A Blockchain-based Model for Securing IoT Transactions in a Healthcare Environment

Mohamed Abdel Kader Mohamed Elgendy[1], Mohamed Aborizka[2], Ali Mohamed Nabil Allam[3]
Computer Science Department[1, 2]
Information Systems Department[3]
Arab Academy for Science, Technology and Maritime Transport (AASTMT)
Cairo, Egypt

***Abstract*—A blockchain is a data structure that is implemented as a distrusted database or digital ledger. The transactions are saved to a block of transactions that is attached in turn to the blockchain after the verification process, in which each block in the chain contains a hash signature of the previous block in addition to the hash signature of the block itself. The blocks on the blockchain are chained as an immutable list using the proof-of-work procedure, where there is no way to alter or delete an attached block due to the strict security policy used for structuring the chain of blocks. Each node holds a copy of the blockchain in which the miners take the responsibility of verifying and attaching blocks to the blockchain. The Ethereum blockchain introduced the smart contract which holds logic to be processed once the contract is established. These smart contracts are developed via the Solidity programming language. This proposed paper exploits the Ethereum blockchain along with smart contracts as the base technology for implementing the proposed blockchain-based model. The paper aims to develop a multilayered blockchain-based model, in which the blockchain model is set up on a private blockchain Ethereum network where the nodes share the electronic medical records (EMR) among the P2P (peer-to-peer) network that will be used to secure the IoT medical transactions. Solidity smart contract, introduced by Ethereum, is deployed to handle the EMR "open-query-transfer" operations on the private network, whereas the miners are responsible to validate the transactions. Finally, the research conducts a performance analysis of the Ethereum network using the Ethereum Caliper, considering several performance factors, which are: Maximum Latency, Minimum Latency, Average Latency, and Throughput.**

***Keywords*—Blockchain; ethereum; electronic medical records (EMR); ioi secure transactions; smart contracts; proof-of-work**

I. Introduction

The blockchain technology has significantly contributed to putting an end to the interoperability challenges found in current and legacy healthcare IT systems, enabling individuals, healthcare service providers, healthcare entities, and medical institutions to securely share electronic healthcare sensitive data. Blockchains can enhance communication efficiency and increase security over the network, as the potential for using blockchain in healthcare is to overcome the challenges related to data security, privacy, sharing and storage [6], as well. It can also be applied in many software domains including financial and banking sectors, healthcare systems, and public services.

Although the blockchain-based models are increasingly used in modern software solutions, such models raised a significant number of challenges and objectives such as scalability, performance, processing speed over the network, data management on distributed nodes, and security breaches and attacks. Moreover, IoT network devices have been growing rapidly, as the number of installed IoT devices in the year 2022 is estimated to be 31 million devices. However, over the past few years, the blockchain model has become more stable and most of such issues and concerns have been resolved at most of the blockchain well-known platforms.

Thereby, as blockchains have become an excellent candidate to replace traditional transaction database systems, strict standards including acceptable behavioral guidelines must be laid out. These guidelines will facilitate the process of integrating the blockchain technology onto the healthcare domain systems, within the two blockchain main types: private and public blockchains. Thus, a blockchain network is basically either public or private, where a private blockchain is constructed for usage on a private network mainly used within a single entity such as a financial institution, for example. Generally speaking, blockchains are immutable, and thereby miners hold the responsibility of verifying the attached blocks to a blockchain on both private and public blockchains. But in the case of private networks, miners validate the blocks with a much stricter secured policy.

Miners create blocks of transactions, verify the blocks, and then attach every verified block to the blockchain. In blockchains, proof of stake or proof of work is being used to control the difficulty of the mining process. Thus, raising the complexity of the hash computations within the proof-of-work algorithm would increase the verification time of the newly attached block to the blockchain.

A block consists of five basic components: previous hash signature, nonce, transaction, timestamp, and the hash signature generated using proof of work or proof of stake. Data within the block attached to the blockchain is immutable; it is extremely difficult to be altered due to distributed nature of the blockchain structure. Furthermore, each block on the blockchain has a reference to the previous block hash signature, any change on the signature cannot be accomplished as in this case the hash has to be recalculated, and as a result, the blockchain will detect that change through the data verification process of the block.

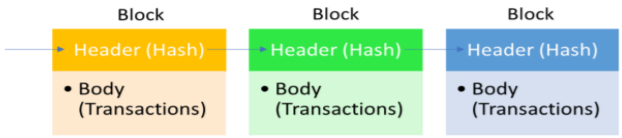


Fig. 1. Blockchain Architecture [16].

A block, which is chained onto a blockchain, holds transactions. As shown in Fig. 1, a block data structure consists of a block header and a block body. The block header is formed using multiple parameters which may vary based on the blockchain network provider. For the sake of integrity and verification, the header contains the parent hash signature which points to the previous block, in addition to the hash signature of the current block. The block body holds the block transactions. The block size is defined by the blockchain network provider, as this may vary based on the blockchain network provider.

