ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

Safekeeping of Medical Imaging Data with a Blockchain-Based System

[1] V. Rajshekhar, [2] M. Srinija, [3] P. Abisheik, [4] K. Pranav, [5] B. Sudeer, [6] Mrs. P. Revathy, [7] Mrs. B. Mrudula,

[1,2,3,4,5]UG Scholar, Dept. of CS, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[6] Assistant Professor, Dept. of CSE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[7] Assistant Professor, Dept. of EEE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

Abstract—

Medical applications including X-ray, ultrasound, computer tomography scans, and so on produce a massive amount of patient pictures daily. Patients' right to privacy and autonomy are directly tied to their medical records. This highlights the need of protecting the confidentiality of medical photographs. A secure technique of protecting medical photographs is necessary for compliance with the provisions of the Personal Data Protection Act. Hence, we suggested a blockchain-based safe storage solution for Kaggle chest X-ray pictures. In order to verify medical photos and manage role-based access rights, we built a smart contract. After every medical checkup, we encrypted the X-ray and recorded it on the blockchain as a unique fingerprint. In order to detect illnesses associated with pneumonia, artificial intelligence was used. Using standard PACS systems as a benchmark, we evaluated the time needed for X-ray retrieval. According to the findings of the experiments, the suggested blockchain approach only had a 5% overhead. Topics—dapp, smart contract, blockchain, medical picture data.

I. INTRODUCTION

Medical applications such as X-ray, ultrasound, computer tomography scans, and so on produce a massive amount of patient pictures daily. There is a pressing need for massive amounts of data storage space as a result of the exponential growth of IT [1]. Ensuring the security and privacy of medical pictures is crucial, since medical information is closely tied to patients' personal rights and privacy. The centralized Electronic Medical Record (EMR) continues to be the primary repository for medical imaging data. Medical photos must be securely stored in order to meet the requirements of the Personal Data Protection Act. We suggested a blockchain-based safe storage approach for medical picture data in this research. The blockchain is used as both an encrypted data storage system and a safe method of recording transactions. As soon as one node modifies the material, other nodes begin verifying it. When the number of verifications exceeds 51% of the total nodes, this modification is implemented [3]. So, a strong encryption method ensures that no one can alter the data on the blockchain. The original data is difficult to alter or unlawfully alter since the alteration is done under a certain degree of verification. Using Ethereum's smart contract technology, we put the suggested solution into action in this research [4]. To ensure the safety of patient picture data, we created a system to store medical images using blockchain technology.

II. BACKGROUND KNOWLEDGE

A. Blockchain

The blockchain is a decentralized database. All data is disseminated when participants are linked via a peer-to-peer network. One is the common node, while the other is the billing node [4]. Bookkeeping nodes provide bookkeeping services and maintain ledgers, while ordinary nodes process transactions and other activities. Blockchain technology enables efficient recording of transactions by all parties involved and permanent querying and verification of transaction information via a distributed ledger. Bitcoin and other digital currency transactions that are part of blockchain 1.0 take place on the ever-changing blockchain ecosystem. An Ethereum-developed technology called "smart contracts" is at the center of Blockchain 2.0's development efforts [5]. Following the success of AI technology, Blockchain 3.0 is starting to take shape [6]. This study makes use of blockchain 2.0 technology, namely the Ethereum smart contract. What follows is a synopsis of blockchain's key characteristics. Data is validated and exchanged between each other in a decentralized database. Direct communication, storage, and transport of data between nodes is possible in a peer-to-peer network. Transparent and anonymous: users are identifiable using a code that exceeds 30 characters while maintaining their anonymity. The transaction data will be saved permanently and cannot be altered after it has been written. The logic algorithm may be adjusted by users so that node transactions can be initiated automatically. Every block in a blockchain system is strongly connected to every other block. All blocks must be re-verified if their content changes, and new blocks must also undergo this process. To guarantee the data integrity, each block contains a lot of information, such as the label of the current block, the hash value of the previous block, the hash value of the current block, an exhaustive guess value (nonce), a timestamp, and the hash value and content of the transactions [7,8]. Each block uses a two-pass hash algorithm to secure the contents of the transaction. The confidentiality of patient data is ensured in this study by hash converting the

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

picture data before uploading it to the blockchain. Furthermore, in reference to the blockchain's verification mechanism, every node competes to determine the next block's exhaustive guess value (once). More than half of the nodes in the network must be verification nodes for the block's contents to be considered authentic. Figure 1 shows the blockchain architecture.

