Patient-Centric Token-Based Healthcare Blockchain Implementation Using Secure Internet of Medical Things

Narendra K. Dewangan, *Member, IEEE*, and Preeti Chandrakar

***Abstract*— Security and privacy are becoming increasingly difficult to preserve in today's fast-growing data world. We use the Internet of Medical Things (IoMT) to make quick diagnoses and get results. IoMT devices measure the human body's functioning in various parameters and collect, process, and store data in cloud servers. There is a public channel used to collect data from humans and send it to the cloud, which is highly unreliable and vulnerable to attacks. Another risk is storing data in a single centralized system, which can be vulnerable to a single-point failure. Blockchain is being used for secure data storage in a decentralized manner to avoid single-point failure. We propose a secure IoMT-based data collection method for patients and storing the data on the blockchain in accordance with general data protection regulation (GDPR). In the proposed system, IoMT devices send data to the cloud via the patient's personal digital assistant (PDA), and the cloud server transacts data on the blockchain. We propose a miner selection algorithm to avoid bias in the blockchain. We simulate various attacks on open channels between IoMT devices and the cloud servers. This scheme is implemented on a custom Python-based blockchain. IoT simulation is performed using the Bevywise IoT simulator and the message queuing telemetry transport (MQTT) simulator. The security protocol is analyzed using Scyther.**

***Index Terms*— Authentication, blockchain, healthcare, Internet of Medical Things (IoMT), privacy, security.**

NOMENCLATURE

A	Adversary.
Dev_k	kth IoMT device.
ID_{U_i}	Identity of U_i.
CS_j	jth cloud server.
L	160-bit secret key of ASKG.
$Priv_{(CS_j,PDA_l)}$	Secret key of cloud sever to PDA device.
ECDSA	Elliptic curve digital signature algorithm.
u_i, v_i	Biometric identity of user i.
r	Random generated nonce.
$\|\|$	Concatenation operator.
ΔT	Maximum transmission delay.
U_i	i^{th} User.
PDA_l	lth PDA of U_i.
PID_{U_i}	Peudoidentity of U_i.
ASKG	Trusted server to generate session keys.
$h()$	Cryptographic hash function.
$E()$	Encryption function.
TSP_i	ith timestamp.
TKN_i	Token of user i.
x	Random generated large prime number.
$\oplus$	Bitwise XOR operator.

Manuscript received 3 February 2022; revised 14 June 2022 and 18 July 2022; accepted 25 July 2022. Date of publication 8 August 2022; date of current version 6 December 2023. *(Corresponding author: Narendra K. Dewangan.)*

The authors are with the Department of Computer Science and Engineering, National Institute of Technology, Raipur 492010, India (e-mail: nkdewangan.phd2019.cse@nitrr.ac.in).

Digital Object Identifier 10.1109/TCSS.2022.3194872

I. INTRODUCTION

A. Healthcare and Blockchain

IN THE healthcare sector, the privacy of patient data is very important. Blockchain is a way to provide secure, immutable, structured, verified, and trusted data storage. Blockchain technology is used to store data and send a copy of that data to all nodes in the blockchain network, while, in the patient-centric blockchain networks, we are taking data from the patients and storing it in the blockchain so that the data do not contain any personal identifiable information (PII). Patient-centric blockchain focuses on the benefits of patients and comforts of the patient during the transaction and block creation process. Since it is secured cryptographically and provides anonymity using hashing of transactions and blocks, the blockchain-based healthcare system is trusted and verified, and protects the identifiable information of patients. Healthcare data are used for many other purposes, such as health insurance and medical research. The blockchain platform used for healthcare record sharing is often distinct in insurance or research agencies compared to hospital blockchain. In the healthcare sector, general data protection regulation (GDPR) compliance is a must. Blockchain offers an encryption-based data protection scheme based on cryptographic hashes that generate secure records in terms of adversary attacks and hacking scenarios. According to Hasan and Salah [1], centralized systems suffer from a single point of failure; however, because blockchain is a decentralized system, it is immune to this problem.

B. Internet of Medical Things (IoMT) and Body Area Network (BAN)

The Internet of Things (IoT) is a network of small devices working with lower power and small memory, processing their data in the clouds and reverting results to the user's portable devices. Since these little devices operate in an open, insecure network environment and are, therefore, vulnerable to hackers, data sharing security and connection should be

a top priority. The IoMT refers to the various devices used to measure and transmit data to remote doctors in order to remotely monitor patients' health. These devices can be used to measure heart rate, blood pressure, blood oxygen level, electrocardiogram (ECG), electroencephalogram (EEG), and injuries. These devices are made to fulfill a specific task for medical treatment purposes and send data to the centralized cloud server. A BAN is a network of devices used to measure different body activities. BAN connects with the cloud server to compute and store data for future research and optimization. Since the cloud also has limited storage capacity, the data are deleted after a specific period. According to Shu *et al.* [2], a medical cyber–physical system controls the embedded medical equipment through a wireless network, which senses and monitors the patient's physical data in real time. When the patient has an abnormal situation, the medical equipment sends the early warning information to the medical institution in time.

C. Security and Privacy in Healthcare Blockchains

In the healthcare-based blockchain, communication can be possible between the patient node and doctor node, patient node and pharmacy node, patient node and testing lab node, and vice versa. Since the data in the IoMT-based blockchain are collected anonymously in the cloud node, the patient's identity has remained hidden. The security layer is between the IoMT device and patient, patient portable device and cloud, and cloud server node and report accessing authorities, such as a doctor, nurse, lab, and pharmacy. According to Cai *et al.* [3], blockchain nodes are connected using the solid links of cryptographic signature and hash functions, and communication between the nodes is secure using these cryptographic functions. Authentication between data stored in the blockchain and blockchain nodes is required according to Tang *et al.* [4]. To protect patient data, blockchain security should be resistant to Sybil and man-in-the-middle attacks in cloud-based blockchains, as well as DDoS attack protection in distributed nodes and a variety of other threats.

D. Motivation and Goal

According to the above situation, we want to design a patient-centric blockchain system that measures the patient's data from the IoMT devices and transfers it to cloud servers on a public network with impregnable security and authentication. Cloud servers act as collecting nodes of data from the IoMT devices. Doctor nodes act as approvers for the blocks. The main goals of this article are given as follows.

1) To develop a patient-centric IoMT-based blockchain to collect and store patient data.
2) To measure the security requirements of the proposed system.
3) To propose the miner selection algorithm between the cloud server and doctor node.
4) The Scyther tool is used to do a security analysis of the proposed system. IoTSimulator simulates IoT infrastructure. Python is used to implement the blockchain.

E. Article Organization

The literature review of this article is described in Section II. Section III explains the proposed system and proposed algorithms. Implementation of the proposed system is explained in Section IV. Section V describes the results as well as the security analysis of the implemented system. A comparison of the proposed system with previously developed systems is described in Section VI. Section VII concludes this article with future scope.

II. Literature Review

The private, public, and consortium blockchains can all be used to implement healthcare record management on the blockchain. According to Wang *et al.* [5], wireless BAN is used as a front end in the healthcare blockchain, taking patient health data via IoMT devices, interconnecting sensor nodes, and interacting with a blockchain network in the back end. The users who are a member of the system make up the front end. Rahoof and Deepthi [6] defined the blockchain in the healthcare management system as secure, scalable, and low-storage. They created an attribute-based scheme to work with blockchain and IPFS as a scalable data storage system. Barati and Rana [7] described the use of blockchain in the audit of trail-based IoT devices under GDPR. It translates some GDPR rules into smart contracts to facilitate the automatic verification of smart objects whose roles are data controllers or processors. Ray *et al.* [8] described the IoT-based healthcare system IoBHealth data-flow architecture for integrating blockchain and IoT sensory data collected from patients to be securely accessed and managed by healthcare service providers and stakeholders.