The natural properties of the blockchain technology can be used to face the challenges mentioned previously, (1) permission-based blockchain networks to enable granular access control for medical records can be achieved by supporting granular-level access mechanisms; (2) blockchain-supported smart contracts enable patient-centric and transparent data sharing and control; (3) the blockchain distributed consensus mechanism overcomes the limitation of centralization; and (4) the immutable block preserves the integrity of data, which enables a blockchain to be verifiable and provable [1].

Accordingly, this paper presents a multilayered blockchain-based model for securing IoT transactions in healthcare traditional transaction systems to overcome the aforementioned issues concerning the dispersed and unified patients' medical records. Thus, the contribution of the proposed research is twofold:

1) Taking advantage of the blockchain technology as an immutable database for operating multilayered architecture flexible pattern, designed using clusters with embedded nodes to enforce flexible security level and approval permissions for medical records transactions on the whole blockchain.

2) Exploit the power of the Solidity programming language for developing highly secured smart contracts to handle the operation messages between the nodes from one side and the IoT devices and regular PC devices on the other side.

Therefore, the proposed multilayered blockchain model develops an approach for maintaining and managing patient medical records assembled in clusters with the usage of blockchain-based systems. The presented model uses aggregation (i.e., clusters) on the level of networks via (1) the Network ID, as each network represents a mining facility with its separate miners; (2) private blockchains on the cloud services provider "AWS" with the aim of creating the blockchains, the nodes, in addition to the miners; (3) Ethereum as an open-source blockchain-based distributed computing application platform. Additionally, the efficiency and performance of smart contracts are measured with the sophisticated certified tool "Caliper" which is used for measuring the core functions of the smart contract: Open, Query, and Transfer.

Experiments that are conducted to test the blockchain difficulty configuration, in addition to the number of total miners on the blockchain, shall prove the capability of the proposed model to work with high-security complexity, and with medium or low-security complexity, as well.

This paper is organized as follows: Section II will review the blockchain-based smart home architecture. Besides, it will compare the currently used dispersed healthcare models based on both, relational and NoSQL database architectures. Consequently, Section III will present two main contributions, first is designing the blockchain-based model used for securing and managing data immutable storage for healthcare multi-layered medical systems. Secondly, exploiting the smart contracts with solidity for managing the health organization transactions along with the verification of these transactions. Section IV deals with Implementation of model while Section V presents the experiments that demonstrate the effectiveness of the proposed architecture. At the end of the paper, Section VI will provide a discussion of open problems and will lay out a direction for the future work that could be done.

II. Related Work

Dorri, et al. [2] presented a blockchain-based smart home architecture, shown in Figure 2, which has been used as a core reference in the process of designing our proposed model. This BC-based smart home architecture consists of three tiers, the smart home, the overlay, and the cloud storage, in which communication within these tiers is carried out using block transactions. The smart home consists of IoT devices, local IL, and local storage as demonstrated in Figure 2; overlay is a P2P network with distributed capabilities in addition to cloud storage groups based on identical unique block numbers, where SHM has been used for authentication.

A. Blockchain Systems Versus Traditional Systems

Table I draws a comparison between the features of the blockchain-based systems and the remote patient monitoring system which depends on traditional communication and data storage methods, such as relational databases and cloud computing [3].

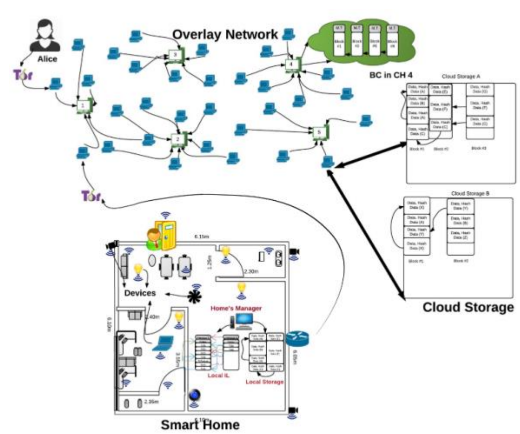


Fig. 2. BC-Based Smart Home.

TABLE I. COMPARISON BETWEEN TRADITIONAL AND BLOCKCHAIN-BASED SYSTEMS

Factor	Traditional Systems	Blockchain-based Systems
Confidentiality	Security level based on the configuration which may vary	High level of security
Availability	Must be manually configured	High service availability
Immutability	Data is exposed for manipulation	Immutable; the attached block cannot be altered or deleted
Traceability	Manually configured with a complex configuration	Traceable
Speed	Depends on network speed and hardware configuration along with the data source provider engine	May vary based on the blockchain verification process

A blockchain-based system runs on a P2P network of computers where each node on the network has an identical copy of the blockchain. Blockchains types can be classified as public, private, or hybrid blockchains.

1) Public blockchain: it was first implemented by Bitcoin and other cryptocurrencies, and it has contributed significantly to the distributed ledger technology (DLT) structure. Issues due to centralization are handled with DLT as it distributes data throughout a P2P network rather than storing it in a single location. Because of its decentralized nature, it forces methods of authentication.

2) Private blockchain: it is set up on a closed private network or controlled by a single entity. Functionality goes on the same basis regarding connectivity and decentralization; however, is substantially smaller.