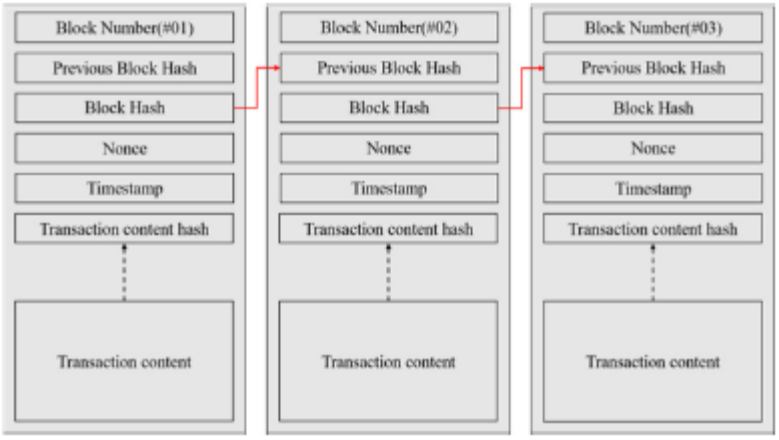

Fig. 1. Blockchain structure.

B. Smart Contract

"A Next-Generation Smart Contract and Decentralized Application Platform" was the moniker given to the blockchain platform Ethereum in its 2015 white paper. Smart contracts were first introduced by Nick Szabo in the early 1990s, but they were mostly ignored until Ethereum re-presented them, applied them to several domains, and even became the so-called "Blockchain 2.0" [9]. Financial transactions, aircraft control, weather predictions, currency exchange, and prorated payments are some of the areas where smart contracts are used

Section C. Electronic Medical Record (EMR) Taiwanese healthcare facilities have embraced electronic medical records (EMR) after the amendment of Article 69 of the "Medical Law" on January 29, 2013. The time savings, ease of data analysis, and other benefits outweigh the expense [10,11]. After making a diagnosis, clinicians may utilize the EMR system to update the patient's record and add pertinent test findings, including X-ray pictures. All of the patient's information is shared on the hospital's main server. Authorization from the host computer is required for physicians to access patient records. Figure 2 shows its operational diagram.

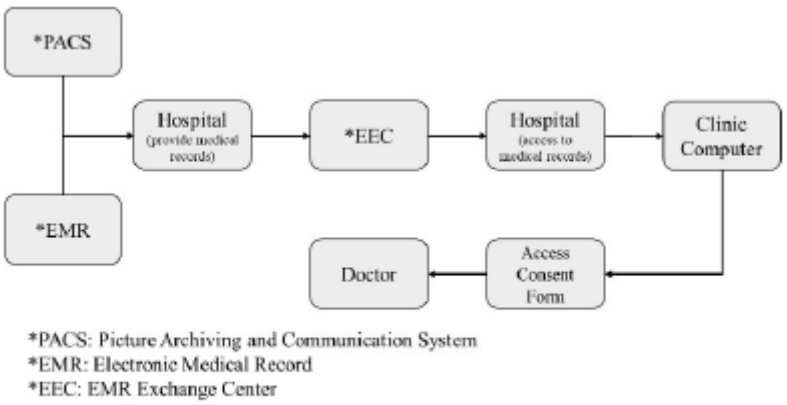

Fig. 2. Electronic medical record.

D. Picture Archiving and Communication System (PACS)

The Picture Archiving and Communication System (PACS) is the backbone of the electronic medical record (EMR) system, which stores and transmits patient picture data. This system captures and converts images after acquiring image data, such X-rays. Images are transformed into DICOM format, which is the Service-Object Pair (SOP) standard for Digital Imaging and Communications in Medicine, so that they may be used in conjunction with the EMR system for tasks like storing, retrieving, and sending. Figures 3 and 4, correspondingly, show them.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

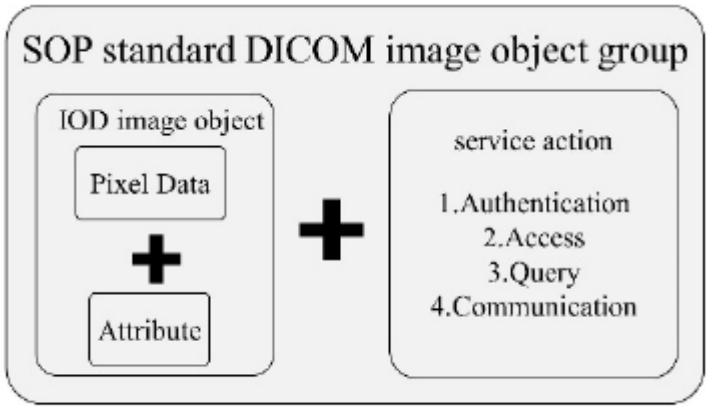

Fig. 3. SOP standard DICOM image object group.

