IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of
Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

A DECENTRALIZED, SECURE BLOCKCHAIN FOR MEDICAL IMAGING DATA

[1]PONNALA SRIVIDYA, [2]KOVILAPU BHARAT KUMAR, [3]O VENKATA DURGESH NIKHIL, [4]GALIPALLI SANTHOSH PRASANN, [5]K GOWTHAM CHOUDARY, [6]Mrs.M.ANITHA RANI, [7]Mrs. G ANITHA,

[12345]Student, Department of DS, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[6]Assistant Professor, Department of CSE, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[7]Assistant Professor, Department of Mechanical Engineering, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

Abstract—

Medical applications including X-ray, ultrasound, computer tomography scans, and so on produce a massive amount of patient pictures daily. Patient privacy and individual rights are at the heart of medical data. As a result, protecting the confidentiality of medical pictures is crucial. Having a secure system in place to preserve medical photographs is essential for meeting the requirements of the Personal Data Protection Act. So, we came up with a blockchain-based solution for safely storing chest X-ray pictures uploaded to Kaggle. To validate medical photos and manage role-based access rights, we built a smart contract. After every medical inspection, we preserved the picture fingerprint in the blockchain by performing a cryptographic operation on the X-Ray image. In order to detect illnesses associated with pneumonia, artificial intelligence was used. In this study, we evaluated the time needed for X-ray retrieval in comparison to more traditional PACS systems. The testing findings demonstrated that the suggested blockchain architecture had a very low overhead of around 5%. "Keywords": blockchain technology, smart contracts, Dapp, and medical picture data.

INTRODUCTION

Applications in the medical field, such as ultrasound, CT scans, X-ray, and others, produce a flood 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. Notwithstanding, the centralized Electronic Medical Record (EMR) remains the primary repository for medical imaging data. Safeguarding medical pictures is essential for meeting the requirements of the Personal Data Protection Act. Using blockchain technology, we presented a safe approach for storing medical picture data in this research. The blockchain is used as both an encrypted data storage system and a safe method of recording transactions. [2] Other nodes begin verifying when one node modifies material. This modification takes effect when the number of verifications exceeds 51% of all nodes [3]. So, a strong encryption method ensures that no one can alter the data on the blockchain. There is a high level of protection for the original data since every change is subject to a stringent verification process. In this research, we used Ethereum's smart contract technology to put the suggested approach into action [4]. We built a mechanism to store medical images on the blockchain to ensure the safety of patient data. Part II: General Information Part A. Blockchain Technology Distributed ledger technology is known as blockchain. All data is disseminated when participants are linked via a peer-to-peer network. The common node and the billing node are the two main positions [4]. Bookkeeping nodes provide bookkeeping services and maintain ledgers, while ordinary nodes process transactions and other activities. Both parties may more effectively record the transaction on the distributed ledger linked via the blockchain, and the substance of the transaction can be permanently searched and confirmed. Bitcoin and other digital currency transactions that are part of blockchain 1.0 take place on the ever-changing blockchain ecosystem. "Smart contracts" are the center of Blockchain 2.0, a technology created by Ethereum [5]. Following the success of AI technology, Blockchain 3.0 is starting to take shape [6]. An Ethereum smart contract, a component of the blockchain 2.0, is the technological backbone of this study. You can find a quick description of blockchain's characteristics below. Database that is not centralized: Information is checked and exchanged between users. The nodes may directly converse, store, and send information with one another in a peer-to-peer transmission. Users are recognized by a code that is more than 30 characters long and they stay anonymous, making it both visible and anonymous. Permanently stored: once the transaction data is written, it cannot be altered in any way. Users have the option to personalize the logic algorithm so that it initiates node transactions automatically. Every block in a blockchain system is strongly connected to every other block. It is necessary to re-verify each block if its content changes; this also applies to newly-created blocks. In addition to the current block's label and the hash value of the preceding block, each block contains a wealth of other information.

data integrity is ensured by using the following: the current block hash value, the exhaustive guess value (nonce), the timestamp, the hash value of the transaction content, and the content of the transaction [7,8]. The content of each transaction is hashed twice in each block to ensure the security of the information. To ensure the safety of patient information, this study does a hash conversion on the picture data before uploading it to the blockchain. Additionally, each node competes to determine the exhaustive guess value (once) of the new block as part of the blockchain's verification procedure. 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.

