Role of Blockchain to Improve Remote Healthcare System

Adarsha Bhattarai
Department of *Electrical and Computer Engineering*
University of Nebraska-Lincoln
Omaha, USA
abhattarai3@huskers.unl.edu

***Abstract*—Remote healthcare system forms a distributed network that consists of multiple processing layers. These layers might include a local sensor, an edge computer, a fog server, and a cloud server. A robust security system is a necessity in a network formed by those layers. Hence, a secure path for information sharing is required to improve remote healthcare. A system consisting of multiple processing layers is proposed that relies on a blockchain platform for information sharing and verification. The diagnosis is divided into three categories initial diagnosis in the local device, intermediate diagnosis in the edge device, and final diagnosis in the medical server. The local device starts the medical session and reports it to the edge device. Then the edge device will broadcast the session initiation to the medical server, medical staff, and professionals. When they verify the initiated session, the edge device is eligible to establish communication with the medical server. Finally, the medical server performs the final diagnosis and updates the reports to the blockchain. All other layers can have a copy of the updated blockchain by requesting it. The proposed idea is demonstrated using the combination of Insomnia API, python, and web application flask.**

Keywords—Blockchain, distributed network, medical server

I. Introduction

The availability of resources such as high-speed internet, 4G and 5G cellular networks, Bluetooth, Wi-Fi, cloud services, small but powerful embedded systems, and powerful processing platforms have led to rapidly growing telemedicine applications. However, with this growth, there come challenges concerning whether all the security requirements are fulfilled or not for safe and sound remote healthcare applications. Like patients' health, patients' medical records, transactions, and identity are also equally important and should be kept safe and anonymous. Blockchain is a rising technology and could play a vital role in building safer remote healthcare systems with better privacy and transparency [2-4]. However, we must make sure that the strategy and architecture we develop using blockchain are feasible and implementable in the medical field.

Though the blockchain was first introduced as the foundation for cryptocurrency bitcoin due to its decentralized feature with high security, pseudo-anonymity, and data integrity it allows being applied to medical information sharing. The blockchain can play a vital role to create a secure channel to share medical data and patient information in the distributed network. Since the current state-of-art e-healthcare system consists of a local sensor, edge/fog computing, and cloud computing [5-8], the deployment of a blockchain system would be able to authenticate genuine multiple processing layers and filter the malicious nodes pretending to be one of the processing nodes in the network. Every medical record could be considered as a transaction like in the bitcoin system where several hash pointers are used to address a specific transaction.

This work proposes a unique system that consists of multiple processing layers and relies on blockchain for information sharing and verification. The medical server makes a final update to the blockchain after the three-layer diagnosis is completed. Each layer can request to obtain the updated blockchain. The chain is used to access the past records of the patient and verify other diagnostic information.

II. Literature Review

The main aim of this review was to know about various criteria that are necessary to design a blockchain-based medical application. A blockchain must be designed according to the need of the specific problem we are dealing with. Du, M., Chen et al. mentioned about three types of blockchain private blockchain, public blockchain, and consortium blockchain [5]. They differ from each other in their functionality. Public blockchains like bitcoin and Ethereum are accessible to all users in an open manner. In contrast, private blockchains are limited to authorized users only with fragile decentralization. Similarly, a consortium blockchain consists of both the features of private and public blockchains.

Du, M., Chen, et al. proposed a new mixed Byzantine fault tolerance algorithm (MBFT) as a consensus algorithm. They claimed their proposed algorithm reduces blockchain forks while ensuring high fault tolerance and consensus speed [5]. They also mentioned that the sharing platform of medical records between hospitals could reduce costs by omitting unnecessary checkups. Moreover, they used Golang to program their smart contract to upload key data to the proposed blockchain.