Kim *et al.* [9] explained that the healthcare system implemented using a wireless network adds an extra layer of security while transaminating data of patients from the IoMT devices to the cloud server on an insecure network, i.e., the public internet. To protect against any network, they devised a biometric-based efficient password-authenticated key exchange (SBAKE) protocol. According to Garg *et al.* [10], the body sensors send data to the mobile devices. In the blockchain scenario, these devices are linked to the cloud server as the patient's node. This model uses key management using different cryptographic hashing features and XOR operations with each entity. This model performs an analysis of security threats and their countermeasures. They used a private blockchain model using the ripple protocol consensus algorithm (RPCA) and AVISPA to analyze and verify this protocol.

Wang *et al.* [5] explained the layered-based architecture to connect nodes and entities in the blockchain healthcare network, providing security and performance analysis of the proposed model in the simulation of the proposed system. However, they were unable to maintain the blockchain concept in this article. Cao *et al.* [11] developed a cloud-based blockchain healthcare system with a key management scheme. It described the patient's registration to the hospital, diagnosis by the doctor, and, finally, the doctor uploads the reports into the cloud. The transactions in this system are maintained on the blockchain network. Ethereum is used as a public

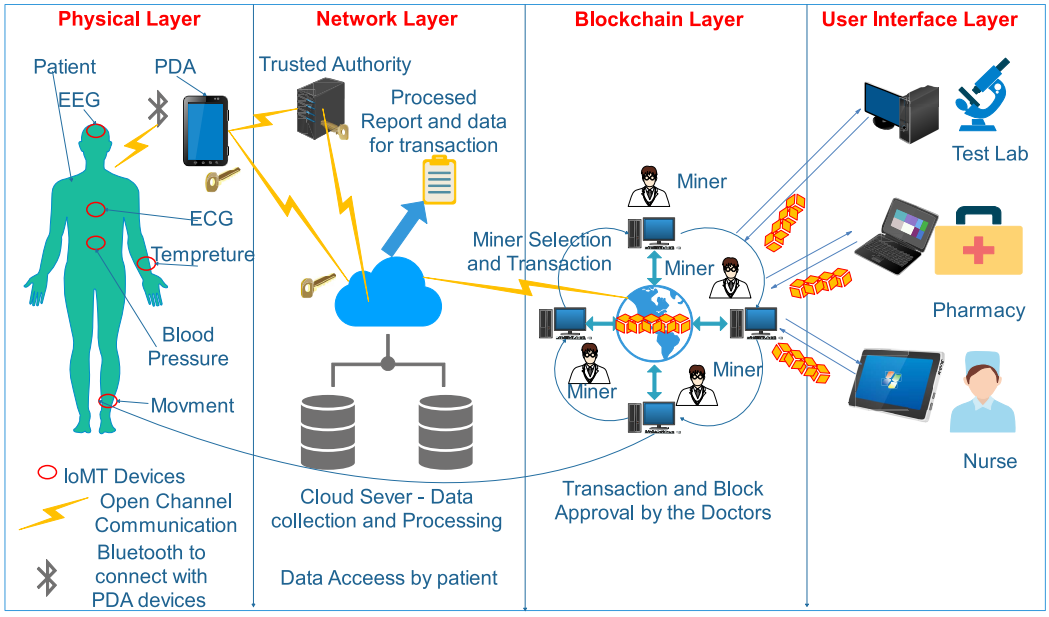


Fig. 1. Overall proposed system layered architecture.

blockchain to implement the proposed model and costs USD $37 per patient for whole transactions.

Khan *et al.* [12] proposed a secure framework for IoT-based medical sensor data authentication and encryption utilizing enhanced ECC, which provides a mechanism to send data from the patient's sensor node to the cloud server through long range radio (LoRa). They proposed an improved ECC algorithm to eliminate threats and improve the security of public network data transmission. Özyilmaz and Yurdakul [13] explained the use of low-power devices to implement IoT-based applications using the ethereum, swarm, and smart contracts. Because ethereum and swarm are employed in this system, the gas consumption during transactions is a problem. Zhaofeng *et al.* [14] proposed an IoT and big data system that is a reward-based system for collecting and utilizing data in public and permissioned blockchains. Bera *et al.* [15] proposed a scheme that resists the various attacks on IoT devices during public network transmission using 5G and blockchain. Their paper mainly focused on drones, and it discussed security and performance analysis. Połap *et al.* [16] proposed a model with consortium mechanism-based agents to classify results from the many machine learning solutions. They used artificial intelligence in particular federal learning techniques to indicate the minimization of time. Their system resulted in that they can reduce the time for diagnosis. According to Wang *et al.* [17], wireless medical sensor networks (WMSNs) have two major problems as centralized-server and physical-layer security. They propose a combination of cutting-edge technology physical unclonable functions (PUFs) and blockchain technology. Their scheme used a biometric fuzzy extractor to extract biometric information and conduct proper authentication.

A. Research Gap

The above literature survey discovered that there is a lack of nodes that are able to share information, such as a one-way medical healthcare blockchain mechanism in which doctors cannot send data to the patient, and the data coming from the sensors are directly written in the blockchain. Thus, the patient's data are going on the blockchain without the approval of doctors. It may be that sensors are perfect, but one-eye observation is required before completing the task of patient data. Major gaps are found in not implementation of blockchain and implementing the real-time IoT system or cloud server without blockchains, lack of sufficient node creation for data sharing, and approval in blockchains. The cloud is working as the miner, so the cloud approves the transactions coming from the patient node without consulting the doctors. Some of the papers mentioned the IoMT technology but did not use the blockchain and did not implement their paper with the GDPR. Some papers do not have a patient-centric blockchain system, so the patient cannot see the records. In some papers, the patient does not have control over their own data. These types of papers are implemented using blockchain technology but do not have GDPR compliance. These research gaps are solved in the proposed paper.

We are developing a blockchain model to store patient data from IoMT devices utilizing the cloud server in the proposed scheme, rather than immediately submitting transactions to the blockchain as the previously built systems did. Before approving the transactions, the doctors are monitoring the data. The miners verify the IoMT's authentication first, and then, the transactions are validated. The proposed scheme is novel in the fields of data storage, security, connectivity, blockchain monitoring, and patient-centric data.

III. Proposed System

The proposed system contains a layered architecture, which includes the patient layer (physical layer), the network layer, the blockchain layer, and the user interface layer. From Fig. 1, we can see that the physical layer includes the IoMT devices in the patient's body, and the network layer includes the secure connection establishment between the patient's personal device

and the cloud. The blockchain layer includes the transmission of data from a cloud server node to the blockchain and the mining of transactions and blocks by the selected miner. The user interface layer of this system includes the web and application interfaces used by the various human-controlled interfaces in this blockchain network. The notations used in this article are given in Nomenclature. Each layer of this proposed system is explained in the following points.

A. Physical Layer

IoMT devices are supposed to operate in this layer, which selects an exclusive symmetric bivariate polynomial

$$\chi^{x,y} = \sum_{m,n=0}^{t} a_{m,n} x^m y^n \mathrm{EGF}(p)[x, y] \tag{1}$$

of degree t over a finite field (Galois field) GF(p) ($=Z_p$), where the coefficients $a_{i,j}$'s are selected from GF(p) and $Z_p = 0.1.2., \ldots, p-1$, where p is a suitably large prime and t is more than the total number of IoMT devices to be put on the patient. The IoMT devices (Dev), the patient's mobile device personal digital assistant (PDA), and the cloud server CS are all registered in this step.

1) *Device Registration:* If the secrete key of the authentic session key generator (ASKG) is L and the identity of the IoMT device k is $\mathrm{ID}_{\mathrm{Dev}_k}$, the device's pseudoidentity is determined as

$$\mathrm{PID}_{\mathrm{Dev}_k} = h(\mathrm{ID}_{\mathrm{Dev}_k} || L). \tag{2}$$

Then, ASKG computes

$$\chi^{(\mathrm{PID}_{\mathrm{Dev}_k}, y)} = \sum_{m,n=0}^{t} (a_{m,n} (\mathrm{PID}_{\mathrm{Dev}_k})^m) y^n \tag{3}$$

which is certainly a univariate polynomial of the same degree t. Now, ASKG stored $(\mathrm{PID}_{\mathrm{Dev}_k}, \chi^{(\mathrm{PID}_{\mathrm{Dev}_k}, y)})$ in IoMT device memory.

