Blockchain-based Secure Storage System for Medical Image Data

Chu-Hsing Lin
Department of Computer Science,
Tunghai University
Taichung, Taiwan
chlin@go.thu.edu.tw

Sin-Ping Li
Department of Computer Science,
Tunghai University
Taichung, Taiwan
zx785999@gmail.com

Yu-Chiao Lin
China Medical University Hospital
Taichung, Taiwan
u107070301@cmu.edu.tw

Chiao-Hsu Tsai
China Medical University Hospital
Taichung, Taiwan
u107070303@cmu.edu.tw

2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB) | 979-8-3503-3386-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEIB57887.2023.10170051

***Abstract*—A large number of patient images are generated every day in medical applications such as ultrasound, Computer Tomography scans, X-Ray, and so on. Medical data is related to patient privacy and personal rights. Thus, the security and privacy of medical images become an important issue. In order to comply with the regulations of the Personal Data Protection Act, medical images need to be protected under a secure method. Therefore, we proposed a secure storage method for chest X-Ray images from Kaggle based on blockchain technology. We developed a smart contract to control role-based access permissions and authenticate medical images. We performed a cryptographic operation on the X-Ray image after each medical examination and stored the image fingerprint in the blockchain. Artificial intelligence was used to identify pneumonia-related diseases. We compared the required X-Ray retrieval time with the conventional PACS systems. The experimental results showed that the overhead of the proposed scheme using blockchain was only about 5 %.**

Keywords—medical image data, blockchain, smart contract, Dapp.

I. Introduction

A large number of patient images are generated every day in medical applications including ultrasound, Computer Tomography Scans, X-Ray, and so on. Due to the rapid development of information technology, huge data storage is required [1]. Since medical information is related to patient privacy and personal rights, the security and privacy of medical images are important. Nevertheless, the storage of medical image data still relies on the centralized Electronic Medical Record (EMR) for storage. To comply with the regulations of the Personal Data Protection Act, medical images need to be protected under a secure method. In this study, we proposed a secure storage method for medical image data based on blockchain technology.

In addition to a secure transaction recording system, the blockchain is used as a data storage system with an encryption mechanism. [2] When a node makes a content change, other nodes start validating. This change is established when the number of verifications reaches more than 51% of the total number of nodes [3]. In this way, any modification of the data on the blockchain is guaranteed by a rigorous and secure encryption mechanism. Since the modification is completed under a certain degree of verification, the original data is not easy to be tampered with or illegally modified. In this study, we implemented the proposed method using the smart contract technology of Ethereum [4]. We developed a blockchain-based medical image data storage system, to make patient image data more secure.

II. Background Knowledge

A. Blockchain

Blockchain is a distributed ledger system. Participants are connected through a peer-to-peer network, and all information is released in the form of a broadcast. There are two roles: common node and billing node [4]. Ordinary nodes are used to operate transactions and other actions, while bookkeeping nodes are responsible for bookkeeping services and maintaining ledgers. The distributed ledger connected by the blockchain allows both parties to record the transaction more efficiently, and the transaction content can be permanently queried and verified. The blockchain ecology is constantly evolving, and it is used for Bitcoin and other virtual currency transactions belonging to blockchain 1.0. Blockchain 2.0 focuses on the development of "smart contracts", which is a technology developed by Ethereum [5]. Blockchain 3.0 is gradually emerging after the rise of Artificial Intelligence (AI) technology [6]. The technology used in this research is the Ethereum smart contract that belongs to blockchain 2.0.

The features of blockchain are briefly described as follows.

- Decentralized database: Data is verified and traded with each other.
- Peer-to-Peer transmission: The nodes can communicate, store and transmit information with each other directly.
- Anonymous and transparent: users are identified by a code of more than 30 characters and remain anonymous.
- Not tampered: after the transaction data is written, it will be stored permanently and cannot be changed.
- Logic algorithm: users can customize the algorithm to automatically start node transactions.