We must specify the role of running nodes in the network before we propose a network model for any blockchain application. In particular, a distributed system in the medical field has multiple processing nodes like local sensors, edge devices, and medical servers [4-5]. These nodes might have equal levels of authority or varying levels of authority depending upon the application requirement. For example, remote ECG diagnosis sensor nodes have direct access to the measured ECG signals. Later, signals are transferred to other nodes for further processing. Hence, the system must be designed in a way that addresses all the security requirements.

Jamil, Faisal et al. proposed the medical blockchain platform designed and developed using Hyperledger fabric [3]. Hyperledger Fabric is an open-source distributed ledger system proposed and used in the enterprise sector [9]. This platform allows to development of smart contracts in general-purpose programming languages like Java, Node.js, and Go. Jamil, Faisal et al. took the advantage of this platform to have an extensive, immutable history log, and global access to medical information from anywhere at any time [3]. Moreover, they used *Libelium* e-Health toolkit to get the physiological signal such as ECG signals of the patient. Their system architecture was divided into four parts application layer, blockchain service layer, network layer, and physical layer. The client application used the REST application programming interface (API) in the proposed blockchain platform to access the application and manage the blockchain network. Authors also claimed that their approach had a significant increase in the overall throughput and reduction in latency. The smart contract they modeled had three main categories: asset, participant, and transaction. The components chosen in these categories decide how robust and comprehensive the system could be in terms of services, security, and efficiency.

Ethereum is another blockchain development platform that could be used to model medical applications [4]. Du, Mingxiao, et al. proposed a private blockchain developed using the Ethereum platform. The processing nodes were the wearable sensor and the edge computer. The smart contract they proposed was modeled to create data queries, retrieve patient information by medical personnel, record the diagnosis, treatment, and therapy, and send feedback to patients and medical staff, and professionals [4].

III. System architecture

The proposed system model consists of multiple processing layers dedicated to providing quality remote health services. Each device has access to the blockchain where patients' information and diagnosis results are registered. In the following section, the role of each processing layer is discussed:

Local Device: This device is attached to the patient body physically since it will be monitoring the physiological signals. One of the local devices could be an ECG sensor attached to the patient's body. The ECG sensor measures the ECG signal that tracks the heart condition. The initial diagnosis is performed by this device. The embedded system uses a simple algorithm to detect abnormal physiological signals. R-peak detection algorithm in case of abnormal hear-beat detection. Necessary patients' information and initiation of medical sessions are registered in the blockchain.

Edge Device: This device receives data from the local device. Data consist of patients' private information, physiological signals, and initial diagnosis result. Edge device uses advanced diagnostic tools to verify the abnormality detected by the local device. Artificial intelligence could be used to diagnose various diseases. For example, a Convolutional neural network is used to diagnose arrhythmia and normal sinus rhythm [10]. After the analysis, the report is added to the blockchain.

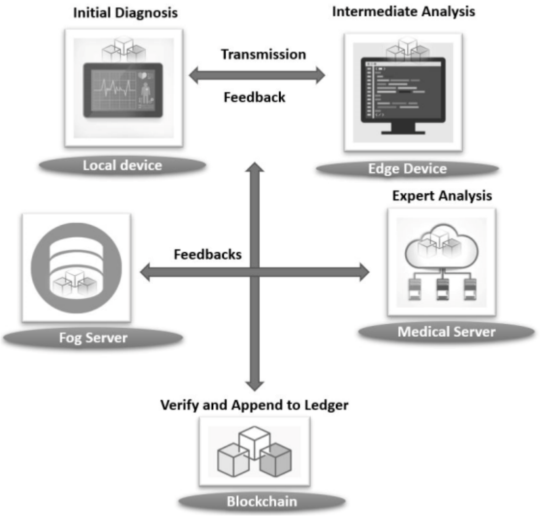


Figure 1. Proposed System Architecture

Medical Server: This server is present at the hospital. It receives every detail of the initial diagnosis and intermediate diagnosis results from other nodes. It further verifies the information recorded on the blockchain and proceeds to the final diagnosis. Analysis by medical professionals and advanced machine learning algorithms constitutes the final diagnosis.