3) Hybrid blockchain: it includes private and public blockchain characteristics. It allows the creation of a private permission-based system along with a public permissionless system, in addition to regulations for access to specific data on the blockchain [4].

B. Smart Contracts using Solidity

A smart contract can be defined as a piece of code that lives on a blockchain and is then executed automatically when one or more conditions are met. In the case of the Ethereum blockchain, smart contracts are implemented via the “Solidity” object-oriented language, in which users can execute the smart contracts through an application binary interface [6]. This property enables entities to perform their job functionalities such as access management, request handling, and data transmission. Ethereum enhanced the communication between a patient and a physician, as sharing medical prescriptions with the patients became much faster and easier. Accordingly, patients share their historical treatment data with doctors in a fast and accurate manner [5].

C. Comparison of the Data Management Mechanisms

Traditional legacy medical records are paper-based medical records (PMR), making it very difficult to keep track of a patient’s health history. Thus, saving historical data in such a way will cause data loss in addition to increasing the potential of inaccurate historical data, which potentially may lead to maltreatment. This serious issue has been faced by utilizing electronic medical records (EMR), the digital transformation of paper-based medical records. Electronic access to historical health records significantly improved the quality of treatment in addition to better disease diagnosis and preventive care [6]. Thereby, blockchain-based systems played an important role in modern healthcare solutions. Table II reviews and compares these main blockchain-based research exploited in the healthcare sector.

TABLE II. REVIEW OF THE BLOCKCHAIN-BASED RESEARCH IN THE HEALTHCARE SECTOR

Research	Blockchain Characteristics	Type(s) of Data	Merits
Castaldo & Cinque [7]	A private blockchain that does not rely on proof-of-work	EMR	Sharing E-health data across the EU via audit logging
Yue, *et al.* [8]	Private blockchain	EMR & PMR	Smart App to manage and share healthcare data
Patel [9]	A private blockchain that guarantees proof-of-stake	Medical Image Records	Securely sharing medical images
Fan, *et al.* [10]	Hybrid consensus mechanism based on practical byzantine fault tolerance (PBFT)	EMR	Secure sharing of healthcare data
Ji, *et al.* [11]	Proof-of-work	Patients’ Locations	Multilayer location sharing schema
Azaria, *et al.* [12]	Ethereum blockchain with proof-of-work	EMR	EMR management and sharing of healthcare data
Zhu, *et al.* [13]	Ethereum platform	EMR	Data management in the cloud environment
Genestier, *et al.* [14]	Hyperledger platform	Medical Records	Managing personal data in the e-health environment
Wang & Song [15]	Consortium blockchain	Medical Records	Coupling encryption and signature for robust security

III. RESEARCH METHODOLOGY

In this section, we propose and develop an approach for maintaining and managing patient medical records assembled in clusters with the usage of blockchain-based systems. Basically, the proposed model uses clusters on the level of networks using the network ID, as each network represents a mining facility with its separate miners. The proposed approach exploits the cloud services provider “AWS” to create the blockchains, the nodes, and the miners. Also, the proposed model uses “Ethereum” as an underlying base technology for managing the blockchain operations; the Ethereum platform is widely used as an open-source blockchain-based distributed computing application utility.

Moreover, the proposed model uses smart contracts developed with the "Solidity" programming language, which will be first deployed with the address to the blockchain, and then executed using its current hexadecimal address (Open-Query-Transfer).

It is worth mentioning that "Solidity" exploits the hashing algorithm KECCAK-256, as an alternative to the NIST standardized SHA-3 hash function, to verify the chained blocks (i.e., proof-of-work). The algorithm is defined as: (m,n) = POW(H~~n~~, Hn, d) where m is the mixHash, n is the nonce, H~~n~~ is the new block's header, Hn is the nonce of the block header, and d is the DAG (is a large dataset). The mixHash is a hash that, when combined with the nonce, proves that this block has carried out enough computation.

Thus, the proof of work (POW) controls the level of difficulty of attaching a block to the blockchain, and as a result, increasing the difficulty level will make the process of formulating the hash which matches the target hash more complex and will eventually consume more time.

A. Proposed Model

This research introduces a novel blockchain-based model for securing IoT transactions in the healthcare environment; the model that simulates the blockchain workflow on healthcare-based systems contains the following main components:

1) Hospital / Clinic Miners: miners are responsible for creating block transactions and attaching them to the blockchain.

2) Transactions: each transaction in the blockchain holds internal logic pertaining to the relevant smart contract. Once the transaction processing starts, the smart contract gets executed and the final output is conducted. There are three types of transactions within the proposed model: the "Open", "Query", and "Transfer" transactions, each of which is responsible for executing operations on the patient EMR file.

3) Local Blockchain: in each healthcare entity (hospital or clinic), a local private blockchain holds the blocks with its transactions, where each block has its own signature in addition to the previous block hash signature. The very first block in the blockchain is called the "genesis block"; it contains the setup configuration which controls the behavior of the blockchain in addition to controlling the security measures. The genesis block parameters will affect the process of attaching the new block of transactions, as the hashing algorithm will get harder based on the genesis block parameters, and thereby such configuration will directly affect the performance of the transactions' processing.

