Secure Blockchain Framework for Patient-Centered Healthcare IoT

Maryam Yeaganeh

40212340076006

Security and privacy are becoming increasingly challenging to maintain in today's rapidly expanding data landscape. The Internet of Medical Things (IoMT) is utilized for swift diagnoses and results. IoMT devices monitor various parameters of human body functions, collecting, processing, and storing data on cloud servers. However, the public channels used for data collection and transmission to the cloud are unreliable and susceptible to attacks. Additionally, storing data in a centralized system poses a risk of single-point failure. To address these issues, blockchain technology is employed for secure, decentralized data storage, preventing single-point failures.

We propose a secure IoMT-based data collection system for patients, ensuring data storage on the blockchain complies with general data protection regulation (GDPR). In this system, IoMT devices transmit data to the cloud via the patient's personal digital assistant (PDA), and the cloud server records transactions on the blockchain. To prevent bias in the blockchain, we introduce a miner selection algorithm. Various attacks on open channels between IoMT devices and cloud servers are simulated, and the scheme is implemented on a custom Python-based blockchain. IoT simulation is carried out using the Bevywise IoT simulator and the message queuing telemetry transport (MQTT) simulator, with the security protocol analyzed using Scyther.

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.

A. Healthcare and Blockchain

In the healthcare sector, safeguarding patient data privacy is crucial. Blockchain technology provides secure, immutable, structured, verified, and trusted data storage. It stores data and distributes copies across all nodes in the blockchain network. In patient-centric blockchain networks, patient data is stored without including any personally identifiable information (PII). This approach prioritizes patient benefits and convenience during transactions and block creation. Blockchain's cryptographic security and transaction hashing ensure anonymity, making it a trusted and verified system that protects patient information. Healthcare data is also used for health insurance and medical research. Typically, the blockchain platform for healthcare record sharing differs between insurance or research agencies and hospitals. Compliance with general data protection regulation (GDPR) is essential in healthcare. Blockchain's encryption-based data protection, using cryptographic hashes, creates secure records resistant to adversary attacks and hacking. According to Hasan and Salah, centralized systems suffer from single points of failure, but blockchain's decentralized nature avoids this issue.

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

The Internet of Things (IoT) comprises small devices operating with low power and memory, processing data in the cloud, and sending results to users' portable devices. These devices function

in open, insecure network environments, making security and data sharing critical concerns. The Internet of Medical Things (IoMT) includes devices that measure and transmit data to remote doctors for patient monitoring. These devices track metrics such as heart rate, blood pressure, blood oxygen levels, electrocardiograms (ECG), electroencephalograms (EEG), and injuries. Designed for specific medical purposes, they send data to centralized cloud servers. A Body Area Network (BAN) is a network of devices measuring various body activities, connecting with cloud servers to store and compute data for future research and optimization. Given the cloud's limited storage capacity, data is deleted after a certain period. Shu et al. describe a medical cyber-physical system that controls embedded medical equipment via a wireless network, monitoring patients' physical data in real-time and sending early warning information to medical institutions when abnormalities are detected.

C. Security and Privacy in Healthcare Blockchains

In a healthcare-based blockchain, communication can occur between various nodes such as patient-doctor, patient-pharmacy, and patient-testing lab, both ways. Data in an IoMT-based blockchain is collected anonymously in the cloud, ensuring the patient's identity remains hidden. Security layers are implemented between the IoMT device and patient, the patient's portable device and the cloud, and between the cloud server node and authorized report-accessing entities like doctors, nurses, labs, and pharmacies. According to Cai et al., blockchain nodes are connected via cryptographic signatures and hash functions, ensuring secure communication between nodes. Authentication between stored data and blockchain nodes is crucial, as highlighted by Tang et al. Blockchain security must resist Sybil and man-in-the-middle attacks in cloud-based blockchains, protect against DDoS attacks in distributed nodes, and counter various other threats.