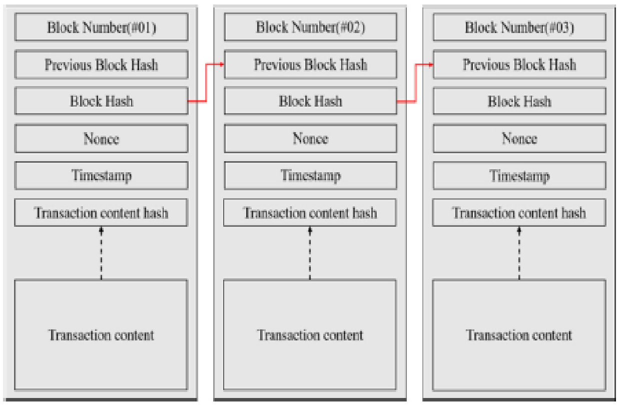


Fig. 1. Blockchain structure.

B. Intelligent Contract "A Next-Generation Smart Contract and DecentralizedApplication Platform" was the moniker given to the blockchain platform Ethereum in its 2015 white paper. Before Ethereum re-introduced and applied smart contracts to numerous fields—becoming the so-called "Blockchain 2.0"—they did not attract much attention, despite being presented by Nick Szabo in the early 1990s [9]. Among the many uses for smart contracts are in the realms of finance, meteorology, flight control, currency exchange, and proportional payment. Section C. EMR, or Electronic Health Record 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. Medical records may only be accessed by authorized individuals via the host computer. Figure 2 shows its operational diagram.

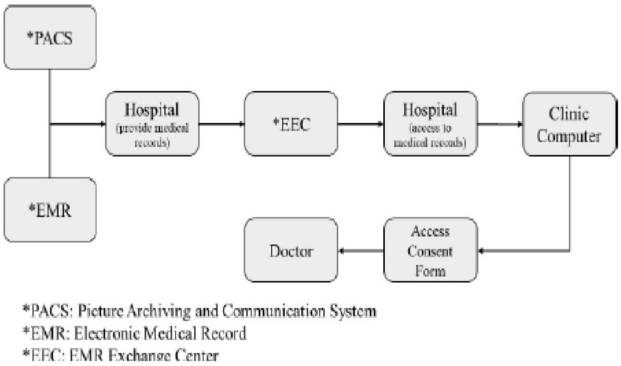

*PACS
*EMR
Hospital
(provide medical records)
*EEC
Hospital
(access to medical records)
Clinic Computer
Doctor
Access Consent Form
*PACS: Picture Archiving and Communication System
*EMR: Electronic Medical Record
*EEC: EMR Exchange Center

Fig. 2. Electronic medical record.

The PACS System for Image Archiving and Communication 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. Converting the picture to the Service-Object Pair (SOP) standard of Digital Imaging and Communications in Medicine (DICOM) allows for operations like storing, remembering, and transmitting, which are necessary for the image to work with the *EMR system. Figures 3 and 4, correspondingly, show them.*

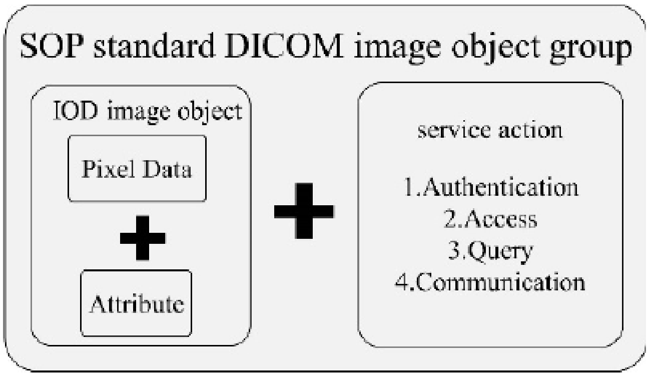


Fig. 3. SOP standard DICOM image object group.