4) Global Blockchain: as the healthcare sector usually consists of more than one medical entity, peering between these entities is established to sync the mining operations on different entities, in which such a peering process shall establish the global blockchain scope. The peering process is achieved via the "addpeer" request, and once the two blockchains are peered, the mining process is synced between the two blockchains, as demonstrated in Fig. 3. Upon successful peering request process, the mining operation will be the responsibility of more than one miner, as each blockchain has one miner, each of which works at the same difficulty level value of the two blockchains.

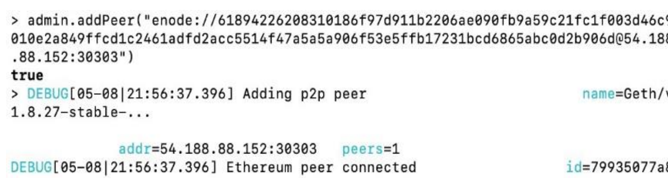
5) Overlay Network: it enables a distributed functionality on the proposed model architecture. Clusters, which are a group of network nodes, are used to decrease network overhead.

B. Proposed Model Design

Figure 4 illustrates the proposed model which clusters the network using the Network ID that represents a mining facility with its separate miners. The model also consists of private blockchains on the cloud services provider "Amazon Web Services (AWS)" for the sake of creating the blockchains and the nodes in addition to the miners, utilizing "Ethereum" as an open-source blockchain-based distributed computing application platform.

C. Proposed Model Transaction Structure and Flow

On each entity, the private Ethereum blockchain is set up with the predefined setup configuration found on the Genesis block, where the mining process on the blockchain is controlled according to the setup configuration parameters on the genesis block. In each mining operation, a new block is created, and the transactions are included within the newly created block, and at the end, the block is attached and added to the chain.



```
> admin.addPeer("enode://61894226208310186f97d911b2206ae090fb9a59c21fc1f003d46c9
010e2a849ffcd1c2461adfd2acc5514f47a5a5a906f53e5ffb17231bcd6865abc0d2b906d@54.188
.88.152:30303")
true
> DEBUG[05-08|21:56:37.396] Adding p2p peer                          name=Geth/v
1.8.27-stable-...

           addr=54.188.88.152:30303    peers=1
DEBUG[05-08|21:56:37.396] Ethereum peer connected                  id=79935077a8
```

Fig. 3. Ethereum Blockchain Peering.

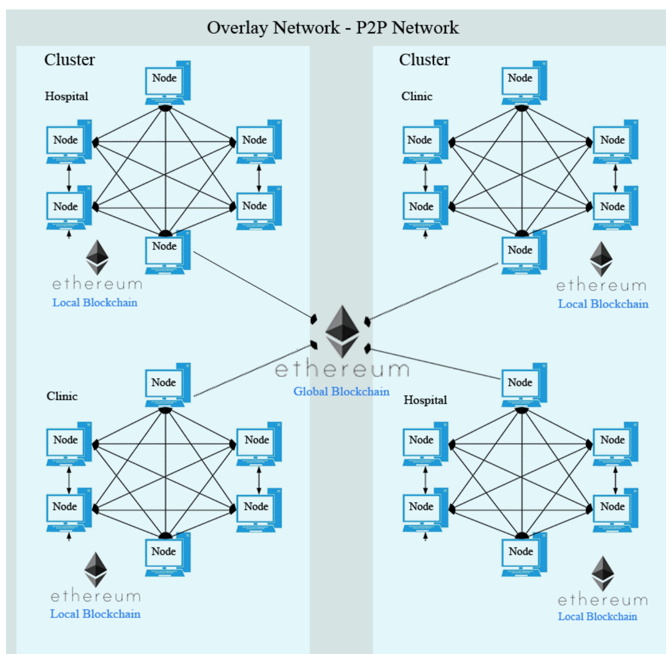


Fig. 4. Proposed Model Design.

Three main types of transactions are configured in the proposed model, namely, the “Open”, “Query”, and “Transfer” transactions, where each of them is executed via the smart contracts that contain the designated logic to be executed.

IV. Implementation

Fig. 5 is a flowchart that illustrates the steps of the addition and validation cycle in the proposed system.

A. Blockchain Initialization: The Genesis Block

The Genesis block contains the configuration of the private Ethereum network which will be used by all the miners, clusters, and nodes. The ChainId is used as the cluster identifier, where other configuration data inside the Genesis block will contain the following parameters/attributes:

- nonce: 64-bit string hash which, along with the mixHash, controls the amount of computation made for attaching the block to the blockchain.
- config: optional attribute which contains the ChainId unique identification of the private network; EIP150Block is used for fast sync, EIP155Block is used to reduce the probability of replay attacks, and EIP158Block controls how Ethereum clients handle empty accounts.
- timestamp: mainly used for verifying the order of the block within the blockchain.
- parentHash: a KECCAK 256-bit hash that points to the parent block.
- gasLimit: a scalar value that represents the limit of gas expenses of a single block in the blockchain.
- extraData: an optional parameter of 32 bytes at most, used for saving additional information if any.
- mixHash: a 256-bit hash which, together with the nonce, controls the level of computations used for verifying and attaching the block to the blockchain.
- coinbase: a 160-bit address, which is also called “etherbase”, holds all the successful mining operations amount.
- difficulty: a hexadecimal value that defines how hard it is to mine a block; the higher the value, the slower the mining process, since the mining operation will require more complex computations. Based on such difficulty value, hash computation is expected to run before obtaining a successful mining operation.
- alloc: optional parameters for predefining start balance on the mining account.