Fog server: This device is present between the medical server and the edge device. In case of poor communication situations between the edge device and the medical server, this server will be activated. They are present nearer to edge devices than the medical servers. The main task of this node is to make the final diagnosis and verify all the diagnosis results obtained from the local device and edge device. Finally, the reports are registered on the blockchain.

Feedback: A two-way communication is present between every node to improve the quality of diagnosis. This channel is used to request additional information for improved diagnosis. That additional information could be patients' past diagnosis results, genetic history, etc.

Blockchain: This blockchain records all the medical sessions initiated by the local device. The diagnosis starts from the local device and ends at the medical server. Hence, all the details of the diagnosis by multiple layers are appended to the blockchain. The nodes are capable of updating the chain with new records, verifying the chain, retrieving the updated chain, and reporting suspicious records in the chain.

IV. Methodological framework

Multiple nodes present in the network form the distributed system. Each node has its own function and communicates with another node to transfer diagnosis information and request additional data. Each node verifies the other using digital signatures. The public key of each node is known to all the nodes present in the network. Every node sends the data along with their digital signature signed by their private key. Later, when it is received by other nodes, it is verified by using the public key of the sender node.

A. *Algorithms*

The following algorithms explain in detail the workflow of the proposed architecture with an example of ECG diagnosis:

Algorithm 1. Start of Medical Session (Local device)

Input: Physiological signal of the patient
Output: Abnormality detection and transmission to edge device

Step 1: Local device starts to measure the ECG signal
Step 2: Check for the abnormalities by calculating heartbeats per minute
Step 3: Transmit the abnormal ECG data to the edge device with a digital signature

After the local device makes its initial diagnosis the related ECG signals and patients' information are transmitted to the edge device. The edge device runs the diagnosis and broadcasts the medical session to all the nodes. Multiple nodes are present to authenticate the broadcasted medical sessions. They are medical servers and other medical personnel and staff. The following algorithm 2. elaborates the tasks of the edge device.

Algorithm 2. Broadcast the medical session (Edge device)

Input: Abnormal records received from the local device
Output: Medical session verified by nodes and intermediate diagnosis report added to the blockchain

Step 1: Initiate further diagnosis using CNN classification.
Step 2: Broadcast the initiated session to all the nodes in the network.
Step 3: Nodes verify the session and record it to the chain.
Step 4: Edge device transmits diagnosis results to the medical server

After the intermediate diagnosis, all the reports from local and edge devices are transferred to the medical server. The medical server verifies the data integrity and also scans the blockchain for the initiated medical session. The following algorithm 3. elaborates the tasks happening on the medical server.

Algorithm 3. Expert Analysis and update chain (Medical Server)

Input: Abnormal records received from the edge device
Output: Final diagnosis of the patient. Verify and update the chain with the final diagnosis report

Step 1: Previous results from the local device and edge device are analyzed by the experts.
Step 2: Run sophisticated machine learning algorithm if required
Step 3: Take necessary actions for the treatment of the patient
Step 4: Medical session is updated to the chain with the latest reports

B. *Requests Types*

The local device, edge device, fog server, and medical server can make various requests in the blockchain network. They include:

1. Obtain the chain
2. Verify the chain
3. Add to the chain
4. Update chain

Table 1. Summary of Requests

Connected Devices	Add Medical sessions to the Blockchain	Verify Sessions	Obtain Blockchain	Update Blockchain
Medical Server	✓	✓	✓	✓
Local Device	✗	✓	✓	✗
Edge Device	✓	✓	✓	✗

1. *Obtain chain:* This request could be made by all of the processing layers present in the system model. This allows each genuine layer to have one copy of the blockchain. The records are used to access patients' information and diagnosis reports.
2. *Verify chain:* This request allows the layers present in the network model to scan for any suspicious records appended to the blockchain and discard those records.
3. *Add to the chain:* The new information on medical sessions is broadcasted by the edge device. After the verification from multiple nodes, that information is added to the chain. The medical server is also authorized to add information to the chain after the final diagnosis is completed.
4. *Update chain:* When a patient's diagnosis finishes from the local device to the medical server, a final report on the patient's diagnosis is added to the blockchain by the medical server. Therefore, the blockchain is updated with every detail of that particular session with the patient.