In a blockchain system, each block is highly correlated with each other. As long as the content of one of the blocks changes, it needs to be re-verified, and the same is true for new blocks. There is a lot of information in each block, including the current block label, previous block hash value, current block hash value, exhaustive guess value (nonce), timestamp, transaction content hash value and transaction content to ensure the data integrity [7,8]. For the security of information, the transaction content is hashed twice in each block. In this research, the image data is hash converted first and then uploaded to the blockchain, so is the security of patient data. In addition, regarding the verification method on the blockchain, each node is used to compete to calculate the exhaustive guess value (nonce) of the new block. To ensure the correctness of the content in the block, the number of verification nodes must exceed 51% of the total nodes. The structure of blockchain is shown in Fig. 1.

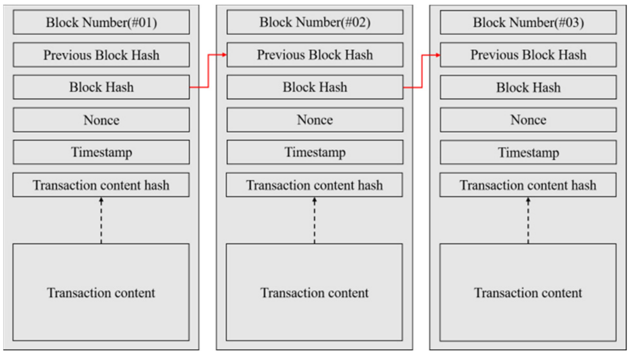


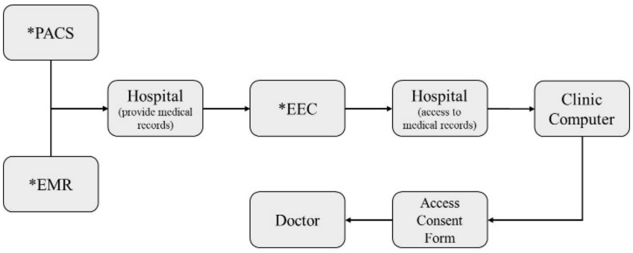
Fig. 1. Blockchain structure.

B. Smart Contract

Ethereum, a blockchain platform started in 2015, was named "A Next-Generation Smart Contract and Decentralized Application Platform" in its white paper. Although smart contracts were proposed by Nick Szabo in the early 1990s, they did not receive much attention at the time until Ethereum re-proposed and applied them to various fields, and even became the so-called "Blockchain 2.0" [9]. Areas in which smart contracts are applied to include financial transactions, weather forecasts, flight control, currency exchange, and payments on a prorated basis.

C. Electronic Medical Record (EMR)

Electronic Medical Record (EMR), since Article 69 of the "Medical Law" was amended on January 29, 2013, has become popular in medical institutions in Taiwan. Though it is costly, it allows the reduction of time, convenience of data analysis, and so forth [10,11]. The electronic medical record system is used for doctors to edit the electronic medical record after diagnosing the patient and upload the relevant examination results such as X-ray images. Medical records are exchanged in the host computer of the hospital. If doctors access medical records, they must be through the host computer under a certain authority. Its operation diagram is shown in Fig. 2.



*PACS: Picture Archiving and Communication System
*EMR: Electronic Medical Record
*EEC: EMR Exchange Center

Fig. 2. Electronic medical record.

D. Picture Archiving and Communication System (PACS)

The operation of the entire EMR system is dedicated to storing and transmitting patient image data, Picture Archiving, and Communication System (PACS). In this system, after obtaining image data such as X-Ray, image capture and conversion are performed. To cooperate with the EMR system, the image is converted into Service-Object Pair (SOP) standard Digital Imaging and Communications in Medicine (DICOM), and then functions such as saving, recalling, and transmitting can be performed. They are shown in Figs. 3 and 4, respectively.