A typical example of the Genesis block structure:

```
{
  "timestamp": "0x5ca916c6",
  "nonce": "0x0000000000000042",
  "gasLimit": "0x2fefd8",
  "difficulty": "0x200",
  "mixHash":
"0x000000000000000000000000000000000000000000000000
0000000000000000",
  "coinbase":
"0xe1a7bcd0261e667651dc0e245d7b96e63c293d03",
  "number": "0x0",
  "config": {
    "chainId": 80,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0 },
  "gasUsed": "0x0",
  "parentHash":
"0x000000000000000000000000000000000000000000000000
0000000000000000",
  "alloc": {
    "0xe1a7bcd0261e667651dc0e245d7b96e63c293d03":
{
      "balance":
"0x200000000000000000000000000000000000000000000000
000000000000000"
} } }
```

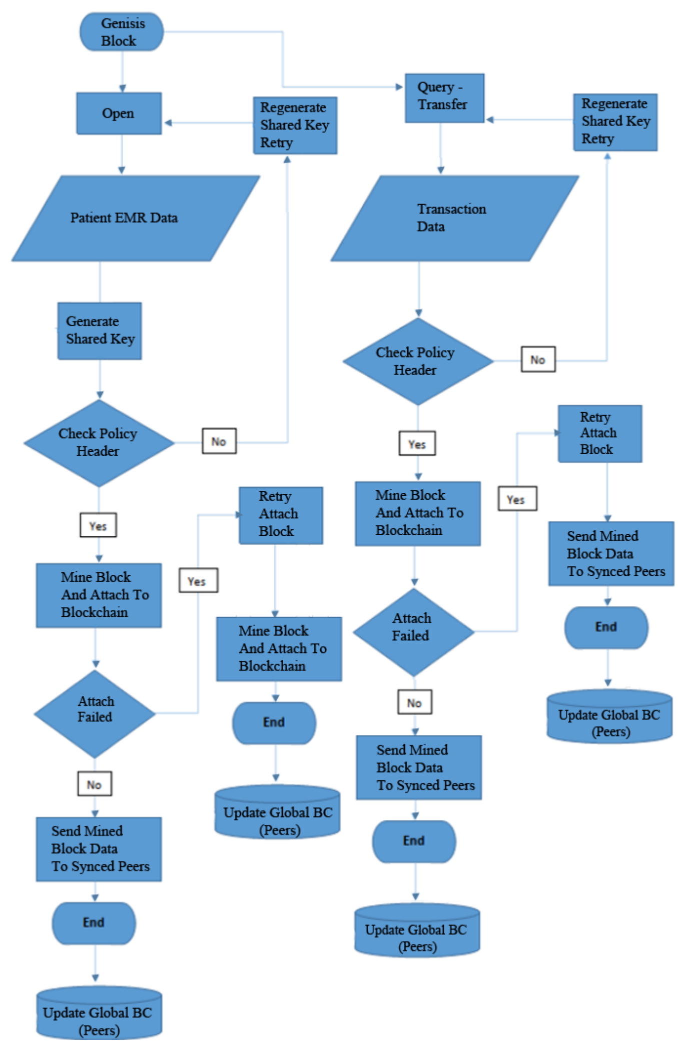


Fig. 5. Process Flow of the Proposed Model.

B. Smart Contract Implementation

A patient's account (i.e., medical record) has three main methods to be implemented:

1) Open: adding or creating a new patient's account (record or file).

2) Query: querying existing patient's account.

3) Transfer: transfer gas to another patient's account. Gas balance is essential for other operations like using contracts for updating a patient's record/file.

```
pragma solidity >=0.4.22 <0.6.0;
contract patientsol
{
   mapping(string => int) private accounts;

   function open(string memory acc_id, int amount) public
   {
      accounts[acc_id] = amount;
   }

   function query(string memory acc_id) public view returns (int amount)
   {
      amount = accounts[acc_id];
   }

   function transfer(string memory acc_from, string memory acc_to, int amount) public
   {
      accounts[acc_from] -= amount;
      accounts[acc_to] += amount;
   }
}
```

V. Experimental Environment and System Performance

The testing environment was configured on a VPC network (Virtual Private Cloud) via the AWS Cloud Server using an EC2 64-bit Ubuntu Server with a 2-core processor, a memory of 4GB, and an SSD hard drive of 8GB. The security group settings, which are linked to the EC2 instance, are set to allow a secure shell connection (SSH) via the secure 22 port; another port for setting up the blockchain is set to be 30301.

Typically, the Genesis block configuration parameters are initially set up to create the first initial block in the blockchain private network. Consequently, the blockchain graph construction process command starts, and then the mining process starts as the next step via the "mine" command. At that point, the blockchain private network has been constructed and is then ready for any transactions or performance testing operations.