2) *Personal Device Registration:* The PDA_l registration is required because it transmitted data collected from the IoMT devices to the cloud server. For this, ASKG has its own identity as $\mathrm{ID}_{\mathrm{ASKG}}$ and now calculates the pseudoidentity of ASKG as

$$\mathrm{PID}_{\mathrm{ASKG}} = h(\mathrm{ID}_{\mathrm{ASKG}} || L). \tag{4}$$

Let the identity of PDA_l is $\mathrm{ID}_{\mathrm{PDA}_l}$ and token generated for the PDA_l is $\mathrm{TKN}_{\mathrm{PDA}_l}$; then, we calculates the pseudoidentity of PDA_l as

$$\mathrm{PID}_{\mathrm{PDA}_l} = h(\mathrm{ID}_{\mathrm{PDA}_l} || L || \mathrm{TKN}_{\mathrm{PDA}_l}). \tag{5}$$

Now, compute peremptory credentials as

$$\mathrm{TEMPC}_{\mathrm{PDA}_l} = h(\mathrm{ID}_{\mathrm{PDA}_l} || \mathrm{ID}_{\mathrm{ASKG}} || \mathrm{TSPR}_{\mathrm{PDA}_l} || L) \tag{6}$$

where $\mathrm{TSPR}_{\mathrm{PDA}_l}$ is the timestamp of the PDA's registration in the ASKG authorized server

$$\chi^{(\mathrm{PID}_{\mathrm{PDA}_l}, y)} = \sum_{m,n=0}^{t} (a_{m,n} (\mathrm{PID}_{\mathrm{PDA}_l})^m) y^n \tag{7}$$

and then ASKG stores the information $\langle \mathrm{PID}_{\mathrm{PDA}_l}, \mathrm{TEMPC}_{\mathrm{PDA}_l}, \mathrm{PID}_{\mathrm{ASKG}}, \chi^{(\mathrm{PID}_{\mathrm{PDA}_l}, y)} \rangle$ in the PDA_l database before deployment of this PDA device.

3) *Cloud Server Registration:* ASKG selects unique identity of CS_j as $\mathrm{ID}_{\mathrm{CS}_j}$ and computes the pseudoidentity as

$$\mathrm{PID}_{\mathrm{CS}_j} = h(\mathrm{ID}_{\mathrm{CS}_j} || L). \tag{8}$$

Then, ASKG stores the information $\langle \mathrm{PID}_{\mathrm{CS}_j}, \mathrm{PID}_{\mathrm{ASKG}} \rangle$ into cloud server database before deployment.

Security and Authentication Between Devices: This part explains the key management between IoMT to PDA and PDA to CS. The overview of this part can be seen in Fig. 2.

1) *Key Management Between Dev_k and PDA_l:* Dev_k generates the current timestamp TSP_1 and sends $(\mathrm{PID}_{\mathrm{Dev}_k}, \mathrm{TSP}_1)$ to the PDA_l through the open channel. PDA_l received $(\mathrm{PID}_{\mathrm{Dev}_k}, \mathrm{TSP}_1)$ at timestamp TSP_1^*. So far, verification of time delay PDA_l calculates $|\mathrm{TSP}_1 - \mathrm{TSP}_1^*| \leq \Delta T$. If this is valid, then PDA_l generates TSP_2 and computes

$$\varphi = \chi^{(\mathrm{PID}_{\mathrm{PDA}_l}, \mathrm{PID}_{\mathrm{Dev}_k})} \tag{9}$$

the private key as

$$\mathrm{Priv}_{(\mathrm{PDA}_l, \mathrm{Dev}_k)} = h(\varphi || \mathrm{TSP}_1) \tag{10}$$

and

$$\mathrm{MSG}_1 = h(\mathrm{Priv}_{(\mathrm{PDA}_l, \mathrm{Dev}_k)} || \mathrm{TSP}_2) \oplus \mathrm{PDA}_l \tag{11}$$

and then transmits the message $\langle \mathrm{PID}_{\mathrm{PDA}_l}, \mathrm{MSG}_1, \mathrm{TSP}_2 \rangle$ to the Dev_k via the public channel. After receiving $\langle \mathrm{PID}_{\mathrm{PDA}_l}, \mathrm{MSG}_1, \mathrm{TSP}_2 \rangle$ from PDA_l at time stamp TSP_2^*, Dev_k verifies the maximum delay time

$$|\mathrm{TSP}_2 - \mathrm{TSP}_2^*| \leq \Delta T. \tag{12}$$

If the verification results are valid, then it computes

$$\varphi' = \chi^{(\mathrm{PID}_{\mathrm{Dev}_k}, \mathrm{PID}_{\mathrm{PDA}_l})} \tag{13}$$

and private key as

$$\mathrm{Priv}_{(\mathrm{Dev}_k, \mathrm{PDA}_l)} = h(\varphi' || \mathrm{TSP}_1) \tag{14}$$

and

$$\mathrm{MSG}_1' = h(\mathrm{Priv}_{(\mathrm{Dev}_k, \mathrm{PDA}_l)} || \mathrm{TSP}_2). \tag{15}$$

Then, Dev_k verifies that $\mathrm{MSG}_1' = \mathrm{MSG}_1$. If this is true, then the condition is correct for connection establishment. For future reference, both devices store the private keys as $\mathrm{Priv}_{(\mathrm{Dev}_k, \mathrm{PDA}_l)}$ and $\mathrm{Priv}_{(\mathrm{PDA}_l, \mathrm{Dev}_k)}$.

2) *Key Management Between PDA_l and CS_j:* PDA_l calculates the

$$M_1 = h(\mathrm{TKN}_l || \mathrm{TEMPC}_{\mathrm{PDA}_l}) \oplus \mathrm{PID}_{\mathrm{ASKG}} \tag{16}$$

at current timestamp TSP_1 and $M_2 = h(M_1 || \mathrm{TSP}_1 || \mathrm{PID}_{\mathrm{ASKG}})$. PDA_l sends $\langle M_1, M_2, \mathrm{TCP}_1 \rangle$ via the open channel to CS_j. The receiving timestamp of this message at CS_j is TSP_1^*. Then, CS_j verifies the validity of maximum time delay by computing $|\mathrm{TSP}_1 - \mathrm{TSP}_1^*| \leq \Delta T$. If this is valid, then CS_j computes

$$h(\mathrm{TKN}_{\mathrm{PDA}_l} || \mathrm{TEMPC}_{\mathrm{PDA}_l}) = M_1 \oplus \mathrm{PID}_{\mathrm{ASKG}} \tag{17}$$

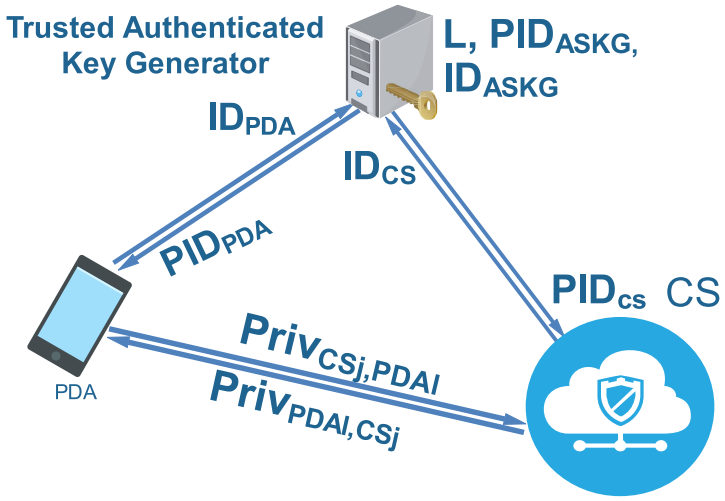


Fig. 2. Network layer of the proposed system.