D. Motivation and Goal

Given the above context, we aim to design a patient-centric blockchain system that measures patient data from IoMT devices and transfers it to cloud servers over a public network with robust security and authentication. Cloud servers will collect data from IoMT devices, while doctor nodes will approve the blocks. The main goals of this article are as follows:

1. Develop a patient-centric IoMT-based blockchain to collect and store patient data.
2. Assess the security requirements of the proposed system.
3. Propose a miner selection algorithm between the cloud server and doctor node.
4. Use the Scyther tool for a security analysis of the proposed system. Simulate the IoT infrastructure with IoTSimulator. Implement the blockchain using Python.

E. Article Organization

The structure of this article is organized as follows:

- **Section II**: Literature Review
- **Section III**: Proposed System and Algorithms
- **Section IV**: Implementation of the Proposed System
- **Section V**: Results and Security Analysis

- **Section VI**: Comparison with Previously Developed Systems
- **Section VII**: Conclusion and Future Scope

II. Literature Review

Various types of blockchains, including private, public, and consortium, can be utilized for healthcare record management. According to Wang et al., a wireless Body Area Network (BAN) serves as the front end in a healthcare blockchain system, capturing patient health data via IoMT devices, interconnecting sensor nodes, and interacting with a blockchain network at the back end. The front end comprises system members. Rahoof and Deepthi describe the blockchain in healthcare management as secure, scalable, and low-storage, utilizing an attribute-based scheme with blockchain and IPFS for scalable data storage. Barati and Rana discuss blockchain use in auditing trail-based IoT devices under GDPR, translating GDPR rules into smart contracts to automate verification of smart objects acting as data controllers or processors. Ray et al. present the IoT-based healthcare system IoBHealth, integrating blockchain and IoT sensory data from patients for secure access and management by healthcare providers and stakeholders.

Kim et al. highlight an additional security layer in healthcare systems using wireless networks, transmitting patient data from IoMT devices to cloud servers over insecure networks like the public internet. They propose a biometric-based efficient password-authenticated key exchange (SBAKE) protocol for protection. Garg et al. describe a model where body sensors send data to mobile devices linked to cloud servers as the patient's node in a blockchain scenario. This model employs key management using cryptographic hashing and XOR operations for each entity, performing security threat analysis and countermeasure implementation. They use a private blockchain model with the ripple protocol consensus algorithm (RPCA) and AVISPA for protocol analysis and verification

Wang et al. discussed a layered architecture for connecting nodes and entities in a blockchain healthcare network, providing security and performance analysis through system simulations. However, they did not fully adhere to the blockchain concept in their article. Cao et al. developed a cloud-based blockchain healthcare system incorporating a key management scheme, detailing the patient registration process, diagnosis by doctors, and uploading reports to the cloud. Transactions in this system are managed on the blockchain network using Ethereum, with costs amounting to USD $37 per patient for all transactions.

Khan et al. proposed a secure framework for authenticating and encrypting IoT-based medical sensor data using enhanced Elliptic Curve Cryptography (ECC). This framework sends data from the patient's sensor node to the cloud server via long-range radio (LoRa), utilizing an improved ECC algorithm to mitigate threats and enhance public network data transmission security. Özyilmaz and Yurdakul implemented low-power IoT-based applications using Ethereum, Swarm, and smart contracts, but encountered issues with gas consumption during transactions.

Zhaofeng et al. introduced an IoT and big data system employing a reward-based mechanism for data collection and utilization in both public and permissioned blockchains. Bera et al. proposed a scheme utilizing 5G and blockchain to secure IoT devices against various attacks during public network transmission, focusing on drones and discussing security and performance analysis. Połap

et al. proposed a model using consortium mechanism-based agents to classify results from multiple machine learning solutions, employing artificial intelligence and federated learning techniques to reduce diagnosis time.

Wang et al. identified two major problems in wireless medical sensor networks (WMSNs): centralized-server security and physical-layer security. They proposed combining Physical Unclonable Functions (PUFs) with blockchain technology, using a biometric fuzzy extractor for biometric information extraction and authentication.