"Hyperledger Caliper" is a benchmark tool that measures the performance of the blockchain operations using a predefined set of use cases. After processing the test cycles, the Caliper tool generates a detailed report that shows the performance factors, such as TPS (transactions per second), transaction latency, resource utilization, etc. The experiments conducted on the private blockchain network consisted of three tests for each individual operation:

1) Open → Testing the account creation process using the deployed smart contract (create a new patient's file).

2) Query → Testing the query operation on the created account using the deployed smart contract (query patient's file).

3) Transfer → Testing the transfer gas balance operation on the created accounts using the deployed smart contract. As mentioned before, gas balance is essential for other operations like using contracts to update a patient's file).

Four consecutive experiments are conducted iteratively to test the proposed system, where the benchmark input parameters at each experiment are set and configured to certain values. These configured input parameters are the nonce, mixHash, and difficulty, which are used to create the Genesis block on the blockchain. These parameters shall control the proof of work complexity, which will directly affect the process of attaching a new block to the chain. Thereby, the benchmark outputs the following performance indicators:

- Send Rate of a transaction per second.
- Maximum Latency (in seconds).
- Minimum Latency (in seconds).
- Average Latency (in seconds).
- Throughput of a transaction per second.

At each experiment, the total number of submitted transactions on the "*Open*" operation is 1000 with a rate of 50 transactions per second, whereas the "*Query*" operation was tested with a total of 1000 submitted transactions with a rate of 100 transactions per second, and finally the "*Transfer*" operation was initialized with 50 submitted transactions with a rate of 100 transactions per second.

The following subsections show the performance results of each of the four experiments, which are demonstrated in the four tables (from Table III to Table IV), and graphically illustrated in their corresponding charts (from Figure 6 to Figure 13).

A. Results of Experiment (1)

Genesis block initial configuration:

nonce → "0x000000000000052"
difficulty → "0x080"
mixHash →
"0x0000000000000000000000000000000000000000000000000000000000000000"

TABLE III. PERFORMANCE INDICATORS IN EXPERIMENT (1)

Operation	Number of Successes	Number of Failures	Send Rate (TPS)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	Throughput (TPS)
Open	1000	0	51.3	2.59	0.06	1.07	47.9
Query	1000	0	100.1	0.02	0.00	0.00	100.1
Transfer	50	0	5.5	2.64	0.08	0.96	5.2

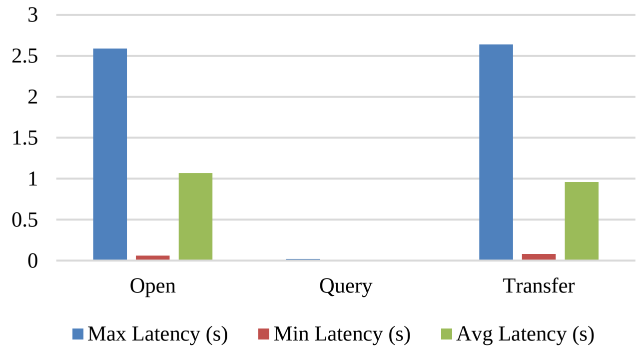


Fig. 6. Operations Latencies in Experiment (1).

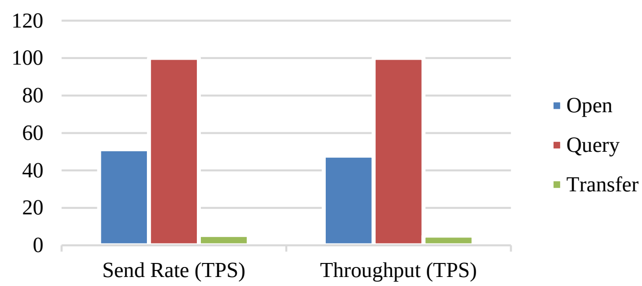


Fig. 7. Transactions Send Rate and Throughput in Experiment (1).

B. Results of Experiment (2)

The initial parameters of the genesis block are changed and updated in experiment (2). The blockchain has been reconfigured, and the DB has been reset to start the blockchain with the newly updated parameters, as follows:

Genesis block updated configuration

nonce → "0x0000000000000062"
difficulty → "0x080"
mixHash →
"0x0000000000000000000000000000000000000000000000000000000000000000"

TABLE IV. PERFORMANCE INDICATORS IN EXPERIMENT (2)

Operation	Number of Successes	Number of Failures	Send Rate (TPS)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	Throughput (TPS)
Open	1000	0	54.6	2.99	0.06	1.22	52.9
Query	1000	0	100.2	0.01	0.00	0.00	100.2
Transfer	50	0	5.3	3.65	0.08	1.59	4.2

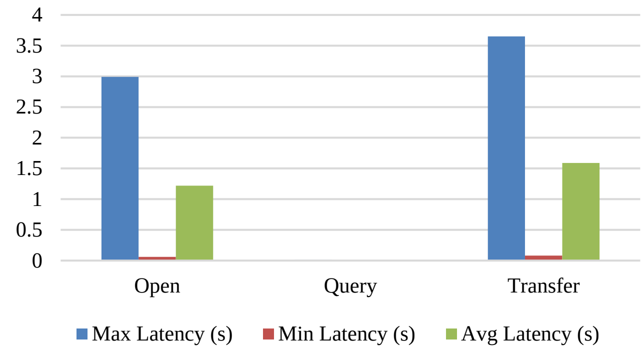


Fig. 8. Operations Latencies in Experiment (2).