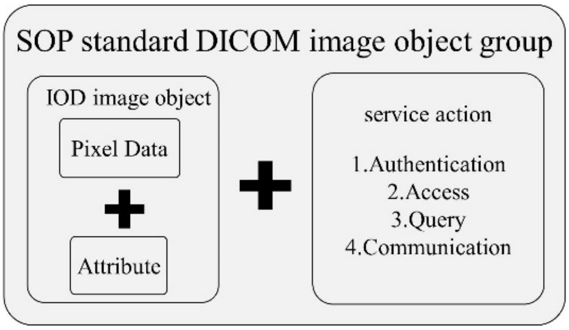


Fig. 3. SOP standard DICOM image object group.

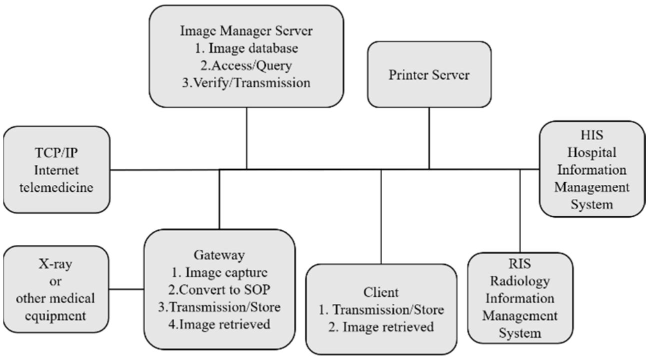


Fig. 4. Picture Archiving and Communication System (PACS).

III. Research Methods

To enhance the security and integrity of the existing PACS, improved functions are proposed as follows: writing smart contracts to automatically verify role permissions, deploying the smart contract to the blockchain, performing the hash conversion on the image data, and uploading it to the blockchain.

A. System Development Platform

1) Node.js

Currently maintained by the OpenJS Foundation, node.js is a cross-platform and open-source execution environment for running JavaScript on the server side. It is a high-performance and easy-to-expand web application development framework. It has several outstanding features. Each user uses the web server, and high-performance "asynchronous I/O" does not delay waiting and supports third-party modules with low power consumption.

2) Truffle

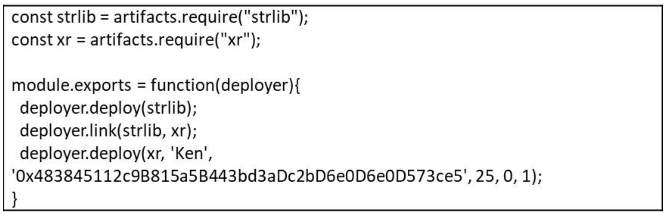
Truffle is currently the mainstream Ethereum development framework. Its development language is JavaScript to support the compilation, deployment, and testing of smart contracts. In the development of smart contracts, JS or Solidity is used to write and automatically test. Truffle manages the deployment switch between the public chain and the private chain.

Truffle only needs to use the command line in Windows to execute the command "$ npm install truffle -g" after the

Node.js installation is finished to complete the installation. After the installation is complete, it continues to enter the following command. "$ mkdir Dapp_xray" to create a new folder of Dapp_xray, "$ cd Dapp_xray" to move the absolute path to the folder of Dapp_xray, and "$ truffle init" to initialize a new Ethereum project. After executing the above commands, files appear in the folder of Dapp_xray.

Then, three Solidity files are created in contracts, which are xray.sol, xr.sol and strlib.sol. xray.sol is the parent contract of the X-Ray contract, defining state variables, enumeration (enum), structure (struct), modification word (modifier), event (event), and internal functions. xr.sol is an X-Ray contract, providing various functions for operating this contract, including Get, Set and Remove operations. strlib.sol is a string library. After writing the contract, 1_initial_migration.js is executed in the migrations folder to configure the settings required for deployment. The settings are shown in Fig. 5. After the settings are completed, the following commands are executed.

"$ truffle compile" compiles the smart contract.



```
const strlib = artifacts.require("strlib");
const xr = artifacts.require("xr");

module.exports = function(deployer){
  deployer.deploy(strlib);
  deployer.link(strlib, xr);
  deployer.deploy(xr, 'Ken',
'0x483845112c9B815a5B443bd3aDc2bD6e0D6e0D573ce5', 25, 0, 1);
}
```

Fig. 5. Related settings before deployment.