and $M_2' = h(M_1||\text{PID}_{\text{ASKG}}||\text{TSP}_1)$, and then, CS_j checks that $M_2' = M_2$. If this valid, then PDA_l has verified connection with CS_j; otherwise, connection is terminated. After connection verification, CS_j further computes $M_3 = h(r||\text{PID}_{\text{CS}_j}) \oplus \text{PID}_{\text{ASKG}}$, where r is random generated nonce in the cloud server CS_j at timestamp TSP_2. Then, CS_j computes its private key as $\text{Priv}_{(\text{CS}_j,\text{PDA}_l)} = h(h(r||\text{PID}_{\text{CS}_j})||\text{PID}_{\text{ASKG}}||h(\text{TKN}_{\text{PDA}_l}||\text{TEMPC}_{\text{PDA}_l})||\text{TSP}_1||\text{TSP}_2)$ and $M_4 = h(\text{Priv}_{(\text{CS}_j,\text{PDA}_l)}||\text{TSP}_2)$ and then transmits the message $\langle M_3, M_4, \text{TSP}_2\rangle$ via the public channel to PDA_l. After receiving $\langle M_3, M_4, \text{TSP}_2\rangle$ from CS_j at timestamp TCP_2^*, PDA_l has to verify the maximum delay time by computing $|\text{TSP}_2 - \text{TSP}_2^*| \leq \Delta T$; if this is verified successfully, then PDA_l computes $h(r||\text{PID}_{\text{CS}_j}) = M_3 \oplus \text{PID}_{\text{ASKG}}$ private key $\text{Priv}_{(\text{PDA}_l,\text{CS}_j)} = h(h(r||\text{PID}_{\text{CS}_j})||\text{PID}_{\text{ASKG}}||h(\text{TKN}_{\text{PDA}_l}||\text{TEMPC}_{\text{PDA}_l}||\text{TSP}_1||\text{TSP}_2))$ and

$$M_4' = h(\text{Priv}_{(\text{PDA}_l,\text{CS}_j)}||\text{TSP}_2). \tag{18}$$

Then, PDA_l checks validity of $M_4' = M_4$; if it is valid, then CS_j is authenticated by PDA_l, and session key computation is correct. Furthermore, PDA_l computes

$$M_5 = h(\text{Priv}_{(\text{PDA}_l,\text{CS}_j)}||TSP3) \tag{19}$$

at generated timestamp TSP_3 and sends message $\langle M_5, \text{TSP}_3\rangle$ to the CS_j via open channel. After receiving $\langle M_5, \text{TSP}_3\rangle$ from PDA_l, at timestamp of TSP_3^*, CS_j computes $|\text{TSP}_3 - \text{TSP}_3^*| \leq \Delta T$ to verify the time delay; if this is verified, then computes

$$M_5' = h(\text{Priv}_{(\text{CS}_j,\text{PDA}_l)}||\text{TSP}_3) \tag{20}$$

and checks for $M_5' = M_5$; if this is valid, then PDA_l computes the correct key. Finally, both save the keys $\text{Priv}_{(\text{CS}_j,\text{PDA}_l)}$ and $\text{Priv}_{(\text{PDA}_l,\text{CS}_j)}$ for future use.

B. Network Layer

In this layer, the user registration process is done. This layer includes the user registration, login, and authentication phase between the cloud server, PDA, and IoMT devices.

1) User Registration: Let user U_i has identity as ID_{U_i} and has a 256-bit hashed token generated by the user's biometric information $(u_i, v_i) = \text{Bio}(U_i)$, which is $\text{TKN}_{U_i} = h(\text{ID}_{U_i}||u_i)$. U_i sends $\langle \text{ID}_{U_i}, \text{TKN}_{U_i}\rangle$ to the ASKG to maintain the communication between CS_j and U_i. Now, ASKG generated pseudoidentity of U_i as $\text{PID}_{U_i} = h(\text{ID}_{U_i}||L)$ and also calculates

$$\text{TEMPC}_{U_i} = h(\text{TKN}_{U_i}||\text{PID}_{\text{ASGK}}||L) \tag{21}$$

and sends back to the U_i. For the secure communication between the CS_j and ASGK, a common symmetric key $\text{CK}_{\text{CS-ASGK}}$ and $\text{CK}_{\text{ASGK-CS}}$ is agreed.

2) User Login: For user login, credential already existed in the ASKG and CS_j. When user try for login using credential $\langle \text{ID}_{U_i}, \text{TKN}_{U_i}\rangle$, CS_j requested ASKG for the

$$\text{TEMPC}_{U_i} = h(\text{TKN}_{U_i}||\text{PID}_{\text{ASGK}}||L) \tag{22}$$

and PID_{U_i} and matches data in the cloud server. If a match is verified, then the user is logged in, otherwise denied.

C. Blockchain Layer

In this layer, CS_j is working as the blockchain node, and other nodes of the blockchains are doctor, nurse, pharmacy, and test labs since, in this proposed system, the cloud server is not an approver or miner for the data provided by the IoMT devices through the public network. In this layer, ECDSA is used as the cryptographic tool, and SHA-512 is used for the hashing purpose. After collecting data and processing that data in the cloud server, CS_j has to transact that data to the doctor, which is working as the approver node. This blockchain needs to select a miner based on the consensus algorithm. Thus, the following steps are included in miner selection, consensus algorithm, transaction format, transaction approval, and block creations.

1) *Miner Selection:* Let us say that there are 100 miners in the network, and the consensus algorithm gives all of them the same chance to mine the data. Thus, the consensus algorithm searches the history in the blockchain and finds the nodes that are not yet mined

any transaction, and these nodes are available as miners for the transactions sends by the cloud server. This miner selection procedure is explained in Algorithm 1. In Algorithm 1, the input is all nodes in the blockchain that is taken. The miners, who already had a chance of mining, are searched in the blockchain. From the total number of miners, miners from the search results are eliminated, and the output provides all available miners for the mining process.

Algorithm 1 Miner Selection Algorithm

Input: List of all active miners
Output: Selected list of miners available for mining
$M \leftarrow Miners$
$B \leftarrow Blockchain$
$T \leftarrow Transaction$
$CS_j \leftarrow$ Cloud Server j
for $i \leftarrow 1$ **to** $n-1$ **do**
  $B \leftarrow search(T)$
  if $B[M] == M$ **then**
    $M = M[n] - -$;
  end if
  list[M];
end for
return list[M];

2) *Consensus Algorithm:* Algorithm 2 is based on the equal chance for all miner nodes in the blockchain network. This algorithm searches the blockchain history for the previously mined data. It matches the addresses of the miners with the miner's list. The cloud server selects the miners from the list, who have not done any mining yet, and gives the chance to the cloud server to select one of those miners. The list priority is set by the joining time of the node in the network. This network is based on the private permissioned blockchain, so all miners are preverified. When all miners have equal mining, then the consensus algorithm resets the miner list in their joining sequence.
3) *Transaction Format:* Let cloud server CS_j collected the data D_n from the different IoMT devices and process the data and make it understandable for the doctors. The transaction format is shown in Fig. 3 and includes the sender ID, i.e., PID_{CS_j}, and the receiver ID, i.e., ID of the doctor; data type means the type of diagnosis. It includes PII in terms of Boolean yes/no, timestamp of transaction, header hash of transaction, consent from the patient in the form of yes/no, and the signature of the sender CS_j.
4) *Transaction Approval and Block Creation:* When a transaction is listed in the miner's queue, the miner checks the transaction for patient consent. If the patient's consent is no, then the transaction is rejected. If consent is yes, the transaction is checked for the authenticity of the machine-generated report. If the report is accurate, the miner approves the transaction and adds it to the block. When a block is mined, it is added to the blockchain, and a copy of that block is distributed to all blockchain nodes, including the patient, as PID_{PDA_i}. As a result, the patient has access to the blockchain data.

