# Blockchain Technology in Electronic Health Records

COMPSCI 9644B Scalable and Robust Distributed Systems

Neha Mary Thomas
*University of Western Ontario*
London, Canada
nthoma54@uwo.ca

Pavani Madhur
*University of Western Ontario*
London, Canada
pmadhur@uwo.ca

***Abstract*— Blockchain has been gaining popularity in the recent years for its ability to maintain a secure, confidential and decentralized records of transaction. This technology has applications in different domains, a popular one being the healthcare. Healthcare professionals are required to maintain accurate records of patients electronically which cannot be tampered from an outside source. In this paper, we discuss how Blockchain technology can be implemented in maintaining records of the patient and checking if the records are tampered using consensus algorithm such as ProofOfWork (POW).**

## I. INTRODUCTION

During the early days, patient records were maintained in the form of papers which could be easily tampered with. This paper-based system was inefficient, insecure, unorganized and was not temper-proof [1]. It also faced the issue of data- duplication and redundancy as all the institutions that patient visited had various copies of patient's medical records [1]. The recent advancements in technology are making changes in our everyday life. These changes are also been shown in the healthcare sector. The main benefits with these advancements include enhancements in security, user experience. These benefits were offered by Electronic Health Record (EHR). There are still some issues that are not yet solved, one of it being the security of patient records. A solution to this problem is using Block Chain technology.

Blockchain has applications in many domains such as cryptocurrency systems, Banking and Finance, Smart Contracts, Supply Chain Management, HealthCare. Block Chain can be thought of as distributed database that is shared among the nodes of a computer network [2]. The basic unit of block chain is block, which stores all information together. The blocks are filled with information and are closed and linked to the previous block forming a chain called blockchain. Each block in the chain is given a timestamp of when it was added. Each block stores a hash of the current block and also the hash of the previous block. Hash codes are created by a mathematical function, if the information stored in the block is altered then the hash of the block would be changed. The first block of a blockchain is called as Genesis Block. This block does not have any previous hash value which is by default 0. Block Chain relies on a decentralized architecture which allow the multiple nodes to be involved in a system which is more secure and resistant to single point of failure. If an intruder manipulates a record, it would manipulate only the record of the current node and no other nodes would be changed thereby achieving decentralized security.

Some of the advantages of block chain are firstly transactions are secure, private. Secondly, the accuracy of blockchain is better with very less involvement from human for verification. Thirdly, decentralization makes blockchain very hard to tamper with.

The paper is organized as follows. In section II, we review the consensus algorithms. Section III describes the Architecture. Section IV presents the design challenges. Section V focus on the test scenarios while Section VI concludes the paper.

## II. CONSENSUS ALGORITHMS

Consensus algorithm or consensus protocol is a mechanism through which all nodes in a network reach a common agreement about the current state of the blockchain. Blockchain that are decentralized work on a global scale without any single authority. With the dynamically changing status of the blockchain, there needs to be mechanism to make sure all transactions in the network are genuine and all nodes agree on a consensus on the status of the blockchain.

There are many consensus algorithms that works on different principles. The common one being ProofOfWork (POW). The central idea of this algorithm is to find a node that solve a complex mathematical problem in very less time. There is a term called Difficulty which is used to determine how hard it is to find the hash required to mine a new block. For our project, we have set the difficulty to 2. The node that solves the puzzle first acts as a miner and adds the new block to the chain. It would then announce the addition of new block to the peers of the network thereby receiving a reward. The main advantages of this consensus algorithm are that it is very hard to find a solution for the complex mathematical problem thereby providing high levels of security

## III. Architecture

We have implemented Web based Electronic Health Record system using Flask, html, CSS and mongo dB to show the high level implementation of block chain on a decentralized platform. It facilitates the users to login as doctor, or patient and perform certain niche operations.