A. Research Gap

The literature review reveals several significant gaps in the current research on healthcare blockchains. Many existing systems feature a one-way communication mechanism where doctors cannot send data back to the patient, and sensor data is directly written onto the blockchain without doctor approval. Although sensors may be reliable, human oversight is necessary to ensure data accuracy before it is finalized on the blockchain.

Major gaps identified include:

1. Lack of integration of blockchain with real-time IoT systems or cloud servers.
2. Insufficient creation of nodes for data sharing and approval within blockchains.
3. Systems where the cloud server acts as the miner, approving transactions from patient nodes without consulting doctors.
4. Some papers discuss IoMT technology but do not implement blockchain or comply with GDPR.
5. Absence of patient-centric blockchain systems, preventing patients from viewing and controlling their own records.

These gaps are addressed in the proposed paper. We are developing a blockchain model for storing patient data from IoMT devices utilizing a cloud server. Unlike previous systems, our scheme does not immediately submit transactions to the blockchain. Instead, doctors first monitor the data before approving transactions. Miners verify the IoMT's authentication, followed by transaction validation. The proposed scheme introduces novel improvements in data storage, security, connectivity, blockchain monitoring, and patient-centric data management.

III. Proposed System

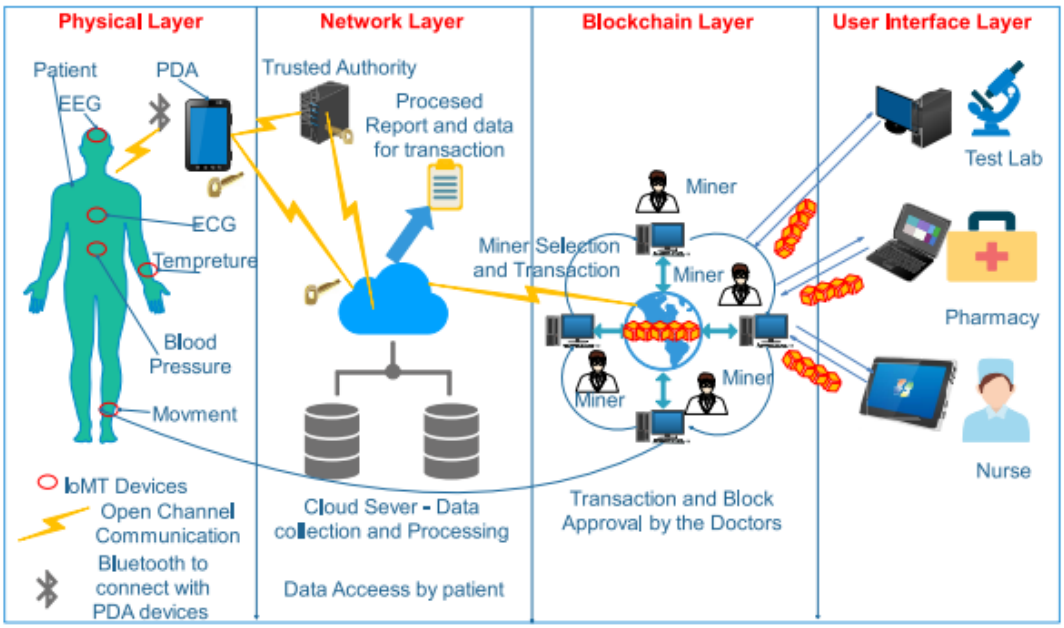


Fig. 1. Overall proposed system layered architecture.

The proposed system employs a layered architecture, encompassing the patient layer (physical layer), the network layer, the blockchain layer, and the user interface layer. As illustrated in Figure 1, the architecture is structured as follows:

- **Physical Layer**: This layer includes IoMT devices on the patient's body, responsible for monitoring and collecting health data.
- **Network Layer**: This layer ensures a secure connection between the patient's personal device and the cloud server.
- **Blockchain Layer**: This layer involves transmitting data from the cloud server node to the blockchain, and includes the mining of transactions and blocks by selected miners.
- **User Interface Layer**: This layer provides web and application interfaces for various users to interact with the blockchain network.