Algorithm 2 Consensus Algorithm

Input: List of miners
Output: Block creation from transaction
$M \leftarrow Miners$
$B \leftarrow Blockchain$
$T \leftarrow Transaction$
$CS_j \leftarrow$ Cloud Server j
for $i \leftarrow 1$ **to** $n-1$ **do**
  $B \leftarrow search(T)$;
  if $B[M] == M$ **then**
    $M = M[n] - -$;
  end if
  list[M];
end for
$list[M] \leftarrow CS_j[T]$;
if $CS_j[T] \leftarrow PII$ **then**
  $reject_{transaction}$
else
  $B[M] = M \leftarrow CS_j[T]$;
  $B = B + 1$;
end if
return B

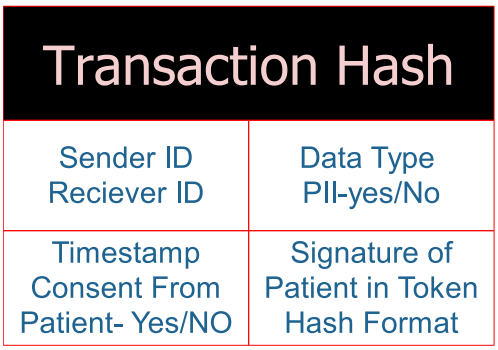


Fig. 3. Transaction format in the blockchain network.

TABLE I
USER INTERFACE ACCESS LEVELS

Node Type	Access Details
Patient	Can view all record using token based login
Doctor	Mine the transaction and view the all other data
Nurse, Pharmacy & Test Lab	View all patient data

D. User Interface Layer

In this layer the patients, doctors, nurses, pharmacies, and test labs can access the data according to their use and access control levels by the web apps and decentralized applications (dApps). The access levels for different nodes are shown in Table I.

IV. IMPLEMENTATION OF PROPOSED SYSTEM

For implementation, the proposed system is separated into three sections: 1) IoT and cloud; 2) security protocol verification using Scyther; and 3) private blockchain implementation using Python-based blockchain. Table II shows the

```
const exp: Function; const hash: Function; hashfunction h; const XOR:Function;
const h1:Function; const plus:Function;const mod:Function;
protocol Authorizedlogin(U,CS)
{
role U {
const IDi, Pu, Bu, SIDj,k,s,b,g,p,IDu,l;
fresh N1: Nonce;
var N2: Nonce;
macro b = h(b);
macro k = mod(exp(g,b),p);
macro Pij = h(XOR(XOR(h(IDi), h(h(IDi,h(s))), plus(h(N2),1)),h(IDi,k)));
macro Lij = XOR(h(N2) , h(IDi,k));
send_1(U,CS, XOR(h(SIDj,N1), h(IDu)),N1);//C1
recv_2(CS,U, XOR(h(N2), h(N1,1)));//C2
send_3(U,CS,h(XOR(XOR(h(IDu),h(h(IDu),h(s))),plus(h(N2),1)),h(IDu,k)));//Pij
send_4(U,CS,Lij);
claim_i1(U, Secret, XOR(h(SIDj,N1), h(IDu)));//C1
claim_i2(U,Secret,XOR(h(N2), h(N1,1)));//C2
claim_i3(U,Secret,h(XOR(XOR(h(IDu),h(h(IDu),h(s))),plus(h(N2),1)),h(IDu,k)));
//Pij
claim_i4(U, Secret, h(s));
claim_i10(U,Secret,k);
claim_i11(U,Secret,h(IDu));
claim_i12(U,Secret,N1);
claim_i5(U,Secret,h(N2));
claim_i13(U,Secret,Lij);
claim_i8(U,Alive);
claim_i9(U,Weakagree);
claim_i10(U, Commit, CS,N1,N2);
}
role CS{
const IDi,Pi,N2,Bi, SIDj,k,s,b,g,p,l,IDu;
var N1:Nonce; fresh N2: Nonce;
recv_1(U,CS, XOR(h(SIDj,N1), h(IDu)),N1);//C1
send_2(CS,U, XOR(h(N2), h(N1,1)));//C2
recv_3(U,CS,h(XOR(XOR(h(IDu),h(h(IDu),h(s))),plus(h(N2),1)),h(IDu,k)));//Pij
recv_4(U,CS,Lij);
claim_r13(CS,Secret,Lij); //Lij
claim_r1(CS, Secret, XOR(h(SIDj,N1), h(IDi)));//C1
claim_r2(CS, Secret, XOR(h(N2), h(N1,1)));//C2
claim_r2(CS,Secret,h(XOR(XOR(h(IDi),h(h(IDi),h(s))),plus(h(N2),1)),h(IDi,k)))
;//Pij
claim_r3(CS,Secret,h(s));
claim_r10(CS,Secret,k);
claim_r4(CS,Secret,h(IDi));
claim_r9(CS, Weakagree);
}}
```

Fig. 4. Security verification in Scyther.

TABLE II

SYSTEM REMARKS FOR IMPLEMENTATION OF THE PROPOSED SYSTEM

Property	Details
Hardware	Intel i3 processor, 4 GB RAM
Operating System	Linux Mint 20.04 64-bit
Simulators	IoT Simulator bevywise, MQTT Simulator
Blockchain	Python-based custom blockchain
Security Protocol Tester	Scyther

TABLE III

IOT PARAMETERS

Parameter	Details
Total Sensors Used	12
Total Used Devices	10
Total Run Time	480s

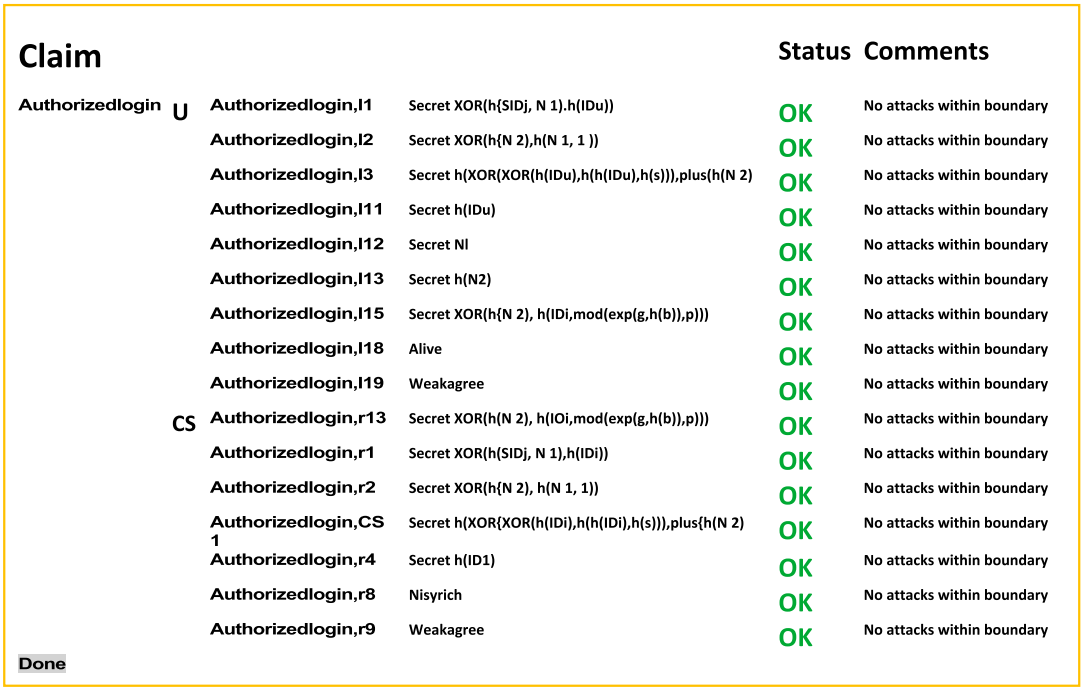
system requirements. The following are the details of this implementation.

A. IoT and Cloud Implementation

We use the Bevywise IoT simulator to create IoT and cloud infrastructures. It is a user-friendly simulation tool that allows you to simulate tens of thousands of message queuing telemetry transport (MQTT) customers in a single box. Test the functionality, performance, and capacity of cloud and on-premise MQTT applications. The test environment is described in Table III, and the IoT simulator is accessed via the local host with cloud connectivity via MQTT on the local host. The simulation environment has 12 IoMT devices with different types and configurations. The simulation run time is 480 s, and the result is discussed in Section V. Table III has a list of IoMT devices used for simulation.