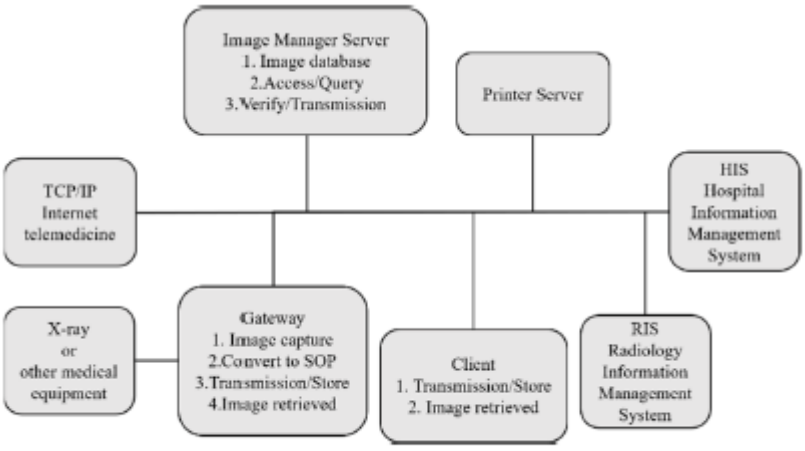

Fig. 4. Picture Archiving and Communication System (PACS).

III. RESEARCH METHODS

Here are some suggested enhancements to the current PACS that would make it more secure and reliable: creating smart contracts to check role permissions automatically, deploying those contracts to the blockchain, hash converting image data, and finally uploading the data to the blockchain. Platform for System Development (A) One, Node.js Currently in the care of the OpenJS Foundation, node.js is an open-source, cross-platform environment for server-side JavaScript execution. It is a framework for building web applications that is both highly performant and easily extensible. It has a number of remarkable qualities. The web server is used by every user, and third-party modules that are energy efficient and include high-performance "asynchronous I/O" do not prolong the waiting process. 2) Truffles At the moment, the most popular framework for developing Ethereum is Truffle. To facilitate the creation, distribution, and evaluation of smart contracts, JavaScript serves as its primary language of development. Writing and automatically testing smart contracts is done using either JS or Solidity. The Truffle network controls the handoff between the public and private chains during deployment. After the Node.js installation is complete, Truffle can be installed simply by using the command "$ npm install truffle -g" from the Windows command line. It keeps entering the following command after installation is finished. The following commands will establish a new folder named "Dapp_xray": "$ mkdir Dapp_xray"; "$ cd Dapp_xray" will shift the absolute path to that folder; and "$ truffle init" will start a new Ethereum project. The files will appear in the Dapp Xray folder once you execute the instructions above. Then, three Solidity files—xray.sol, xr.sol, and strlib.sol—are generated in the contracts. State variables, enumeration, structure, modification words, events, and internal functions are defined in the xray.sol parent contract of the X-Ray contract. The X-Ray contract xr.sol provides a number of operations for managing the contract, including as Get, Set, and Remove. The string library is called strlib.sol. Executing 1_initial_migration.js in the migrations folder configures the parameters necessary for deployment after authoring the contract. In Figure 5, you can see the configuration. The following instructions are run when the settings are finished.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

"$ 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.

When the build is finished, truffle-config.js establishes the blockchain connection. Ganache is the package that the setting is associated with. The following command is executed when the setting is finished. For the smart contract to be deployed, run "$truffle migrate". Choc ganache Together, Ganache and the Truffle suite, which was introduced in the previous section, launch a virtual Ethereum blockchain and run a battery of tests. The makers of Ganache don't have to worry about setting up private nodes since it mimics the Ethereum network. It checks the balance, status, address, key, and transactions of all the nodes that are currently online. Additionally, many mining solutions may have their log output from the internal blockchain configured and inspected at any time. The node accounts on Ganache begin trading after you have gone through Truffle's deployment contract requirements in the previous section. B. Procedure for System Operation The system's architectural diagram is shown in Figure 6. The front-end relies on the original PACS to function, while the back-end processes the program by hashing it and then sends it on to the smart contract for role verification. The last step is to post the picture on the blockchain.