The information used to authenticate the user at login is collected and stored on the cloud server (MongoDB). A block is a structure consisting of headers, EHR transactions, and metadata. The header consists of the block's index, the current hash, and the previous hash. An EHR transaction is a patient record created by a certified physician. Each EHR transaction consists of a set of medical records and patient demographic attributes consisting of patient_id, doctor_id, first name, last name, and disease-related information. The metadata consists of the creation timestamp. In the blockchain, all transactions are logged, every node of the distributed network has a complete copy of the blockchain, and transactions are based on cryptographic principles such as the Proof of Work (PoW) consensus algorithm which is validated by the miner who maintains the transaction ledger. These principles also ensure that these nodes automatically and continuously agree on the current status of the ledger and all transactions within it. If someone tries to destroy the transaction, the node will not reach consensus and will refuse to consolidate the transaction on the blockchain.

We will be using a popular Python microframework called Flask to create a REST-API to interact and invoke various operations in our blockchain. We have created the following API endpoints.

*A.* /new_transaction - We need an endpoint for our application to submit a new transaction. This will be used by our application to add new data to the blockchain. Every EHR corresponds to a transaction, which will be added to the list of unconfirmed transactions and then invoke the mine() function, which is used to initiate a command to mine the transaction.

*B.* /chain - Endpoint to return the node's copy of the chain. Our application will be using this endpoint to query all of the data to display

*C.* /register_with - In order to achieve decentralization with a peer-to-peer network, multiple nodes needs to be added maintaining the same blockchain, we used the /register_with api to register new node with all the other peers in the network. A new node participating in the network can invoke the register_with_current_node method via /register_with API endpoint to register with existing nodes in the network which involves requesting the remote node to add a new node to its list of known global peers and also initialize the blockchain of the new node with the blockchain of the remote node and resync the blockchain to the network when the node goes offline.

*D.* However, there is a problem with multiple nodes. The copy of the chain of some nodes may be different for unintended reasons such as intentional operation or network delay. In this case, the nodes need to agree on a version of the chain to maintain system-wide integrity. In other words, we need to reach a "consensus". A simple and efficient consensus algorithm is to agree on the longest valid chain when the chains of different participating nodes in the network appear to be bifurcated. The rationale behind this approach is that the longest chain is a good estimate of the maximum amount of work.

*E.* /add_block - endpoint to add a block mined to the node's chain. The node first verifies the block then adds it to the chain. There should be a way for a node to notify the network of it peer nodes that it has mined a new block. Other nodes simply validate the proof of work and add mined blocks to their respective chains. The /announce_new_block method must be called by the remote node after it has mined a block, so that the peer can add the blocks to the chain.

We have implemented following features in our application

A. Anyone can enroll in the blockchain network as a doctor or patient. Whenever a patient visits a doctor, the doctor has the necessary authority to keep demographics and medical logs in the patient's records. These records are stored in a distributed ledger of transactions over the blockchain network.

B. Doctors must log in with an encrypted username and password to access the dashboard. The dashboard gives you the option to create and view patient records that are uniquely identified by the patient ID.

C. Patients would also require to sign in to access the dashboard and have an ability to view the records

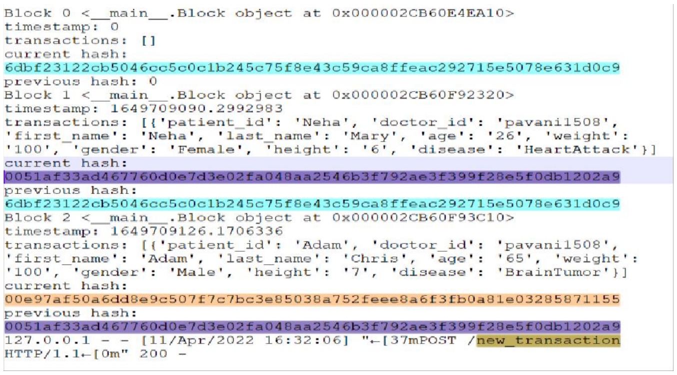
## IV. DESIGN CHALLENGES

One of the challenges we faced was which consensus algorithm should be chosen for HER management. We were confused with Proof of Work and Proof of Stake algorithm, although ProofOfWork is computationally expensive the main disadvantage of Proof of Stake is that since the validation power is given to the largest stakeholder there may arise a chance of a centralised system. Proof Of Work algorithm is very popular and is used by many cryptocurrency systems like Bitcoin.