B. Protocol Testing Using Scyther

We are using Scyther for the testing of the proposed security protocol. Scyther is a tool for the automatic analysis of cryptographic protocols. Scyther is a tool for analyzing cryptography protocols automatically. The ability to execute unbounded verification is one of Scyther's advantages over other tools. Traditionally, tools only evaluated the stated security attributes for a finite subset of conceivable protocol behaviors, a process known as bounded verification. The applied method can be seen in Fig. 4.



Claim				Status	Comments
Authorizedlogin	U	Authorizedlogin,I1	Secret XOR(h{SIDj, N 1).h(IDu))	OK	No attacks within boundary
		Authorizedlogin,I2	Secret XOR(h{N 2),h(N 1, 1))	OK	No attacks within boundary
		Authorizedlogin,I3	Secret h(XOR(XOR(h(IDu),h(h(IDu),h(s))),plus(h(N 2)	OK	No attacks within boundary
		Authorizedlogin,I11	Secret h(IDu)	OK	No attacks within boundary
		Authorizedlogin,I12	Secret NI	OK	No attacks within boundary
		Authorizedlogin,I13	Secret h(N2)	OK	No attacks within boundary
		Authorizedlogin,I15	Secret XOR(h{N 2), h(IDi,mod(exp(g,h(b)),p)))	OK	No attacks within boundary
		Authorizedlogin,I18	Alive	OK	No attacks within boundary
		Authorizedlogin,I19	Weakagree	OK	No attacks within boundary
	CS	Authorizedlogin,r13	Secret XOR(h(N 2), h(IOi,mod(exp(g,h(b)),p)))	OK	No attacks within boundary
		Authorizedlogin,r1	Secret XOR(h(SIDj, N 1),h(IDi))	OK	No attacks within boundary
		Authorizedlogin,r2	Secret XOR(h{N 2), h(N 1, 1))	OK	No attacks within boundary
		Authorizedlogin,CS 1	Secret h(XOR{XOR(h(IDi),h(h(IDi),h(s))),plus{h(N 2)	OK	No attacks within boundary
		Authorizedlogin,r4	Secret h(ID1)	OK	No attacks within boundary
		Authorizedlogin,r8	Nisyrich	OK	No attacks within boundary
		Authorizedlogin,r9	Weakagree	OK	No attacks within boundary

Done

Fig. 5. Scyther security verification result.

C. Blockchain Implementation

In our proposed system, the blockchain is implemented by python. The blockchain nodes are doctor, patient, nurse, pharmacy, and test lab. The resulting outcome and the performance analysis of this blockchain are discussed in Section V.

V. Result and Security Analysis

A. Security Analysis

The proposed system is secure against various attacks as proven in the Scyther implementation. The analysis of security is given as follows.

1) *Secure Against Replay Attack:* We are using an authentication key between IoMT devices to PDA and PDA to cloud server. Session key calculation is done by a trusted authority. Timestamp calculations and validation are done for the connection establishment if the adversary A is required to authentic key management. The proposed scheme has a maximum transmission delay $\triangle T$. Moreover, replaying the old transmitted messages does not provide any information gain to the attacker, which was required for "authentication and key management procedure" in Dev_k, PDA_l, CS_j, and U_i within $\triangle T$. In this way, this system is secure against replay attacks.
2) *Secure Against Man-in-the-Middle Attack:* Let an adversary A eavesdrop an authentication request message $\langle \text{MSA}_1, \text{MSA}_2, \text{TSP}_1 \rangle$, where $\text{MSA}_1 = h(\text{TKN}_l || \text{TEMPC}_{\text{PDA}_l}) \oplus \text{PID}_{\text{ASKG}}$ and $\text{MSA}_2 = h(\text{MSA}_1 || \text{TSP}_1 || \text{PID}_{\text{ASKG}}) \oplus \text{PDA}_l$ with timestamp TSP_1, which was exchanged between U_i and CS_j when A tries to update this message so that it resembles like the original authentication message, as $\langle \text{MSA}'_1, \text{MSA}'_2, \text{TSP}'_1 \rangle$, where $\text{MSA}'_1 = h(\text{TKN}_l || \text{TEMPC}_{\text{PDA}_l}) \oplus \text{PID}'_{\text{ASKG}}$ and $\text{MSA}'_2 = h(\text{MSA}'_1 || \text{TSP}'_1 || \text{PID}'_{\text{ASKG}}) \oplus \text{PDA}_l$ with the help of parameters. For the launching of MITM, A can start the generation of random nonces and current timestamp TSP'_1. However, in the absence of knowledge of private key and pseudoidentity and L, A cannot regenerate another valid authentication request.
3) *Secured Against Impersonation Attack:* Let an adversary A want to impersonate as a valid communicating entity of the network by creating an authentication request message on behalf of that entity say PDA_l. After obtaining PDA_l's authentication request $\langle \text{MSA}_1, \text{MSA}_2, \text{TSP}_1 \rangle$, where $\text{MSA}_1 = h(\text{TKN}_l || \text{TEMPC}_{\text{PDA}_l}) \oplus \text{PID}_{\text{ASKG}}$ and $\text{MSA}_2 = h(M_1 || \text{TSP}_1 || \text{PID}_{\text{ASKG}}) \oplus \text{PDA}_l$ with timestamp TSP_1, which was sent to CS_j, since these messages are generated through the secrete keys and long-term private key $\text{PID}_{\text{PDA}_l}$ and PID_{CS_j}, A is not capable to generate a valid "authentication request message" representing the legitimate user PDA_l without having the knowledge of these secret values. Therefore, this system is resilient against various impersonation attack.
4) *Secure Against Ephemeral Secret Leakage (ESL) Attack:* Secrete key of each session is calculated by the use of timestamp at each communication step by the CS_j and U_i. In the creation of the session key as $\text{SecKey}_{(\text{CS}_j, U_i)} = h(\text{PID}_{\text{PDA}_l} || \text{PDA}_l || \text{TSP}_1 || \text{TSP}_2 || h(\text{PID}_{\text{CS}'_j} || \text{PID}_{\text{ASKG}}))$, the pseudoidentities of CS_j and U_i are used. It also uses the token of the user that is generated by the biometric of the user and the randomly selected key. It is important to notice that the short- and long-term secret keys generated for the communication have the pseudoidentities of each communication node and also hash functions in messages. Therefore, the session key can only be revealed in a situation if A compromises both the "short-term" and "long-term" secret values. Furthermore, because the session keys are calculated using a variety of random nonces and timestamps

TABLE IV
IOT SIMULATION RESULTS

Parameter	Details
Total sensors used	12
Total message sends	110
Total run time	480s

values across all sessions, even if a session key is revealed for a specific session, it will not cause the revealing of session keys of other sessions because of the coalescence of short- and long-term secret values. This system is capable to protect session-temporary information attacks and also against ESL attacks.

5) *Secure Against Privilege Insider Attack:* If A knows all the registration details of the U_i, CS_j, Dev_k, and PDA_l, then also it cannot calculates the secrete key generated as $\text{SecKey}_{(\text{CS}_j,U_i)} = h(\text{PID}_{\text{PDA}_l}||\text{PDA}_l||\text{TSP}_1||\text{TSP}_2||h(\text{PID}_{\text{CS}'_j}||\text{PID}_{\text{ASKG}}))$ by the session key manager and also not able to calculate various long-term secret values, i.e., random nonces, timestamps, secret keys, and identities, as elaborated on earlier. The privileged-insider user of the ASKG does not have the knowledge of this information. Thus, A is not able the compute the "session key" on the behalf of a legitimate communicating entity. Therefore, this system is secured against privileged-insider attacks.