V. RESULTS

To demonstrate the proposed idea, a blockchain was modeled using python, web application flask, and insomnia API.

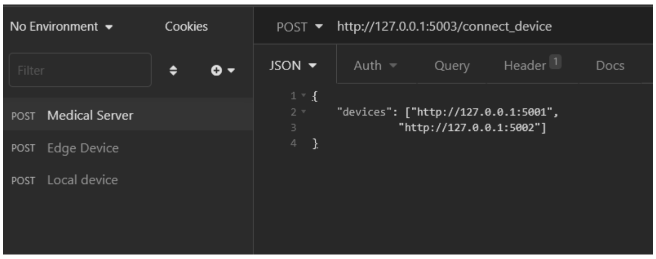
A. *Create Ports*

Three different processing layers local device, edge device, and medical server were modeled using python programming and assigned to three different ports. It was listed in the *devices.json* file with the following addresses:

```
{
   "devices": ["http://127.0.0.1:5001",
           "http://127.0.0.1:5002",
           "http://127.0.0.1:5003"]
}
```

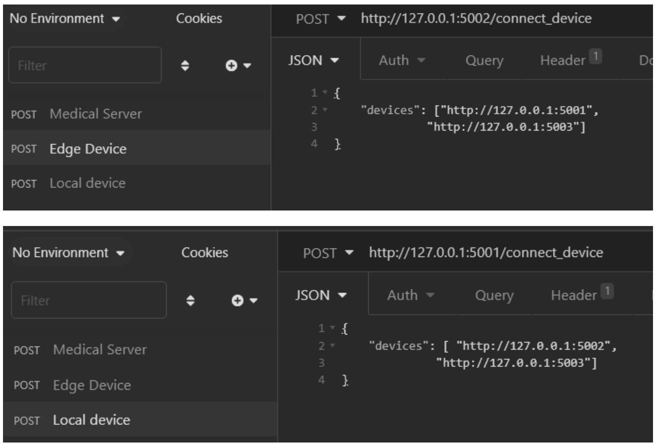
B. *Running the Python program and connecting layers*

The three python programs representing three processing layers were running on the respective address mentioned above. These 3 devices were connected using POST requests in Insomnia API as follows:



POST http://127.0.0.1:5003/connect_device

```
{
    "devices": ["http://127.0.0.1:5001",
                "http://127.0.0.1:5002"]
}
```



POST http://127.0.0.1:5002/connect_device

```
{
    "devices": ["http://127.0.0.1:5001",
                "http://127.0.0.1:5003"]
}
```

POST http://127.0.0.1:5001/connect_device

```
{
    "devices": [ "http://127.0.0.1:5002",
                 "http://127.0.0.1:5003"]
}
```