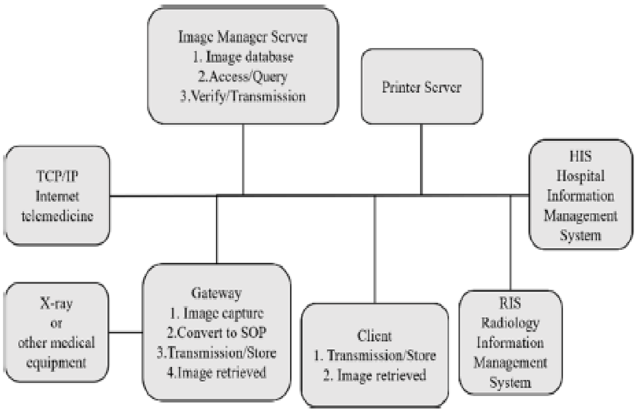
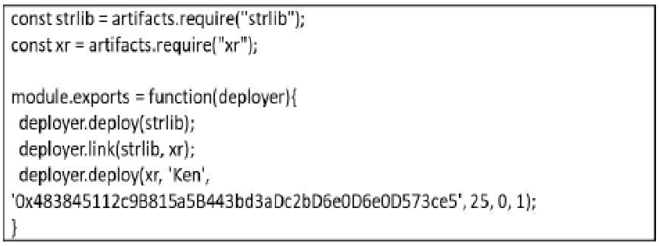


Fig. 4. Picture Archiving and Communication System (PACS).