After the compilation is completed, truffle-config.js sets the connection to the blockchain. The setting content is related to the package Ganache. After the setting is completed, the following command is operated. "$ truffle migrate" deploys the smart contract.

3) Ganache

Ganache starts a virtual Ethereum blockchain and is combined with the suite Truffle introduced in the previous section to conduct a series of virtual tests. Ganache simulates the Ethereum blockchain, and the developers do not need to set up private nodes. It views the status, address, key, transaction, and balance of all current nodes. The log output of the internal blockchain can also be viewed at any time and configure different mining solution. After going through the deployment contract instructions of Truffle in the previous section, the node accounts on Ganache start trading.

B. System Operation Process

Figure 6 shows the architecture diagram of this system. The front-end part is based on the original medical image storage and transmission system (PACS) to operate, and the back-end converts the program through the hash and passes the role verification of the smart contract. Finally, the image is uploaded to the blockchain.

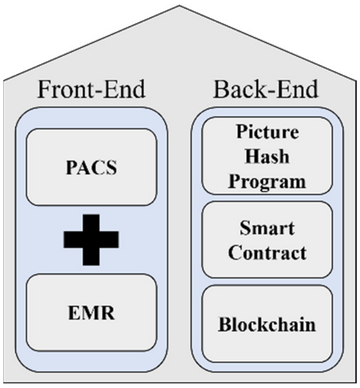


Fig. 6. System architecture.

In this system, the role of MetaMask is to obtain blockchain accounts and provide users with a place to record every data modification. After registering MetaMask, the user enters the smart contract for permission verification (Fig. 7). Since the smart contract allows all doctors in the hospital to do permission verification in this system, the account address of the doctor does not match the role. Instead, it is classified as a patient and has the right to query medical records.

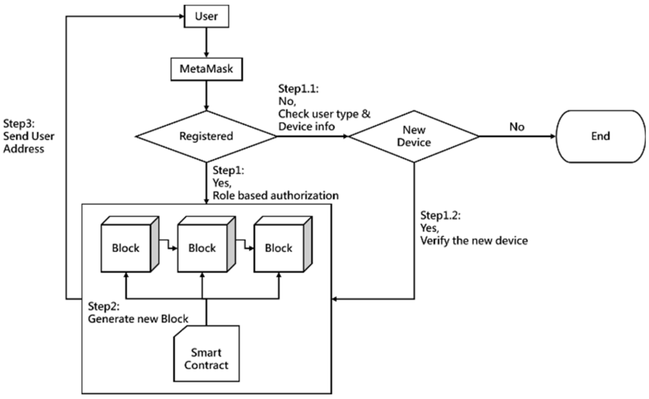


Fig. 7. Users registration process.

In addition, after the registration is completed, both patients and doctors can query the image data. However, the blockchain is only used in this system to ensure the security of data storage. In the query process shown in Fig. 8, doctors can check the data with the doctor after the patient obtains the hash value of the imaging data given by the doctor.

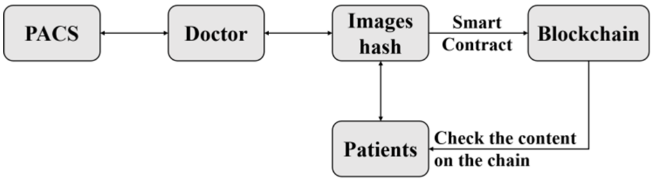


Fig. 8. Image data query/check process.

IV. Implementation and Analysis

Truffle's instructions are used to complete the deployment of the smart contract and then perform the pre-work of setting permissions and data. In addition, the hash value of the patient's X-Ray image must be obtained before performing the pre-work. For an experiment, we used the Chest X-Ray Images data set from Kaggle shown in Fig. 9. We used Python to write a program that converts pictures into hash values added to the system. The operation results are shown in Fig. 10.