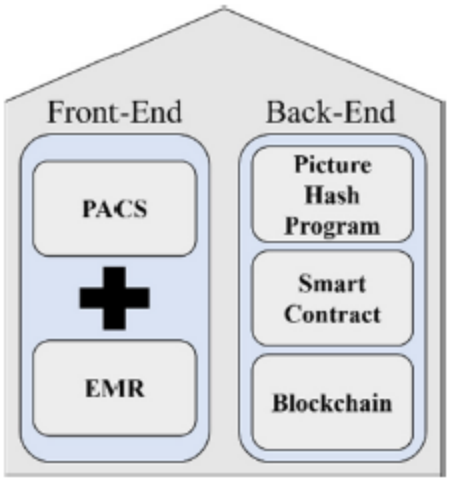

Fig. 6. System architecture.

The function of MetaMask in this setup is to acquire blockchain accounts and provide users a location to document each data alteration. The user inputs the smart contract to verify permissions after registering MetaMask (Fig. 7). The doctor's account doesn't match the role, but all of the hospital's physicians may use the smart contract to verify permissions in this system. Rather, it is considered a patient and granted access to medical records.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

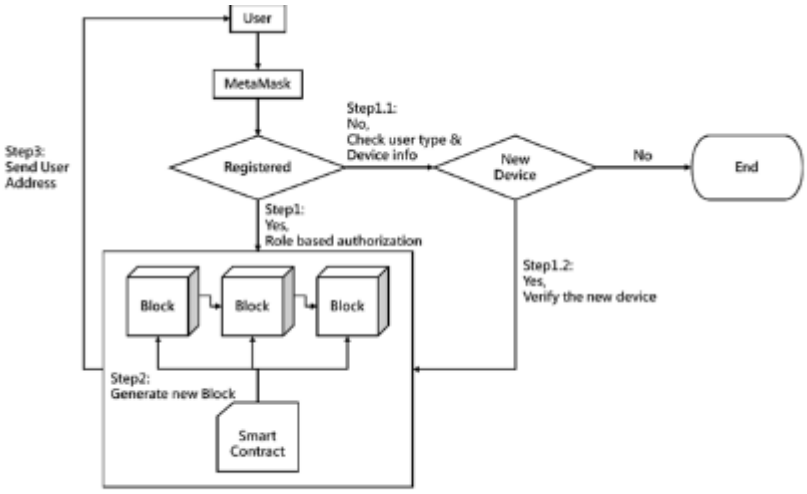

Fig. 7. Users registration process.

Also, after registration is finished, medical professionals and patients alike will be able to access and query the picture data. Nevertheless, the only purpose of the blockchain in this system is to guarantee the safety of data storage. Following the patient's acquisition of the hash value of the imaging data provided by the doctor, the doctor and patient may verify the data using the query procedure shown in Figure 8.

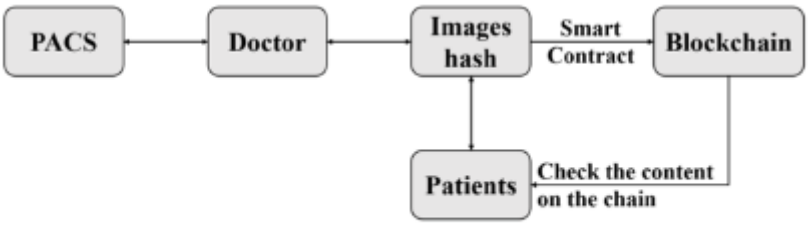

Fig. 8. Image data query/check process.

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.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

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

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

I. Initial Research We entered the console mode to execute the simulated transactions once the deployment was complete by typing >truffle console. We specified the variables to install all accounts: >let accounts so that we may switch accounts at any moment and start transactions. Afterwards, in console mode, we used the built-in web3.js function library, a library often utilized by DApps for account queries and operations like >web3.eth.getAccounts(function(err,res) { accounts = res; }). We made it easy on ourselves by assigning the number of defined edges to the appropriate 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)
```

Setting Permissions (B.) Dr. Lee was granted the ability to establish permissions. The doctor in the UserType enumeration is referenced by the "1" in the command line: > xry.setPermission("Dr. Lee", "doctor", 1, true) Part C: Data Entry The data for Ken's inspection images is entered into Dr. Lee's account when the necessary permissions have been established. For this investigation, we used the chest X-ray pictures as a model. Below you will find the directions for configuring the patient's picture data. The command line takes two arguments: the image hash value and the location to be inspected.