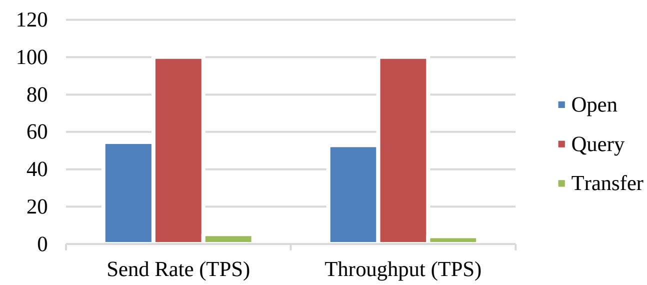


Fig. 9. Transactions Send Rate and Throughput in Experiment (2).

C. Results of Experiment (3)

Again, at an additional round, the parameters of the genesis block are modified. The blockchain has been reconfigured, and the DB has been reset to start the blockchain with the newly updated parameters, as follows:

Genesis block updated configuration

nonce → "0x0000000000000096"
difficulty → "0x00090000"
mixHash →
"0x0000000000000000000000000000000000000000000000000000000000000000"

TABLE V. PERFORMANCE INDICATORS IN EXPERIMENT (3)

Operation	Number of Successes	Number of Failures	Send Rate (TPS)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	Throughput (TPS)
Open	1000	0	61.1	6.73	0.37	3.10	47.5
Query	1000	0	100.2	0.02	0.00	0.00	100.2
Transfer	50	0	13.0	10.63	0.16	5.15	4.5

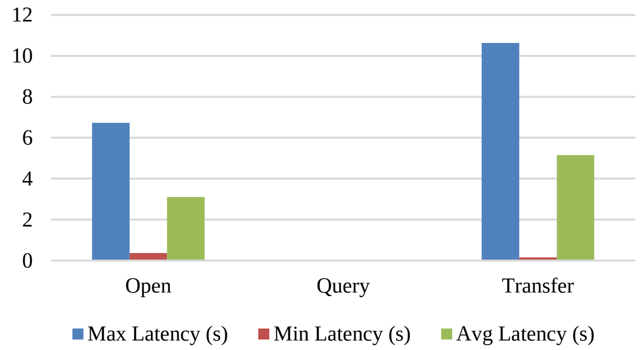


Fig. 10. Operations Latencies in Experiment (3).

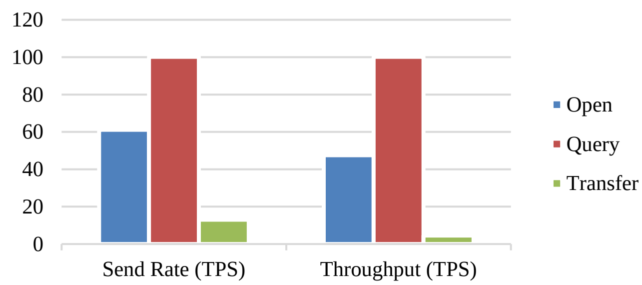


Fig. 11. Transactions Send Rate and Throughput in Experiment (3).

D. Results of Experiment (4)

Finally, in the last round of the experiments, the parameters of the genesis block are remodified. The blockchain has been reconfigured and the DB has been reset to start the blockchain with the new update of the parameters, as follows:

Genesis block updated configuration

nonce → "0x0000000000000196"
difficulty → "0x00099000"
mixHash →
"0x0000000000000000000000000000000000000000000000000000000000000000"

The experimental results of the research show the effect of increasing the security level of the blockchain network, using the input parameters that exist on the genesis block. Raising the values of these parameters has an obvious effect on the blockchain performance, as demonstrated by the increase in the latency values reported in the four experiments.

TABLE VI. PERFORMANCE INDICATORS IN EXPERIMENT (4)

Operation	Number of Successes	Number of Failures	Send Rate (TPS)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	Throughput (TPS)
Open	1000	0	79.5	7.44	0.37	3.48	53.8
Query	1000	0	100.1	0.01	0.00	0.00	100.1
Transfer	50	0	7.5	11.88	1.75	6.92	4.8

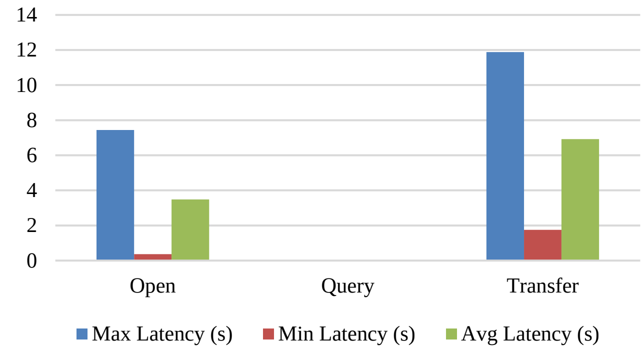


Fig. 12. Operations Latencies in Experiment (4).