Apart from that, we also faced trouble in determining whether to develop the front end for the Flask application using Jinja2 HTML template rendering engine or with another library/framework like React. Flask and React are an excellent match for single-page applications in which the application layers must be extremely responsive to one another. Flask's integration with React includes benefits such as high processing speed and responsiveness. Despite the benefits, considering the time constraints, we decided to employ Jinja 2 templates, as the major emphasis of the project to demonstrate how blockchain technology can be used to securely store data in the Health Care Domain.

## V. TEST SCENARIOS

In this section, we discuss about the scenarios we tested to see if the data is synced properly in the decentralized platform. One instance of the node is up and running on the port 5000. Now, Doctor is logged in and created two transactions which will trigger the /new_transaction endpoint to add each transaction to the new block and then mined to add the blocks to the blockchain.



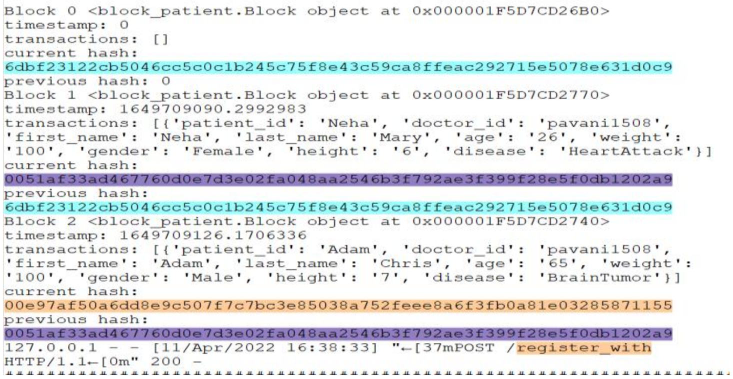
```
Block 0 <__main__.Block object at 0x000002CB60E4EA10>
timestamp: 0
transactions: []
current hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
previous hash: 0
Block 1 <__main__.Block object at 0x000002CB60F92320>
timestamp: 1649709090.2992983
transactions: [{'patient_id': 'Neha', 'doctor_id': 'pavani1508',
'first_name': 'Neha', 'last_name': 'Mary', 'age': '26', 'weight':
'100', 'gender': 'Female', 'height': '6', 'disease': 'HeartAttack'}]
current hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
previous hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
Block 2 <__main__.Block object at 0x000002CB60F93C10>
timestamp: 1649709126.1706336
transactions: [{'patient_id': 'Adam', 'doctor_id': 'pavani1508',
'first_name': 'Adam', 'last_name': 'Chris', 'age': '65', 'weight':
'100', 'gender': 'Male', 'height': '7', 'disease': 'BrainTumor'}]
current hash:
00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155
previous hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
127.0.0.1 - - [11/Apr/2022 16:32:06] "←[37mPOST /new_transaction
HTTP/1.1←[0m" 200 -
```

Figure 1. Block chain copy at Node1 after adding two transactions

Now we spinned another node running on the port 5001. Its time to register the node at port 5001 with the already running 5000 using the /register_with api endpoint. We used postman to send the post requests for the registration to happen. We can also use curl commands like below to register node at port 5001 to 5000.

```
curl -X POST \
http://127.0.0.1:5001/register_with \
-H 'Content-Type: application/json' \
-d '{"node_address": "http://127.0.0.1:5000"}'
```