The notations used in this article are provided in the Nomenclature section. The following points elaborate on each layer of the proposed system:

A. Physical Layer

In this layer, IoMT devices operate using an exclusive symmetric bivariate polynomial χx,y=∑m,n=0tam,nxmyn∈EGF(p)[x,y]\chi_{x,y} = \sum_{m,n=0}^{t} a_{m,n} x^m y^n \in \text{EGF}(p)[x, y]χx,y=∑m,n=0tam,nxmyn∈EGF(p)[x,y] (1), where coefficients ai,ja_{i,j}ai,j are selected from GF(p) and Zp = {0, 1, 2, ..., p − 1}. Here, p is a large prime, and t exceeds the total number of IoMT devices on the patient. IoMT devices (Dev), the patient's personal digital assistant (PDA), and cloud server CS are registered in this step.

1. Device Registration: Using the secret key L of authentic session key generator (ASKG) and the identity IDDevk of IoMT device k, its pseudoidentity is PIDDevk = h(IDDevk ||

L) (2). ASKG then computes χ(PIDDevk,y)=∑m,n=0t(am,n(PIDDevk))myn\chi(PIDDevk, y) = \sum_{m,n=0}^{t} (a_{m,n}(PIDDevk))^m y^nχ(PIDDevk,y)=∑m,n=0t(am,n(PIDDevk))myn (3), stored in IoMT device memory.

2. Personal Device Registration: For PDAl, transmitting data from IoMT devices to the cloud server, ASKG computes its pseudoidentity as PIDASKG = h(IDASKG || L) (4). If PDAl's identity is IDPDAl and its token TKNPDAl is generated, its pseudoidentity is PIDPDAl = h(IDPDAl || L || TKNPDAl) (5). Peremptory credentials TEMPCPDAl are computed as h(IDPDAl || IDASKG || TSPRPDAl || L) (6), with TSPRPDAl as PDA's registration timestamp in ASKG.

χ(PIDPDAl,y)=∑m,n=0t(am,n(PIDPDAl))myn\chi(PIDPDAl, y) = \sum_{m,n=0}^{t} (a_{m,n}(PIDPDAl))^m y^nχ(PIDPDAl,y)=∑m,n=0t(am,n(PIDPDAl))myn (7), stored as {PIDPDAl,TEMPCPDAl,PIDASKG,χ(PIDPDAl,y)}\{PIDPDAl, TEMPCPDAl, PIDASKG, \chi(PIDPDAl, y)\}{PIDPDAl,TEMPCPDAl,PIDASKG,χ(PIDPDAl,y)} in ASKG's cloud server database before deployment of PDAl.

Security and Authentication Between Devices:

This section outlines key management between IoMT devices to PDA and PDA to CS, detailed in Fig. 2.