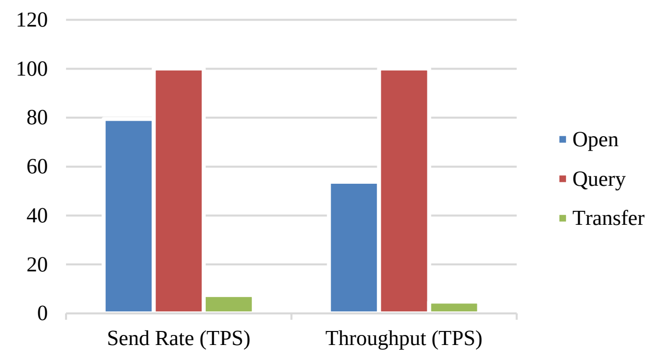


Fig. 13. Transactions Send Rate and Throughput in Experiment (4).

VI. CONCLUSION AND FUTURE WORK

In this paper, a multilayered blockchain-based model has been developed with the aim of securing medical systems and managing the dispersed electronic medical records in immutable unified blockchain storage, in a completely decentralized manner using blockchain technology. Smart contracts were implemented to achieve peer-to-peer network security using the Solidity programming language.

The performance analysis of the most popular Ethereum client, Geth, is carried out using the authorized performance benchmark framework "Ethereum Caliper", considering the Send Rate (TPS), Max Latency (s), Min Latency (s), Avg Latency (s), Throughput (TPS). The results show that the effect of raising the security level on the blockchain network using the input parameters will add heavy overhead on the blockchain network performance as demonstrated in the four performed experiments. Each experiment has been performed with a different genesis block setup to demonstrate the correlation between high security and low performance.

The future work shall focus on enhancing how IoT devices are connected to the blockchain network, in addition to enhancing how IoT devices communicate within different country-wide network clusters.

References

[1] M. K. Elghoul, S. F. Bahgat, A. S. Hussein and S. H. Hamad, "A Review of Leveraging Blockchain based Framework Landscape in Healthcare Systems," International Journal of Intelligent Computing and Information Sciences, vol. 21, no. 3, pp. 71-83, 2021.

[2] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, USA, 2017.

[3] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," Journal of Medical Systems, vol. 42, no. 7, pp. 1-7, 2018.

[4] A. Haleem, M. Javaid, R. P. Singh, R. Suman and S. Rab, "Blockchain Technology Applications in Healthcare: An Overview.," International Journal of Intelligent Networks, vol. 2, pp. 130-139, 2021.

[5] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," IEEE Access, vol. 7, pp. 66792-66806, 2019.

[6] S. Khezr, M. Moniruzzaman, A. Yassine and R. Benlamri, "Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research," Applied sciences, vol. 9, no. 9, 2019.

[7] L. Castaldo and V. Cinque, "Blockchain-based Logging for the Cross-border Exchange of eHealth Data in Europe," in International ISCIS Security Workshop, 2018.

[8] X. Yue, H. Wang, D. Jin, M. Li and W. Jiang, "Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control," Journal of Medical Systems, vol. 40, no. 10, pp. 1-8, 2016.

[9] V. Patel, "A Framework for Secure and Decentralized Sharing of Medical Imaging Data via Blockchain Consensus," Health Informatics Journal, vol. 25, no. 4, pp. 1398-1411, 2019.

[10] K. Fan, S. Wang, Y. Ren, H. Li and Y. Yang, "Medblock: Efficient and Secure Medical Data Sharing via Blockchain," Journal of Medical Systems, vol. 42, no. 8, pp. 1-11, 2018.

[11] Y. Ji, J. Zhang, J. Ma, C. Yang and X. Yao, "BMPLS: Blockchain-based Multi-level Privacy-preserving Location Sharing Scheme for Telecare Medical Information Systems," Journal of Medical Systems, vol. 42, no. 8, pp. 1-13, 2018.

[12] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "Medrec: Using Blockchain for Medical Data Access and Permission Management," in 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 2016.

[13] L. Zhu, Y. Wu, K. Gai and K.-K. R. Choo, "Controllable and Trustworthy Blockchain-based Cloud Data Management," Future Generation Computer Systems, vol. 91, pp. 527-535, 2019.

[14] P. Genestier, S. Zouarhi, P. Limeux, D. Excoffier, A. Prola, S. Sandon and J.-M. Temerson, "Blockchain for Consent Management in the ehealth Environment: A nugget for Privacy and Security Challenges," Journal of the International Society for Telemedicine and eHealth, vol. 5, pp. 1-4, 2017.

[15] H. Wang and Y. Song, "Secure Cloud-based EHR System using Attribute-based Cryptosystem and Blockchain," Journal of Medical Systems, vol. 42, no. 8, pp. 1-9, 2018.

[16] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari and Y. Cao, "A Survey on Blockchain Technology: Evolution, Architecture and Security," IEEE Access, vol. 9, pp. 61048-61073, 2021.