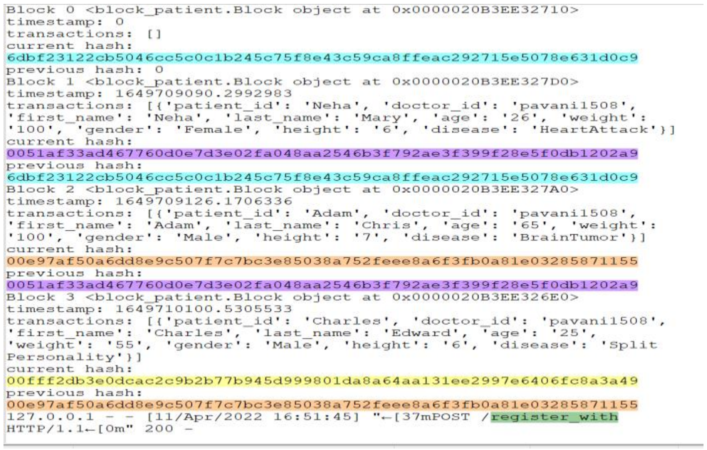
The newer nodes will also sync the chain with the existing node so that they are able to participate in the mining process actively.



```
Block 0 <block_patient.Block object at 0x000001F5D7CD26B0>
timestamp: 0
transactions: []
current hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
previous hash: 0
Block 1 <block_patient.Block object at 0x000001F5D7CD2770>
timestamp: 1649709090.2992983
transactions: [{'patient_id': 'Neha', 'doctor_id': 'pavani1508',
'first_name': 'Neha', 'last_name': 'Mary', 'age': '26', 'weight':
'100', 'gender': 'Female', 'height': '6', 'disease': 'HeartAttack'}]
current hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
previous hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
Block 2 <block_patient.Block object at 0x000001F5D7CD2740>
timestamp: 1649709126.1706336
transactions: [{'patient_id': 'Adam', 'doctor_id': 'pavani1508',
'first_name': 'Adam', 'last_name': 'Chris', 'age': '65', 'weight':
'100', 'gender': 'Male', 'height': '7', 'disease': 'BrainTumor'}]
current hash:
00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155
previous hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
127.0.0.1 - - [11/Apr/2022 16:38:33] "←[37mPOST /register_with
HTTP/1.1←[0m" 200 -
```

Figure 2. Blockchain copy of node 2 running on port 5001 which is created from the blockchain dump of node running on port 5000

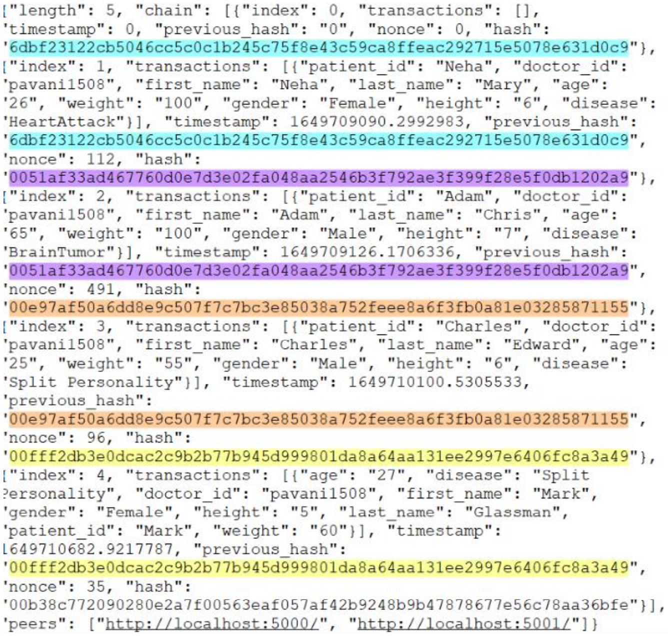
Now let's assume node1 is down and at that time a new record is being added at node2. Now when the node1 comes up, the resync of block chain will happen according to the consensus algorithm and the chain dump from node2 is copied to node1 during the registration process of node1 with node2.



```
Block 0 <block_patient.Block object at 0x0000020B3EE32710>
timestamp: 0
transactions: []
current hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
previous hash: 0
Block 1 <block_patient.Block object at 0x0000020B3EE327D0>
timestamp: 1649709090.2992983
transactions: [{'patient_id': 'Neha', 'doctor_id': 'pavani1508',
'first_name': 'Neha', 'last_name': 'Mary', 'age': '26', 'weight':
'100', 'gender': 'Female', 'height': '6', 'disease': 'HeartAttack'}]
current hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
previous hash:
6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9
Block 2 <block_patient.Block object at 0x0000020B3EE327A0>
timestamp: 1649709126.1706336
transactions: [{'patient_id': 'Adam', 'doctor_id': 'pavani1508',
'first_name': 'Adam', 'last_name': 'Chris', 'age': '65', 'weight':
'100', 'gender': 'Male', 'height': '7', 'disease': 'BrainTumor'}]
current hash:
00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155
previous hash:
0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9
Block 3 <block_patient.Block object at 0x0000020B3EE326E0>
timestamp: 1649710100.5305533
transactions: [{'patient_id': 'Charles', 'doctor_id': 'pavani1508',
'first_name': 'Charles', 'last_name': 'Edward', 'age': '25',
'weight': '55', 'gender': 'Male', 'height': '6', 'disease': 'Split
Personality'}]
current hash:
00fff2db3e0dcac2c9b2b77b945d999801da8a64aa131ee2997e6406fc8a3a49
previous hash:
00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155
127.0.0.1 - - [11/Apr/2022 16:51:45] "←[37mPOST /register_with
HTTP/1.1←[0m" 200 -
```