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. A. Platform for System Development Firstly, Node.js The OpenJS Foundation is now responsible for maintaining node.js, an open-source execution environment that supports several platforms and allows server-side JavaScript execution. This framework is great for building web applications since it is both powerful and extensible. There are a number of noteworthy aspects about it. Each user accesses the server over the web, and low-power third-party modules are supported by high-performance "asynchronous I/O" that does not postpone waiting. 2) Truffles At the moment, the most popular framework for developing Ethereum is Truffle. To facilitate smart contract compilation, deployment, and testing, JavaScript is its programming language. Smart contracts are developed using either JS or Solidity for writing and automatically testing. When switching between the public and private chains for deployment, Truffle is in charge of the transition. To finish installing Truffle, just run the command "$ npm install truffle -g" from the Windows command line after the Node.js installation is complete. The following command is entered once the installation has finished. The following commands will establish a new folder named "Dapp_xray": "$ mkdirDapp_xray"; "$ cd Dapp_xray" will shift the absolute path to that folder; and "$ truffle init" will start a new Ethereum project. When you run the instructions above, you will see files in the Dapp Xray folder. Then, three Solidity files—xray.sol, xr.sol, and strlib.sol—are generated in the contracts. The X-Ray contract's parent contract, xray.sol, defines the contract's internal functions, state variables, enumeration, structure, modification word, and event. The X-Ray contract xr.sol provides a number of operations for managing the contract, including as Get, Set, and Remove. One such string library is strlib.sol. The necessary configuration parameters for deployment are established by executing 1_initial_migration.js in the migrations folder once the contract is written. In Figure 5, you can see the configuration. Following the completion of the settings, the following commands are run. The command "$ truffle compile" puts the smart contract together.



```
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.

The truffle-config.js file establishes the blockchain connection after compilation is finished. 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. At now, it can see the balance, status, address, key, and transactions of every node. Additionally, many mining solutions may have their log output from the internal blockchain configured and inspected at any time. Trading on Ganache's node accounts begins after reviewing the Truffle deployment contract instructions in the preceding section. B. Procedure for System Operation The system's architectural diagram is shown in Figure 6. The original PACS serves as the foundation for the front-end, which processes medical images; the back-end processes the programs by hashing them and then sends them on to the smart contract for role verification. The last step is to post the picture on the blockchain.

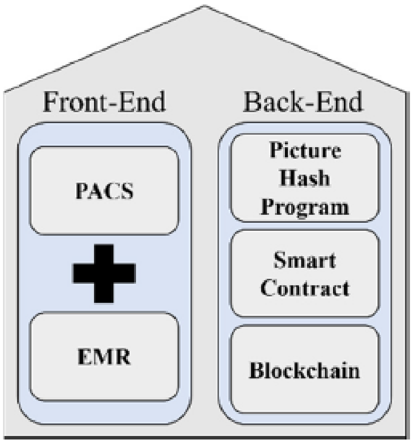


Fig. 6. System architecture.

Here, MetaMask is responsible for acquiring blockchain accounts and giving users a spot to log any data alteration. In order to verify permissions after registering MetaMask, the user must input the smart contract (Fig. 7). The doctor's account does not correspond to their position, even if the smart contract grants authorization verification access to all hospital physicians. It is instead considered a patient and granted access to their medical records.

International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume13, Issue 1,2025

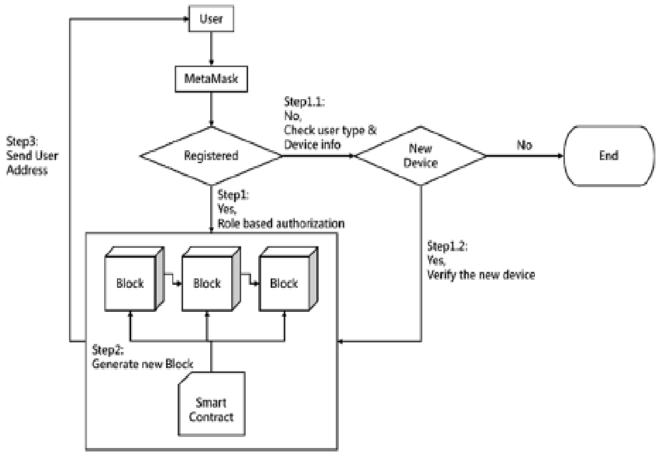
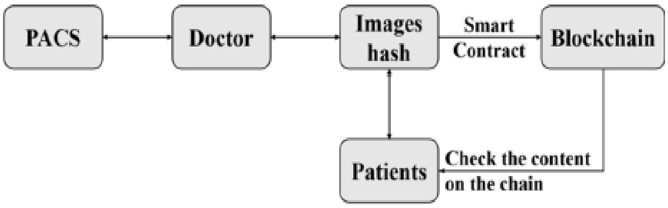


Fig. 7. Users registration process.

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. After patients get the hash value of their imaging data from their physicians, they may use the query method shown in Figure 8 to double-check the data with their doctors.



IMPLEMENTATION AND ANALYSIS

After following Truffle's instructions, the smart contract can be deployed and the necessary rights and data may be set up. Furthermore, prior to carrying out the pre-work, it is necessary to get the hash value of the patient's X-Ray picture. As seen in Figure 9, we conducted an experiment using the Chest X-Ray Images data set available on Kaggle. We developed a softwarein Python to add hash values to the system from images. You can see the outcomes of the procedure in Figure 10.

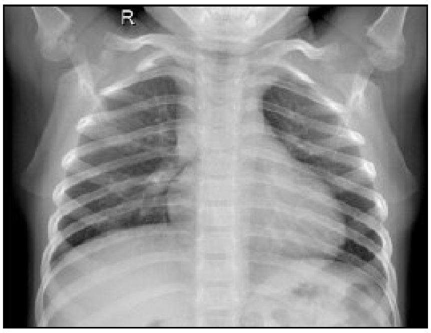
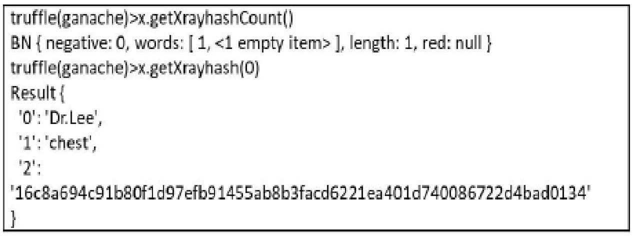


Fig. 9. Chest X-Ray images.

Section A. Initial Tasks To access the console mode and run the simulated transactions, we entered >truffle console once the deployment was finished. The variables to install all accounts were stated, allowing users to swap accounts and begin transactions at any moment. Next, we executed actions like >web3.eth.getAccounts(function(err,res) { accounts = res; } in the console mode using the built-in web3.js function library, a library often used by DApps for querying accounts. We made it easy by assigning each variable the number of stated edges: >let gov = accounts[0] >let host = accounts[1] >let doctor = accounts[2]. This allowed us to define each and every one of the variable accounts' addresses. At last, the smart contract was acquired and its instance, xr, was declared as xry. Then, the smart contract was deployed. then for each instance, set xry to instance Setting Permissions (B.) Dr. Lee's power was granted for the purpose of permission establishing. With >xry.setPermission(doctor, 'Dr.Lee,' 1, true), the "1" on the command line denotes the doctor in the UserType enumeration. Part C: Data Entry After the necessary permissions are established up, Ken's examination image data is put up using Dr. Lee's account. As a starting point for the investigation, we used the chest X-ray scans. 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’,
‘16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d740
86722d4bad0134’, { from: doctor })
```