6) *Secure Against Device Physical Capture Attack:* Each device has the identity as $\chi^{(\text{PID}_{\text{Dev}_k},y)}$ that is utilized for the "authentication and key establishment" related work with different communicating entities. To safeguard against IoMT device physical capture attack is a crucial requirement from the security point of view. Let us say that IoMT devices are physically trapped by an adversary A. We do an evaluation of "IoMT physical capture attack" as the fraction of total secure communications, which are compromised from the capturing (physical stolen) of IoMT devices not including the communication in which the "compromised Dev_k" is clearly extended. For instance, one can calculate the probability of A's expertise to decrypt the "secure communication" between PDA_l and noncompromised Dev_k when adversaries are already compromised (under the influence of attack). From a physically stolen Dev_k, attacker A will have deduced information pseudoidentity and other parameters along with the secret pairwise session key $\text{Priv}_{(\text{Dev}_k,\text{PDA}_l)}$ shared between Dev_k and PDA_l from its memory to notice that all $\text{PID}_{\text{Dev}_k}$ and $\chi^{(\text{PID}_{\text{Dev}_k},y)}$ are different for different IoMTs. Therefore, the physical stolen of Dev_k by A can only help him/her in obtaining of secret session key between that Dev_k and PDA_l not the other session keys. Therefore, this system is unconditionally secure against the IoMT physical capture attack.

This tested result is validated by the Scyther, and the validation result is shown in Fig. 5.

B. Results From Simulations

1) *IoT Simulation:* The total IoMT devices are 12 in our simulation, the total messages captured are 110, and the total time elapsed is 480 s. This simulation result can be seen in Figs. 6 and 7 and Table IV.

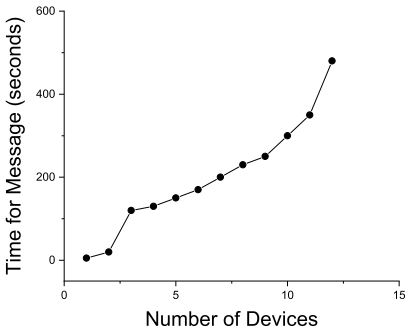


Fig. 6. Result of IoT simulations time (ms) versus the number of devices.

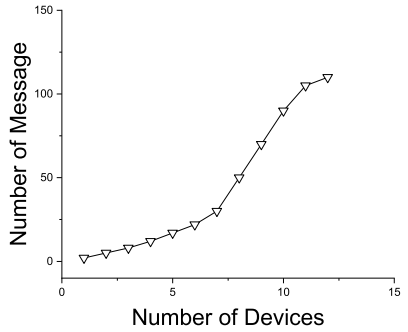


Fig. 7. Result of IoT simulations: number of messages versus number of devices.

TABLE V
BLOCKCHAIN CREATIONS

Parameter	Details
Total miners	100
Maximum time required for all miners	400s
Numbers of blocks mined	200

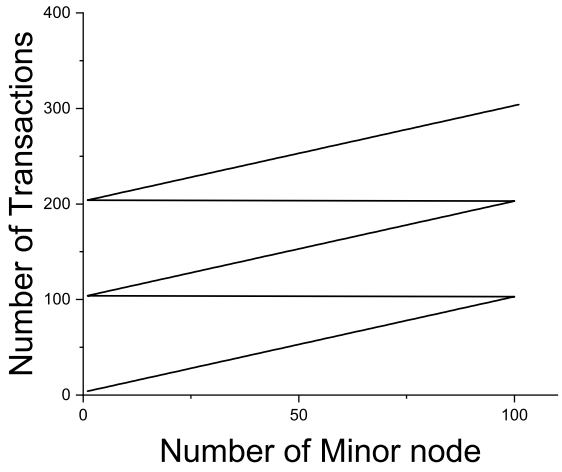


Fig. 8. Blockchain node, miners, and transaction.

2) *Blockchain Simulation:* Results from the simulation show that the node increases with time duration, and the number of patients increases. As shown in Fig. 8, we have a hundred nodes as miners who use them to mine transactions and blocks. Once the first mining is completed, the remaining 99 will be available. However,

TABLE VI
COMPARISON BETWEEN PREVIOUSLY DEVELOPED SYSTEMS AND PROPOSED SYSTEM

System	P1	P2	P3	P4	P5	P6	P7	P8	P9
[4]	Yes	No	No	NA	NA	Yes	640 bits	Yes	No
[9]	Yes	No	Yes	NA	NA	Yes	No	Yes	NA
[10]	Yes	Yes	Yes	Private	Customized	Yes	544 Bits	Yes	NA
[11]	Yes	Yes	No	Private	PoS	Yes	NA	NA	No
[12]	No	No	Yes	NA	NA	Yes	1440 bits	No	NA
[18]	Yes	No	Yes	NA	NA	Yes	1792 bits	No	NA
Proposed system	Yes	Yes	Yes	Hybrid	Customized	Yes	772 bits	No	Yes

Note: P1: Security, P2: Blockchain-based system, P3: IoMT-based system, P4: Blockchain type, P5: Consensus algorithm, P6: Security analysis, P7: Message cost, P8: Complex, P9: Patient friendly, Yes= Feature support or available, No= Feature not support or not available, NA= Feature not applicable.

TABLE VII
COMPARISON BETWEEN PREVIOUSLY DEVELOPED SYSTEMS AND PROPOSED SYSTEM IN TERMS OF SECURITY

System	P1	P2	P3	P4	P5	P6
[9]	No	Yes	No	P4	Yes	No
[10]	Yes	Yes	Yes	Yes	Yes	Yes
[15]	Yes	Yes	Yes	No	Yes	No
[18]	No	Yes	Yes	No	Yes	No
Proposed system	Yes	Yes	Yes	Yes	Yes	Yes

Note: P1: Replay attack, P2: Man-in-the-middle attack., P3: Impersonation attack, P4: ESL attack, P5: Privilege insider attack, P6: Device physical capture attack,Yes=Secure against attack, No= Not secure against attack.

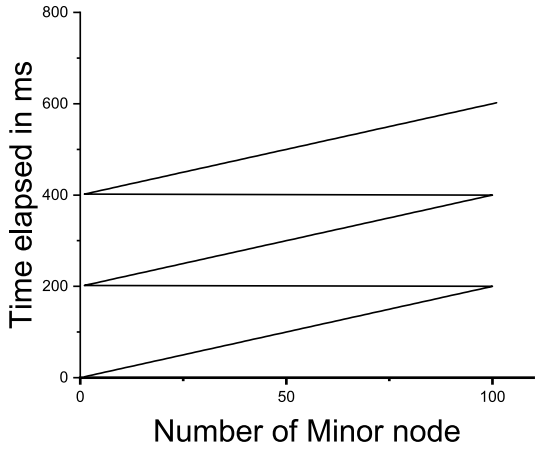


Fig. 9. Miner nodes and time elapsed (ms) with transactions.

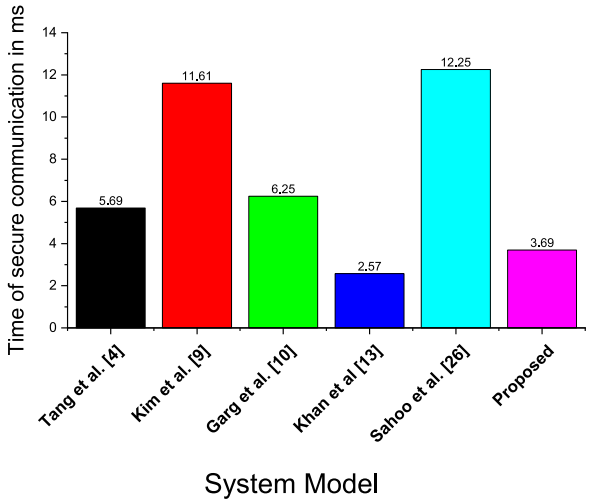


Fig. 11. Comparison of time of communication (ms) in the previously developed systems and the proposed system.