Figure 3. Blockchain copy of node 1 when it comes up

Now both the nodes are up and running, we will add the record at node1 and changes will be synced to node2 as well.



```
["length": 5, "chain": [{"index": 0, "transactions": [],
'timestamp": 0, "previous_hash": "0", "nonce": 0, "hash":
'6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9"},
["index": 1, "transactions": [{"patient_id": "Neha", "doctor_id":
'pavani1508", "first_name": "Neha", "last_name": "Mary", "age":
'26", "weight": "100", "gender": "Female", "height": "6", "disease":
'HeartAttack"}], "timestamp": 1649709090.2992983, "previous_hash":
'6dbf23122cb5046cc5c0c1b245c75f8e43c59ca8ffeac292715e5078e631d0c9",
'nonce": 112, "hash":
'0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9"},
["index": 2, "transactions": [{"patient_id": "Adam", "doctor_id":
'pavani1508", "first_name": "Adam", "last_name": "Chris", "age":
'65", "weight": "100", "gender": "Male", "height": "7", "disease":
'BrainTumor"}], "timestamp": 1649709126.1706336, "previous_hash":
'0051af33ad467760d0e7d3e02fa048aa2546b3f792ae3f399f28e5f0db1202a9",
'nonce": 491, "hash":
'00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155"},
["index": 3, "transactions": [{"patient_id": "Charles", "doctor_id":
'pavani1508", "first_name": "Charles", "last_name": "Edward", "age":
'25", "weight": "55", "gender": "Male", "height": "6", "disease":
'Split Personality"}], "timestamp": 1649710100.5305533,
'previous_hash":
'00e97af50a6dd8e9c507f7c7bc3e85038a752feee8a6f3fb0a81e03285871155",
'nonce": 96, "hash":
'00fff2db3e0dcac2c9b2b77b945d999801da8a64aa131ee2997e6406fc8a3a49"},
["index": 4, "transactions": [{"age": "27", "disease": "Split
?ersonality", "doctor_id": "pavani1508", "first_name": "Mark",
'gender": "Female", "height": "5", "last_name": "Glassman",
'patient_id": "Mark", "weight": "60"}], "timestamp":
l649710682.9217787, "previous_hash":
'00fff2db3e0dcac2c9b2b77b945d999801da8a64aa131ee2997e6406fc8a3a49",
'nonce": 35, "hash":
'00b38c772090280e2a7f00563eaf057af42b9248b9b47878677e56c78aa36bfe"}],
'peers": ["http://localhost:5000/", "http://localhost:5001/"]}
```

Figure 3. Response for the get request

http://localhost:5001/chain

## VI. CONCLUSION

In this paper, we have discussed about how blockchain technology can be useful in healthcare industry especially in the creation and maintenance of Electronic Health Records (EHR). Despite the advancement in technology, EHR faced some security breaches which could be solved using Block Chain technology. Our proposed model provides APIs to accommodate different requests to create EHR records by doctors while maintaining data integrity and also with the participation of multiple nodes in a network our framework it guarantees data availability.

## REFERENCES

[1] Using Blockchain for Electronic Health Record – Ayesha Shahnaz, Usman Qamar, Ayesha Khalid IEEE 2019 conference.

[2] Adam Hayes, "Blockchain Explained," para. 2, Mar. 05, 2022. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp [Accessed Apr. 11, 2022]

[3] Blockchain based electronic healthcare record system for healthcare 4.0 applications – Sudeep Tanwar, Karan Parekh, Richard Evans IEEE 2019 conference.