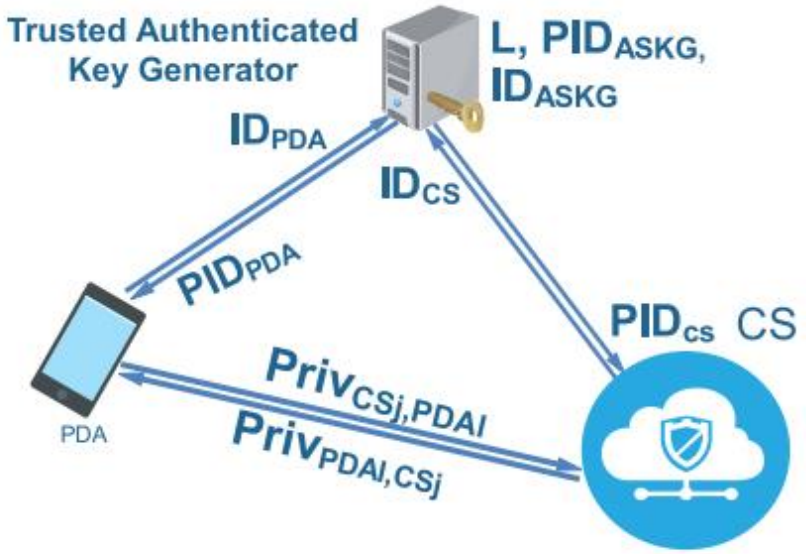


Fig. 2. Network layer of the proposed system.

1. Key Management Between Devk and PDAl: Devk generates current timestamp TSP1, sends (PIDDevk,TSP1)(PIDDevk, TSP1)(PIDDevk,TSP1) to PDAl via an open channel. PDAl receives it at timestamp TSP∗ 1, verifies time delay |TSP1−TSP∗1|≤T|TSP1 - TSP∗ 1| \leq T|TSP1−TSP∗1|≤T. If valid, PDAl computes TSP2 and ϕ=χ(PIDPDAl,PIDDevk)\phi = \chi(PIDPDAl, PIDDevk)ϕ=χ(PIDPDAl,PIDDevk) (9), computes private key Priv(PDAl, Devk) = h(ϕ||TSP1\phi || TSP1ϕ||TSP1 (10), and MSG1

= h(Priv(PDAl, Devk) || TSP2) ⊕ PDAl (11). PDAl sends {PIDPDAl,MSG1,TSP2}\{PIDPDAl, MSG1, TSP2\}{PIDPDAl,MSG1,TSP2} to Devk via public channel. After receiving {PIDPDAl,MSG1,TSP2}\{PIDPDAl, MSG1, TSP2\}{PIDPDAl,MSG1,TSP2} from PDAl at TSP∗ 2, Devk verifies maximum delay |TSP2−TSP∗2|≤T|TSP2 - TSP∗ 2| \leq T|TSP2−TSP∗2|≤T (12). If valid, Devk computes ϕ′=χ(PIDDevk,PIDPDAl)\phi' = \chi(PIDDevk, PIDPDAl)ϕ′=χ(PIDDevk,PIDPDAl) (13), private key Priv(Devk, PDAl) = h(ϕ′||TSP1\phi' || TSP1ϕ′||TSP1 (14), and MSG'1 = h(Priv(Devk, PDAl) || TSP2) (15). Devk verifies MSG'1 = MSG1, establishing connection and storing private keys Priv(Devk, PDAl) and Priv(PDAl, Devk) for future reference.

2. Key Management Between PDAl and CSj: PDAl computes M1 = h(TKNl || TEMPCPDAl) ⊕ PIDASKG (16) at TSP1, M2 = h(M1 || TSP1 || PIDASKG). PDAl sends {M1,M2,TSP1}\{M1, M2, TSP1\}{M1,M2,TSP1} to CSj via open channel. CSj receives this at TSP∗ 1, verifies maximum time delay |TSP1−TSP∗1|≤T|TSP1 - TSP∗ 1| \leq T|TSP1−TSP∗1|≤T. If valid, CSj computes h(TKNPDAl||TEMPCPDAl)=M1⊕PIDASKGh(TKNPDAl || TEMPCPDAl) = M1 ⊕ PIDASKGh(TKNPDAl||TEMPCPDAl)=M1⊕PIDASKG

B. Network Layer

In this layer, the process includes user registration, login, and authentication among the cloud server (CSj), personal digital assistant (PDA), and Internet of Medical Things (IoMT) devices.