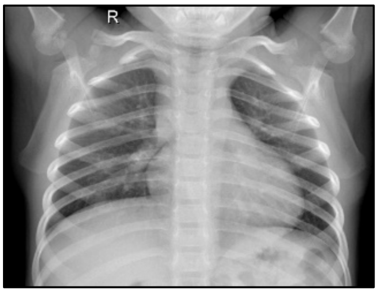
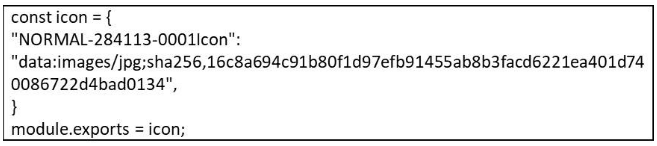


Fig. 9. Chest X-Ray images.



```
const icon = {
"NORMAL-284113-0001Icon":
"data:images/jpg;sha256,16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d74
0086722d4bad0134",
}
module.exports = icon;
```

Fig. 10. Hash value of patient’s X-Ray image.

A. Pre-work

After the deployment was completed, we typed the following command to enter the console mode to conduct the simulated transactions: >truffle console. To switch accounts to initiate transactions at any time, we declared the variables to install all accounts: >let accounts. Then, we used the built-in web3.js function library in the console mode, which was a function library commonly used by DApps to query accounts and conduct operations such as >web3.eth.getAccounts(function(err,res) { accounts = res; }).

For convenience, we stored the number of declared edges into the corresponding variable.

```
>let gov = accounts[0]
>let host = accounts[1]
>let doctor = accounts[2]
```

In this way, all account addresses in the variable accounts were defined. Finally, the smart contract was obtained, and the smart contract xr was declared as xry.

```
>let xry
>xr.deployed().then(instance => xry = instance)
```

B. Permission Setting

For permission setting, Dr. Lee’s authority was given. The “1” in the command line refers to the doctor in the UserType enumeration: >xry.setPermission(doctor, ‘Dr.Lee’, 1, true)

C. Data Setting

After setting permissions, Dr.Lee's account is used to set up Ken's examination image data. We took the chest X-Ray images as an example for the experiment. The instructions for setting the patient image data are as follows. The parameters in the command line are the inspection site and the hash value of the image.

```
>xry.setXrayhash(‘chest’,
‘16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d7400
86722d4bad0134’, { from: doctor })
```

Then, the host account is used to set up the birthday (birthday) and contact information (contact) in the personal information, the command lines are as follows. The first line is for setting the birthday and the second is for the contact information.

```
>xry.setBirthday(‘19970522’, { from: host })
>xry.setContact(‘0912-321-456’, { from: host })
```

D. Query Data

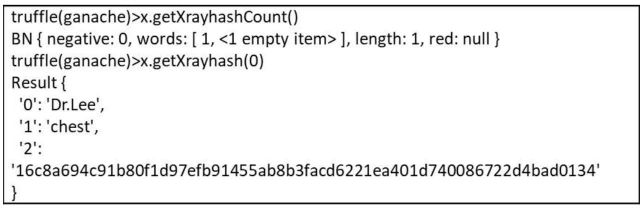
After the above settings are completed, we queried the patient's image-related information. The command line to query the patient's basic information was as follows.

```
>xry.profile()
```

Before querying the image data, the array field must be obtained first using the following two command lines.

```
>xry.getXrayhashCount()
>xry.getXrayhash(0)
```

After the above two instructions, we conducted the command to query the image data. The query command and the result are shown in Fig. 11.



```
truffle(ganache)>x.getXrayhashCount()
BN { negative: 0, words: [ 1, <1 empty item> ], length: 1, red: null }
truffle(ganache)>x.getXrayhash(0)
Result {
 '0': 'Dr.Lee',
 '1': 'chest',
 '2':
'16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d740086722d4bad0134'
}
```

Fig. 11. Query the patient's image data.