```
    >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.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

```
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

Medical professionals lose their authority when they quit. After losing access, the doctor will no longer be able to make changes to the patient's medical records. The code snippet is this: >xry.removePermission(doctor). The removal of permission can be checked with the command provided. The doctor is unable to alter the image after entering the following command: > xry.setXrayhash('chest', '16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d7400 86722d4bad0134', { from: doctor }). Users engage with the system using the aforementioned command procedures. One use case is data querying by both the patient and the physician. The patient's medical records may be double-checked once the command is typed by comparing the findings with the EMR system. Section F: Analyzing Performance Here we show the development cost, examine and compare our system's storage performance and security, and draw some conclusions. For the time being, small and medium-sized healthcare facilities are the best fit for the system that this research suggests. The experimental setup used in this work may be seen 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

Improving the data security of the existing PACS is the basis of the proposed architecture. The application scenarios are shown in small and medium-sized medical facilities to help save time and money. No new hires or replacements are required, hence there is no rise in personnel expenditures. extra expenditure in the system comes from the time and virtual currency needed to build the private chain. It is advised to use the Ethereum private chain for setup. Once setup is finished, mining Ethereum requires the Genesis block. The strategy takes into account the price of the mining equipment. Lastly, the private chain may trade and store data routinely with regular node mining and no special abilities or maintenance required for the follow-up operational expenses. 2) Analysis of Time For varying picture densities, the time required to convert hashes and upload them to the blockchain is shown in Figure 12. It takes a long time for uploads to the blockchain since nodes have to validate each other. In this experiment, a network with ten nodes served as the test bed. Consider a scenario where 100 photos were uploaded in 56830 ms and hashed in 685 ms.

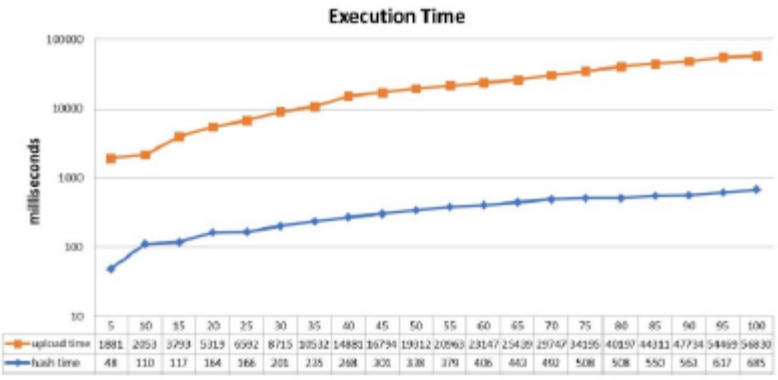

Fig. 12. Hash time & upload time for different numbers of images.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

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

In order to fortify safety, the suggested solution relies on the PACS. Performance in data transfer is quite similar. When it comes to protecting sensitive information, we use the immutable nature of the blockchain. As a result, safety is better than with the present PACS and the conventional X-Ray film operating mode. We compared the three systems' dependability and security, as indicated in Table III. Modifying data without the agreement of relevant units, such as attending physicians and patients, is the first challenge associated with tampering. The transfer of patient image data is straightforward when using conventional X-rays. In addition, patient data might be lost or corrupted in PACS systems due to hacking attempts on the main server. In order to make it more difficult to tamper with patient data, this system uses the decentralized storage of the blockchain and needs authentication of more than half of the total number of nodes. This strengthens the original PACS. When a single computer node fails or there is human error, the system operation procedure determines the entire system dependability. The lower the impact on system functioning, the higher the dependability. The level of security that prevents unauthorized access to and use of data stored in a database is known as data confidentiality. Data theft and unauthorized access are reduced in proportion to the level of secrecy. Last but not least, in a distributed denial of service (DDoS) assault, the perpetrator utilizes a network of controlled devices to flood the target server with packet requests until it crashes.

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 present a new use case for blockchain technology that integrates the immutable nature of the technology with an already popular electronic medical record system in order to validate patient picture data and prevent its manipulation. Authorization verification is the most crucial step in the verification process. We implement a solution for user authentication using Ethereum's smart contract technology, and we finish configuring and querying the private test blockchain using a mix of Truffle Suite and MetaMask. In order to make the front-end user interface more user-friendly, the function is still in the process of developing it to mimic medical records and remove particular rights. Protecting medical photographs in a way that complies with the provisions of the Personal Data Protection Act is essential for information security and privacy. Our main concern is the safekeeping of data pertaining to medical images. In the future, blockchain technology will have various uses in the medical field.

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.