Figure 2. Interconnection of layers

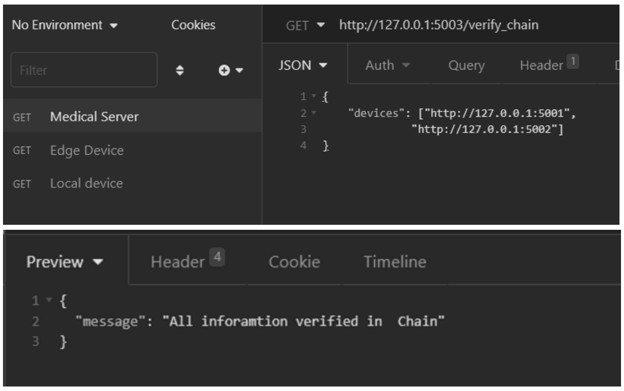
C. *Adding sessions to the Blockchain*

Medical server added multiple sessions of the patient into the blockchain using *GET* request in the Insomnia API. *http://127.0.0.1:5003/add_session* was used to send the GET request. Following is the preview of the chain when a 11th session was appended:

```
{
		"medical_sessions": [
				{
						"patientID":
"3071b1ed5a1f400c9c53329c8340bfa0",
						"receiver": "Medical_Server"
				}
		],
		"message": "Session appended",
		"previous_hash":
"a6be7ee0dbc0022e205b5563dabc33249c2fd112b6e1779d7fb7dbbe11161f
e6",
		"session_index": 11,
		"timestamp": "2022-05-03 17:54:26.740085"
}
```

D. *Verifying sessions in the Blockchain*

All of the three processing layers were able to verify the records in the blockchain using the GET request http://127.0.0.1:<port number>/verify_chain. This checked whether the chain contained invalid hash values.



GET http://127.0.0.1:5003/verify_chain

```
{
    "devices": ["http://127.0.0.1:5001",
                "http://127.0.0.1:5002"]
}
```

```
{
  "message": "All inforamtion verified in  Chain"
}
```

Figure 3. Verifying records

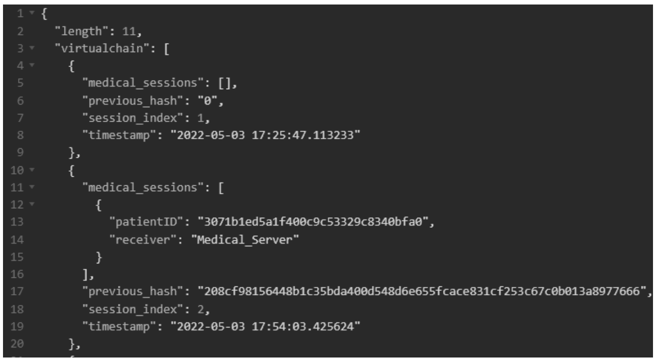
E. *Updating the Blockchain*

The medical server updates the blockchain after a session is completed. After it updates the information, other devices can make requests to obtain the updated chain. *http://127.0.0.1:5003/update_chain* was used to request to update the chain with the recently-completed session. The following is the preview of two sessions after the blockchain was updated:

```
{
		"message": "Medical Chain is up to date",
		"present_chain": [
				{
						"medical_sessions": [],
						"previous_hash": "0",
						"session_index": 1,
						"timestamp":			"2022-05-03
17:25:47.113233"
				},
				{
						"medical_sessions": [
								{
										"patientID":
"3071b1ed5a1f400c9c53329c8340bfa0",
										"receiver":
"Medical_Server"
								}
						],
						"previous_hash":
"208cf98156448b1c35bda400d548d6e655fcace831cf253c67c0b013a89776
66",
						"session_index": 2,
						"timestamp":			"2022-05-03
17:54:03.425624"
				},
				{
						"medical_sessions": [
								{
										"patientID":
"3071b1ed5a1f400c9c53329c8340bfa0",
										"receiver":
"Medical_Server"
								}
						],
						"previous_hash":
"41390e08c0ada08cfe453ea88f7ee4ee67d5550839d9b36bc34f35ae7188a1
61",
						"session_index": 3,
						"timestamp":			"2022-05-03
17:54:14.514383"
				},
```

F. *Obtaining the updated chain*

Local device and edge device were able to make an *obtain chain* request to obtain the updated blockchain. The following screenshot shows 2 of the 11 medical sessions obtained by the local and edge devices.



```
{
  "length": 11,
  "virtualchain": [
    {
      "medical_sessions": [],
      "previous_hash": "0",
      "session_index": 1,
      "timestamp": "2022-05-03 17:25:47.113233"
    },
    {
      "medical_sessions": [
        {
          "patientID": "3071b1ed5a1f400c9c53329c8340bfa0",
          "receiver": "Medical_Server"
        }
      ],
      "previous_hash": "208cf98156448b1c35bda400d548d6e655fcace831cf253c67c0b013a8977666",
      "session_index": 2,
      "timestamp": "2022-05-03 17:54:03.425624"
    },
```

Figure 4. Obtaining records

Hence, four different request types were demonstrated using the API platform Insomnia, python, and web application flask. These experiments indicated that multiple processing layers in remote health care could have a significant contribution to modeling the blockchain. These layers also make the distributed system robust to security threats. They verify the integrity of every medical record before and after appending it into the chain.