E. Permissions Remove

When doctors resign, their authorities are removed. After the authority is removed, the doctor cannot modify the medical record data for the patient. The instruction is as follows: >xry.removePermission(doctor). With the above command, it is tested whether the permission has been removed. By entering the following command, the image cannot be modified by the doctor: > xry.setXrayhash(‘chest’, ‘16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d7400 86722d4bad0134’, { from: doctor }). Through all the above command operations, users interact with the system. For example, the patient and the doctor can query the data. After typing the command, the results are compared with the EMR system and the patient's medical records can be verified.

F. Performance Analysis

In this section, we illustrate the development cost and analyze and compare the storage performance and security of our system. Currently, the system proposed in this study is more suitable for small or medium-sized medical institutions. The hardware and software environment for the experiments in this paper is shown in Table I.

TABLE I. EXPERIMENTAL ENVIRONMENT

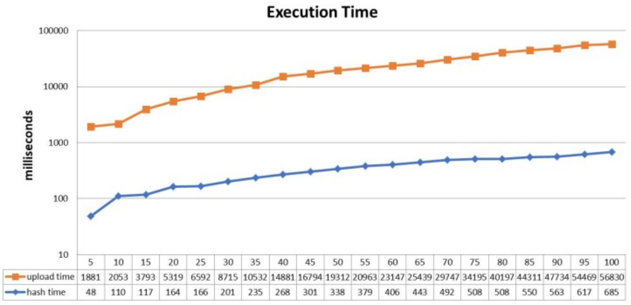
CPU	Intel® Core™ i7-7700HQ CPU @2.80GHz
Graphics card	NVIDIA GeForce GTX 1050
Memory	12.0 GB
OS	Windows 10
Blockchain (test)	Ethereum (Ganache)
Blockchain nodes	10

1) Cost Analysis

The architecture proposed is based on the existing PACS to improve its data security. In order not to waste time and cost, we present the application scenarios in small and medium-sized medical institutions. In terms of personnel costs, there is no need to replace or increase new personnel since it is used on the original system. The most important additional cost in the system is the virtual currency fee and time required for the construction of the private chain. It is also recommended to use the Ethereum private chain to set up, and after the setup is completed, the Genesis block needs to be used to mine to obtain Ethereum. This method considers the cost of the equipment required for mining. Finally, the follow-up operating costs do not require special talents or maintenance but regular node mining to ensure that the private chain can trade and store data normally.

2) Time Analysis

Figure 12 shows the time for hash conversion and uploading to the blockchain for different numbers of images. When uploading to the blockchain, nodes need to verify each other, so it takes a lot of time. The test environment in this experiment was 10 nodes. For example, the uploading time was 56830 ms and the hash time was 685 ms with 100 images.



	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
upload time	1881	2053	3793	5319	6592	8715	10532	14881	16794	19312	20963	23147	25439	29747	34195	40197	44311	47734	54469	56830
hash time	48	110	117	164	166	201	235	268	301	338	379	406	443	492	508	508	550	563	617	685

Fig. 12. Hash time & upload time for different numbers of images.

In Table II, we compared the X-Ray retrieval time required for the three systems. The overhead of the proposed scheme is about 5 % ((4.1-3.9)/3.9=0.05).

TABLE II. SYSTEM COMPARISON

Systems	Time (minutes)
Traditional X-Ray	13.0
PACS	3.9
PACS + our scheme	4.1

G. Security Analysis

The proposed system is based on the PACS to improve security. There is not much difference in performance in data transmission. In terms of data security, we use the characteristics of the blockchain that is difficult to be tampered with. Therefore, the security is improved compared with the traditional X-Ray film operation mode and the current PACS.

As shown in Table III, we compared the security and reliability of the three systems. First of all, the difficulty of tampering refers to the modification of data without the consent of relevant units such as attending doctors and patients. Taking traditional X-Rays as an example, it is easy for patients' image data to be exchanged. Moreover, in PACS, when the central server is hacked, patient data can be destroyed. This system utilizes the decentralized storage of the blockchain and requires verification of more than 1/2 of the total number of nodes to strengthen the original PACS, making it difficult for patient data to be tampered with. Further, system reliability relies on the system operation process when there is human error or a single computer node failure in the overall system. The greater the reliability, the less affected the system operation. Data confidentiality refers to the degree to which the data cannot be stolen and read in the database. The higher the confidentiality, the less likely the data is stolen and read. Finally, for the distributed denial of service attack (DDoS), the attacker uses multiple controlled sources to generate a large number of packet requests, which eventually makes the target server unable to load and causes failure.

