2d75a1d979f",
  transactionIndex: 0,
  v: "0x26",
  value: 0
}
>
```


Fig. 11. *The content of a transaction*

5. Comparison

In a study by Makhdoom et al. (2020), data are shared with stakeholders. For instance, a user's health data can be shared with a medical center, primary medical center, health secretariat, and insurance provider that the user has registered. Data privacy is guaranteed by encryption algorithms and access control policies in smart contracts.

In a study by Hoang, V. H. (2020), this data sharing method only ensures the confidentiality of the data, and data integrity is based on trust between the two partners participating in the exchange process.

The solution proposed in this study can ensure the confidentiality of the shared data as well as data integrity. Furthermore, any user can verify the integrity and reliability of the shared data but cannot understand the content.

6. Conclusions

Privacy-preserving for data sharing plays a crucial role in peer-to-peer networks. The proposed protocol is designed based on existing technologies such as IPFS, BC, and cryptography algorithms. The protocol requires the content providers to publish their identification on the BC network. In many fields, this is necessary to manage the sources that generate original contents. The experimental results show that the protocol operates efficiently, can be easily implemented in practice.

❖ ***Conflict of Interest:*** *Authors have no conflict of interest to declare.*

❖ ***Acknowledgments:*** *This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number NCM2019-18-01.*

REFERENCES

Agrawal, M., & Mishra, P. (2012). *A comparative survey on symmetric key encryption techniques*. International Journal on Computer Science and Engineering, *4*(5), 877.

Baumgart, I., & Mies, S. (2007). *S/kademlia: A practicable approach towards secure key-based routing*. In 2007 International Conference on Parallel and Distributed Systems, IEEE, 1-8.

Bellare, M., & Miner, S. K. (1999). *A forward-secure digital signature scheme*. In Annual International Cryptology Conference, Springer, Berlin, Heidelberg, 431-448.

Benet, J. (2014). *Ipfs-content addressed, versioned, p2p file system*. ArXiv preprint arXiv:1407.3561.

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform.* White paper, *3*(37).

Conti, M., Kumar, S., Lal, C., & Ruj, S. (2018). *A survey on security and privacy issues of bitcoin*. IEEE Communications Surveys & Tutorials, *20*(4), 3416-3452.

Hossain, M. A., Hossain, M. B, Uddin, M. S., & Imtiaz, S. M. (2016). *Performance analysis of different cryptography algorithms.* International Journal of Advanced Research in Computer Science and Software Engineering, *6*(3).

Huynh, T. T., Nguyen, T. D., & Tan, H. (2019). *A Decentralized Solution for Web Hosting*. In 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), 82-87.

Huynh, T. T., Nguyen, T. D., & Tan, H. (2019). *A Survey on Security and Privacy Issues of Blockchain Technology*. In 2019 International Conference on System Science and Engineering (ICSSE), 368-373.

IPFS.IO. (2019). IPFS cluster. Retrieved from https://cluster.ipfs.io.

Katz, J., & Lindell, Y. (2014). *Introduction to modern cryptography*. CRC press.

Merkle, R. C. (1989). *A certified digital signature*. In Conference on the Theory and Application of Cryptology, Springer, New York, NY, 218-238.

Nie, T., & Zhang, T. (2009). *A study of DES and Blowfish encryption algorithm*. In Tencon 2009-2009 IEEE Region 10 Conference, 1-4.

Singh, G. (2013). *A study of encryption algorithms (RSA, DES, 3DES and AES) for information security.* International Journal of Computer Applications, *67*(19).

Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. (2018). *Blockchain-based, decentralized access control for IPFS*. In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 1499-1506.

Zheng, Z., Xie, S., Dai, H. N., & Wang, H. (2018). *Blockchain challenges and opportunities: A survey*. In International Journal of Web and Grid Services, *14*(4), 352-375.

Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., & Ni, W. (2020). *PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities*. Computers & Security, 88, 101653.

Hoang, V. H., Lehtihet, E., & Ghamri-Doudane, Y. (2020, June). *Privacy-Preserving Blockchain-Based Data Sharing Platform for Decentralized Storage Systems*. In 2020 IFIP Networking Conference (Networking) (280-288). IEEE.

GIẢI PHÁP CHIA SẺ DỮ LIỆU ĐẢM BẢO TÍNH RIÊNG TƯ TRÊN MẠNG NGANG HÀNG

Huỳnh Thanh Tâm[1*], Đặng Hải Vân[2], Nguyễn Đình Thúc[2]

[1]Khoa Công nghệ Thông tin,
Học viện Công nghệ Bưu chính Viễn thông cơ sở tại Thành phố Hồ Chí Minh, Việt Nam
[2] Khoa Công nghệ Thông tin
Trường Đại học Khoa học Tự nhiên, ĐHQG TPHCM, Việt Nam
[*]Tác giả liên hệ: Huỳnh Thanh Tâm – Email: tamht@ptithcm.edu.vn

Ngày nhận bài: 23-5-2020; ngày nhận bài sửa: 22-9-2020, ngày chấp nhận đăng: 25-9-2020

TÓM TẮT

Dữ liệu số, trong nhiều lĩnh vực, như ngân hàng hoặc dược phẩm, cần phải được bảo vệ khi lưu trữ và chia sẻ trên mạng internet. Tuy nhiên, trong một vài trường hợp, tính toàn vẹn của dữ liệu chia sẻ cũng phải được đảm bảo để không có sự sửa đổi trên dữ liệu được chia sẻ. Trong bài báo này, chúng tôi xem xét ngữ cảnh sau: một công ti được phẩm đang nghiên cứu về bệnh ung thư X, quảng bá trên internet rằng công ty muốn mua dữ liệu DNA của những người mắc bệnh X. Các câu hỏi đặt ra là: (1) Làm thế nào một người dùng chia sẻ dữ liệu DNS của họ một cách an toàn và ẩn danh cho công ti? Và (2) Làm thế nào công ti có thể xác định tính toàn vẹn của dữ liệu DNA nhận được trước khi thanh toán cho một hợp đồng trực tuyến? Những vấn đề này có thể được giải quyết dựa trên công nghệ blockchain. Chúng tôi đề xuất một giao thức để đảm bảo tính riêng tư, tính toàn vẹn và xác thực của việc chia sẻ dữ liệu trên mạng ngang hàng. Giao thức được thiết kế là sự kết hợp của ba thành phần, bao gồm công nghệ blockchain (BC), hệ thống tệp liên hành tinh (IPFS), hệ thống mật mã. Dữ liệu được chuyển trực tiếp đến người nhận mà không cần sử dụng bất kì hệ thống trung tâm nào. Thông tin của quá trình chia sẻ dữ liệu được công bố trên một blockchain, người dùng có thể xác minh tính toàn vẹn và xác thực của dữ liệu gốc nhưng không thể biết được nội dung của dữ liệu. Kết quả thử nghiệm cho thấy giao thức của chúng tôi đáp ứng tốt các yêu cầu bảo mật, có thể dễ dàng triển khai trong thực tế.

Từ khóa: IPFS; blockchain; chia sẻ dữ liệu; tính riêng tư