Then, under the "personal information," use the following command lines to set up the "birthday" and "contact" fields using the host account. The first line is for the birthday and the second is for the contact information. In the first line, "19970522" is set as the birthdate and in the second line, "0912-321-456" is set as the contact information. D. Data Queries Following the completion of the aforementioned configuration, we requested information pertaining to the patient's images. Please see below the command line that was used to inquire the patient's basic information. The function xry.profile() You need to use the two command lines >xry.getXrayhashCount() >xry.getXrayhash(0) to get the array field before you can query the picture data. We ran the command to query the picture data after the previous two steps. Figure 11 displays the query command along with the result.



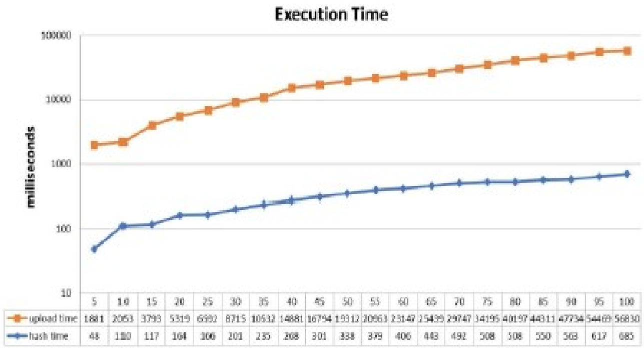
```
truffle(ganache)>x.getXrayhashCount()
BN { negative: 0, words: [ 1, <1 empty item> ], length: 1, red: null }
truffle(ganache)>x.getXrayhash(0)
Result {
  '0': 'Dr.Lee',
  '1': 'chest',
  '2':
'16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d740086722d4bad0134'
}
```

Fig. Eleven. Browse the patient's picture dossier. E. Removing Permissions Medical professionals lose their authority when they quit. The physician will no longer have the ability to make changes to the patient's medical records after they have lost this authorization. Here is the command: >xry.removePermission(doctor). To check whether the permission has been deleted, use the command above. The doctor will not be able to change the picture by running the following command: >xry.setXrayhash('chest', '16c8a694c91b80f1d97efb91455ab8b3facd6221ea401d7400 86722d4bad0134', { from: doctor }. The user communicates with the system by means of the functions listed above. One use case is data querying by both the patient and the physician. The findings are checked with the EMR system once the command is typed, allowing for the verification of the patient's medical data. Section F: Analyzing Performance Here, we show how much it cost to construct our system, as well as examine and evaluate its storage performance and security. At current time, smaller and medium-sized healthcare facilities are better suited to implement the method suggested in this research. 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