TABLE III. SECURITY COMPARISON

	Traditional X-Ray	PACS	PACS + our scheme
Tampering	Easy	Medium	Hard
Reliability	Unreliable	Unreliable	Reliable
Confidentiality	Medium	Medium	High
DDoS Attack		Easy	Hard

V. CONCLUSION

We propose a novel application in the field of blockchain that combines the high security of blockchain with the existing high-usage EMR system to verify patient image data to reduce the risk of being tampered with. In the verification process, the most important task is the authorization verification. We use the smart contract technology from Ethereum to solve the problem of user authorization and the combination of Truffle Suite and MetaMask to complete the setting and query in the private test blockchain. The function of simulating medical records and removing specific permissions continue to develop the front-end user interface to make it more user-friendly. For information security and privacy protection, medical images need to be protected under a secure method and to comply with the regulations of the Personal Data Protection Act. We focus on the secure storage of medical image data. There are many medical applications to develop for the use of blockchain technologies in the future.

ACKNOWLEDGMENT

This work was supported by grants from the National Science and Technology Council, Taiwan, the grant number MOST 111-2221-E-029 -016.

REFERENCES

[1] Langer, S.G. (2011) Challenges for data storage in medical imaging research. Journal of Digital Imaging. 24, 203–207.

[2] Li, R., T. Song, B. Mei, Li, H., Cheng, X. and Sun, L. (2019) Blockchain for large-scale Internet of Things data storage and protection. In IEEE Transactions on Services Computing, vol. 12, no. 5, 762-771, 1 Sept.-Oct. 2019, doi: 10.1109/TSC.2018.2853167.

[3] Aggarwal, S. and Kumar, N. (2021) Chapter Twenty - Attacks on blockchain working model. Editor(s): Shubhani Aggarwal, Neeraj Kumar, Pethuru Raj,Advances in Computers, Elsevier,Volume 121, 2021, 399-410.

[4] Vujičić, D., Jagodić, D. and Ranđić, S. (2018) Blockchain technology, bitcoin, and Ethereum: A brief overview, 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2018, 1-6

[5] Chen, J., Xia, X., Lo, D., Grundy, J., Luo, X. and Chen, T. (2022) Defining smart contract defects on Ethereum. in IEEE Transactions on Software Engineering, vol. 48, no. 1, 327-345, 1 Jan. 2022.

[6] Silvano, W. F. and Marcelino, R. (2020) Iota Tangle: A cryptocurrency to communicate Internet-of-Things data, Future Generation Computer Systems, Volume 112, 2020, 307-319.

[7] Nirjhor, M. K. I., Yousuf, M. A., and Mhaboob, M. S. (2021). Electronic medical record data sharing through authentication and integrity management. In 2021 2nd IEEE International Conference on

Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 308-313).

[8] Johnson, M., Jones, M., Shervey, M., Dudley, J. T., and Zimmerman, N. (2019). Building a secure biomedical data sharing decentralized app (DApp): tutorial. Journal of medical Internet research, 21(10), e13601.

[9] Nizamuddin, N., Salah, K., Azad, M. A., Arshad, J., and Rehman, M. H. (2019). Decentralized document version control using ethereum blockchain and IPFS. Computers & Electrical Engineering, 76, 183-197.

[10] Madine, M. M., Battah, A. A., Yaqoob, I., Salah, K., Jayaraman, R., Al-Hammadi, Y., and Ellahham, S. (2020). Blockchain for giving patients control over their medical records. IEEE Access, 8, 193102-193115.

[11] Sun, J., Yao, X., Wang, S., and Wu, Y. (2020). Blockchain-based secure storage and access scheme for electronic medical records in IPFS. IEEE Access, 8, 59389-59401.