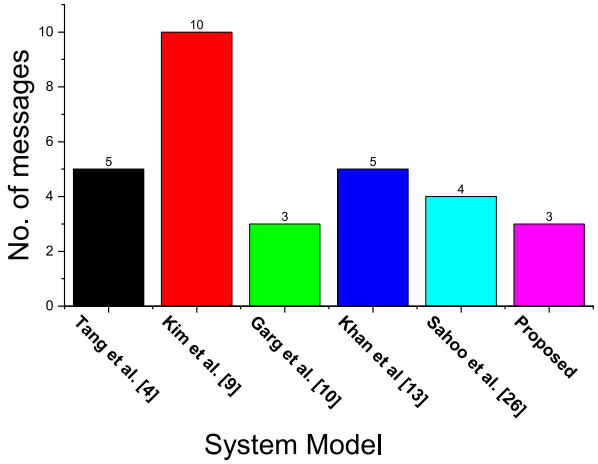


Fig. 10. Comparison of the number of messages in previously developed systems and the proposed system.

once 100 miners were completed, it restarted from the availability of 100 miners. From Fig. 9, we can see that mining one transaction from one cloud server takes 2 ms, and for a total of 100 transactions for 100 miners, it takes 200 ms. Blockchain creation is detailed in Table V.

VI. COMPARISON WITH PREVIOUSLY DEVELOPED SYSTEMS AND RESULTS

From Table VI, we can see that our proposed system is not complex as compared to previously developed systems and is patient-friendly. The proposed system is based on the authentication-based consensus algorithm. Thus, the complexity of the puzzle-solving cases is not involved in the proposed system. Here, complexity also refers to the architecture of the blockchain developed for storing patient data. We reduce the complexity of blockchain-based patient data storage by developing a user interface and prototyping the proposed scheme in a user-friendly way. A comparison of the number of messages in previously developed systems and the proposed system is shown in Fig. 10. Fig. 11 shows the comparison of time of communication (ms) in previously developed systems and the proposed system. The comparison of bits of messages in the previously developed systems and the proposed system is shown in Fig. 12. From Table VII, we can see that the proposed system is secure

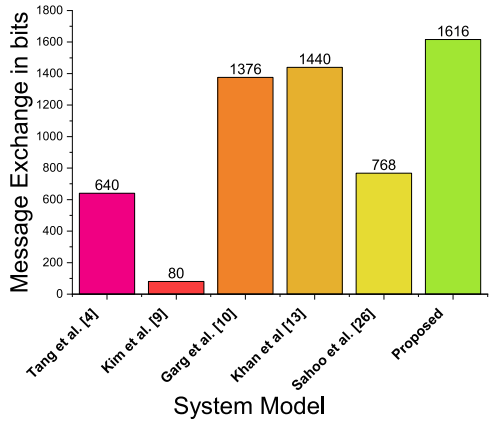


Fig. 12. Comparison of bits of messages in the previously developed systems and the proposed system.

compared to the previously developed systems [4], [9]–[12], [15], [18].

VII. Conclusion and Future Scope

IoT plays an important role in the fast-growing data-based life. Health care services are also made easy by using the IoMT devices to remotely take measures and ratings of the patients. These ratings can help doctors, and researchers do their work even in COVID-19-like situations. Our proposed system is patient-centric as the patient can see their data, and the concept of consent to share data is here. We used the IoT devices to record data from patient BAN and securely share this on the cloud server, where the data are processed for further use and transacted by the cloud server to the miner doctor node. Doctor node mined the transaction and block, and sent records to all nodes in the blockchain. In the future, we can use contactless IoMT devices to get remote data from the patients to avoid the spread of any viruses, such as COVID-19. In the future, we can add many features, such as patient hardware wallets and a selection of doctors, based on their reputation using a feedback system. Our proposed system is better in cost and performance comparison.

References

[1] H. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, vol. 6, pp. 65439–65448, 2018.

[2] H. Shu, P. Qi, Y. Huang, F. Chen, D. Xie, and L. Sun, "An efficient certificateless aggregate signature scheme for blockchain-based medical cyber physical systems," *Sensors*, vol. 20, no. 5, p. 1521, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/5/1521

[3] T. Cai, Z. Yang, W. Chen, Z. Zheng, and Y. Yu, "A blockchain-assisted trust access authentication system for solid," *IEEE Access*, vol. 8, pp. 71605–71616, 2020.

[4] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019.

[5] R. Wang, H. Liu, H. Wang, Q. Yang, and D. Wu, "Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 30–36, Dec. 2019.

[6] T. P. A. Rahoof and V. R. Deepthi, "HealthChain: A secure scalable health care data management system using blockchain," in *Distributed Computing and Internet Technology* (Lecture Notes in Computer Science), vol. 11969, D. Van Hung and M. D'Souza, Eds. Cham, Switzerland: Springer, 2020, pp. 380–391, doi: 10.1007/978-3-030-36987-3_25.

[7] M. Barati and O. Rana, "Enhancing user privacy in IoT: Integration of GDPR and blockchain," in *Blockchain and Trustworthy Systems* (Communications in Computer and Information Science), vol. 1156, Z. Zheng, H.-N. Dai, M. Tang, and X. Chen, Eds. Singapore: Springer, 2020, pp. 322–335, doi: 10.1007/978-981-15-2777-7_26.

[8] P. P. Ray, D. Dash, K. Salah, and N. Kumar, "Blockchain for IoT-based healthcare: Background, consensus, platforms, and use cases," *IEEE Syst. J.*, vol. 15, no. 1, pp. 85–94, Mar. 2021.

[9] M. Kim, J. Moon, D. Won, and N. Park, "Revisit of password-authenticated key exchange protocol for healthcare support wireless communication," *Electronics*, vol. 9, no. 5, p. 733, Apr. 2020.

[10] N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. P. C. Rodrigues, and Y. Park, "BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for Internet of Medical Things deployment," *IEEE Access*, vol. 8, pp. 95956–95977, 2020.

[11] S. Cao, X. Zhang, and R. Xu, "Toward secure storage in cloud-based eHealth systems: A blockchain-assisted approach," *IEEE Netw.*, vol. 34, no. 2, pp. 64–70, Mar./Apr. 2020.

[12] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.

[13] K. Özyilmaz and A. Yurdakul, "Designing a blockchain-based IoT with ethereum, swarm, and LoRa," *IEEE Consum. Electron. Mag.*, vol. 8, no. 2, pp. 28–34, Mar. 2019.

[14] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, "Blockchain-enabled decentralized trust management and secure usage control of IoT big data," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4000–4015, May 2020.

[15] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled Internet of Drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.

[16] D. Połap, G. Srivastava, and K. Yu, "Agent architecture of an intelligent medical system based on federated learning and blockchain technology," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102748. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212621000016

[17] W. Wang *et al.*, "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8883–8891, Jun. 2022.

[18] S. S. Sahoo, S. Mohanty, and B. Majhi, "A secure three factor based authentication scheme for health care systems using IoT enabled devices," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1419–1434, Jan. 2021, doi: 10.1007/s12652-020-02213-6.

Narendra K. Dewangan (Member, IEEE) received the bachelor's degree in computer science and engineering from the New Government Engineering College Raipur (affiliated to Swami Vivekanand Technical University, Bhilai, India), Raipur, India, in 2011, and the master's degree in cybersecurity from the School of Engineering and Research, ITM University, Raipur, in 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, National Institute of Technology, Raipur, under the supervision of Dr. Preeti Chandrakar (Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology, Raipur).

His contribution includes conceptualization, experiments, design of the methodology, and paper writing. His research interests mainly include authentication, cryptography, cybersecurity, and blockchain.

Preeti Chandrakar received the Engineering degree in computer science and engineering from the Chouksey Engineering College (affiliated to Swami Vivekanand Technical University, Bhilai, India), Bilaspur, India, in 2012, and the Ph.D. degree from the Department of Computer Science and Engineering, IIT (Indian School of Mines), Dhanbad, India, in 2018, under the supervision of Dr. Hari Om.

She is currently an Assistant Professor with the Department of Computer Science and Engineering, National Institute of Technology, Raipur, India. She has published numerous research papers in well-reputed conferences and journals. She is working in the areas of cryptography, authentication, cloud computing, and blockchain. Her contribution includes supervision, corrections, and experimental and literature suggestions.