First, an examination of expenses To strengthen the data security of the current PACS, the suggested design builds on top of it. Here we outline the use cases in smaller and medium-sized healthcare facilities so that resources may be better used. Since it is compatible with the previous system, there's no need to hire more or replace existing staff, which means human expenses won't go up. Building the private chain takes time and virtual money, which is the system's most significant added expense. Additionally, the Ethereum private chain is suggested for setup, and after that, mining Ethereum requires the Genesis block. Cost of the mining equipment is taken into account by this procedure. Lastly, frequent node mining is all that's needed to keep the private chain running well and avoid any additional maintenance or specialized skills for the follow-up operational expenses, so data storage and trading should go smoothly. 2) Evaluation of Time For varying picture densities, the time required to convert hashes and upload them to the blockchain is shown in Figure 12. Uploading to the blockchain is a time-consuming process since nodes must validate each other. This experiment used a test environment consisting of ten nodes. It took 56830 milliseconds for the upload and 685 milliseconds for the hash with 100 photos, for instance.



	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

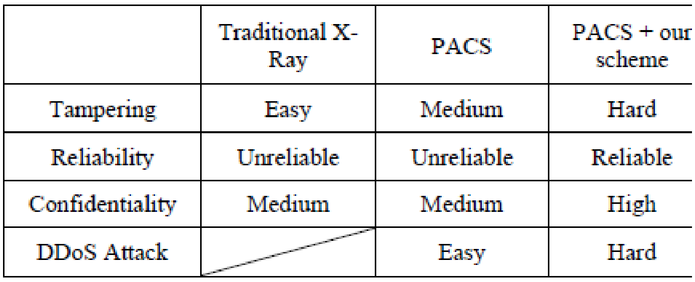
Figure 12: Hash and upload times for varying picture counts. Table II shows the results of our comparison of the three systems' X-Ray retrieval times. There is a 5% overhead with the suggested strategy ((4.1-3.9)/3.9=0.05).

TABLE II. SYSTEM COMPARISON

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

Analysis of Security (G.) The PACS forms the basis of the proposed system, which aims to enhance security. In terms of data transfer performance, there is little variation. We take use of the immutable nature of the blockchain to ensure the safety of your data. 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. It is simple to interchange patients' picture data while using standard X-Rays. In addition, patient data might be lost or corrupted in PACS systems due to hacking attempts on the main server. To further fortify the original PACS, this system employs the decentralized storage of the blockchain and mandates verification of over half of the entire number of nodes, making it very impossible to tamper with patient data. Moreover, in the event of human mistake or the failure of a single computer node within the system as a whole, the system's dependability is dependent on the system operating procedure. The less impact on system functioning there is when dependability is high. The level of protection against unauthorized access to data stored in a database is known as data confidentiality. The likelihood of data theft and unauthorized access decreases as the level of secrecy increases. Lastly, a distributed denial of service (DDoS) assault involves the use of several controlled sources to flood the target server with packet requests, rendering it impossible to load and ultimately leading to 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

CONCLUSION

In order to mitigate the possibility of tampering with patient picture data, we present a new blockchain application that integrates the robust security features of blockchain technology with an already established and widely used electronic medical record system. When it comes to verification, the permission verification is the most crucial step. Our solution to the user permission issue is based on Ethereum's smart contracts, and the private test blockchain's settings and queries are finished using a mix of Truffle Suite and MetaMask. The front-end user interface is being improved to better mimic medical records and remove special permissions. 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. When it comes to healthcare, blockchain technology has a lot of potential future uses.

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, PethuruRaj,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

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

[9]. 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.

[10]. 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.

[11]. 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.

[12]. 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.