CONCLUSION

A remote healthcare system consists of multiple processing layers forming a distributed network. So, patients' private information and physiological signals must be transmitted from one layer to another using a secure path. The secure path could be modeled using a blockchain. Therefore, a blockchain-based information sharing system was proposed. The local device initiated the medical session because it was closest to the patient. Then it forwarded the initial diagnosis reports to the edge device. Edge device broadcasted the initiation of the medical session to all the nodes to verify. Nodes included medical servers and medical staff and professionals. They needed to verify the session before the edge device was authorized to transmit data to the medical server for final diagnosis. Finally, the medical server updated the chain with all the diagnosis reports. Finally, the local device and edge device were able to request a new copy of the updated blockchain.

In the proposed system architecture, local devices and edge devices were dependent on the medical server due to its power computing resources for accurate remote diagnosis. Therefore, this research work could be further improved by the use of homomorphic encryption between the edge device and the medical server.

REFERENCES

[1] PR Newswire. Retrieved March 4,2022 from .https://www.prnewswire.com/in/news-releases/internet-of-medical-things-market-to-reach-us-284-5-billion-by-2027-globally-cagr-18-5-univdatos-market-insights-850218832.html

[2] Liu, Xin, Pan Zhou, Tie Qiu, and Dapeng Oliver Wu. "Blockchain-enabled contextual online learning under local differential privacy for coronary heart disease diagnosis in mobile edge computing." IEEE Journal of Biomedical and Health Informatics 24, no. 8 (2020): 2177-2188.

[3] Jamil, Faisal, Shabir Ahmad, Naeem Iqbal, and Do-Hyeun Kim. "Towards a remote monitoring of patient vital signs based on IoT-based blockchain integrity management platforms in smart hospitals." Sensors 20, no. 8 (2020): 2195.

[4] Taralunga, Dragos Daniel, and Bogdan Cristian Florea. "A blockchain-enabled framework for mhealth systems." Sensors 21, no. 8 (2021): 2828.

[5] Du, Mingxiao, Qijun Chen, Jieying Chen, and Xiaofeng Ma. "An optimized consortium blockchain for medical information sharing." IEEE Transactions on Engineering Management 68, no. 6 (2020): 1677-1689.

[6] Moghadas, Ehsan, Javad Rezazadeh, and Reza Farahbakhsh. "An IoT patient monitoring based on fog computing and data mining: Cardiac arrhythmia usecase." Internet of Things 11 (2020): 100251.

[7] Djelouat, Hamza, Mohamed Al Disi, Issam Boukhenoufa, Abbes Amira, Faycal Bensaali, Christos Kotronis, Elena Politi, Mara Nikolaidou, and George Dimitrakopoulos. "Real-time ECG monitoring using compressive sensing on a heterogeneous multicore edge-device." Microprocessors and Microsystems 72 (2020): 102839.

[8] Sahu, Neerja, Dongming Peng, and Hamid Sharif. "Diagnosis-Steganography-Transmission: An Innovative Integrated Paradigm for ECG Healthcare." SN Computer Science 2, no. 4 (2021): 1-22.

[9] https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html [Accessed April 2021]

[10] V. Shankar, V. Kumar, U. Devagade, V. Karanth, and K. Rohitaksha, "Heart disease prediction using CNN algorithm," SN Computer Science, vol. 1, no. 3, May 2020, doi: 10.1007/s42979-020-0097-6.