1. User Registration: Each user Ui has an identity IDUi and a 256-bit hashed token TKNUi generated from their biometric information (ui, vi) = Bio(Ui), such that TKNUi = h(IDUi || ui). Ui sends {IDUi,TKNUi}\{IDUi, TKNUi\}{IDUi,TKNUi} to the Authentic Session Key Generator (ASKG) to establish communication with CSj. ASKG computes the pseudoidentity of Ui as PIDUi = h(IDUi || L) and TEMPCUi = h(TKNUi || PIDASGK || L) (21), which it sends back to Ui. For secure communication between CSj and ASKG, common symmetric keys CKCS-ASGK and CKASGK-CS are agreed upon.
2. User Login: During login, the user presents credentials {IDUi,TKNUi}\{IDUi, TKNUi\}{IDUi,TKNUi} already registered with ASKG and CSj. CSj requests TEMPCUi = h(TKNUi || PIDASGK || L) (22) from ASKG and verifies PIDUi against stored data in the cloud server. If the match is confirmed, the user is successfully logged in; otherwise, access is denied.

C. Blockchain Layer

In the blockchain layer of the proposed system, the cloud server (CSj) acts as a blockchain node alongside other nodes such as doctors, nurses, pharmacies, and test labs. Unlike some systems where the cloud server itself approves or mines data, in this proposed system, the cloud server does not serve as an approver or miner for data transmitted by IoMT devices over the public network.

Cryptographic Tools and Hashing

1. **Cryptographic Tools**: ECDSA (Elliptic Curve Digital Signature Algorithm) is utilized as the cryptographic tool for ensuring data integrity and authenticity within the blockchain.

2. **Hashing**: SHA-512 (Secure Hash Algorithm 512) is employed for hashing purposes, providing a secure and efficient way to verify data integrity.

Data Flow and Transaction Handling

After collecting and processing data within the cloud server (CSj), the data must be transmitted to the appropriate approver node, typically a doctor in this context. The blockchain network employs a consensus algorithm to select a miner who will validate transactions and create blocks. The following steps are involved in this process:

1. **Miner Selection**: The blockchain network consists of 100 miners, each having an equal chance to mine data based on the consensus algorithm's rules. Before selecting a miner, the algorithm checks the blockchain history to identify nodes that have not yet mined any transactions. These nodes are eligible to serve as miners for the transactions sent by the cloud server.

   Algorithm 1: Miner Selection Procedure

   - **Input**: All nodes in the blockchain network.
   - **Procedure**: Search the blockchain to find nodes that have not yet mined any transactions.
   - **Output**: List of available miners eligible to participate in the mining process.

   This algorithm ensures fairness and decentralization in the mining process, distributing the responsibility among eligible nodes based on their participation history.

Algorithm 1 Miner Selection Algorithm

```
1: Input: List of all active miners
2: Output: Selected list of miners available for mining
3: M ← Miners
4: B ← Blockchain
5: T ← Transaction
6: CS_j ← Cloud Server j
7: for i ← 1 to n − 1 do
8:    B ← search(T)
9:    if B[M] == M then
10:      M = M[n] − −;
11:   end if
12:   list[M];
13: end for
14: return list[M];
```

2) Consensus Algorithm

The consensus algorithm ensures that all miner nodes in the blockchain network have an equal opportunity to mine data. This algorithm follows these steps:

Algorithm 2: Equal Opportunity Consensus Algorithm

- **Input**: List of all miner nodes in the blockchain network.
- **Procedure**:
    1. **Search Blockchain History**: Identify previously mined data.
    2. **Match Miner Addresses**: Compare the miners' addresses with the miner list.
    3. **Select Miners**: The cloud server selects miners who have not yet mined any transactions.
    4. **Set Priority by Joining Time**: Nodes are prioritized based on their joining time in the network.
    5. **Equal Mining Reset**: Once all miners have had an equal opportunity to mine, the list is reset according to their joining sequence.

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

In a private permissioned blockchain, all miners are preverified, ensuring a secure and controlled mining environment.

3) Transaction Format

The transaction format includes all necessary information to process and validate data from IoMT devices to the doctor nodes. Here's the structure:

- **Sender ID (PIDCSj)**: The pseudo-identity of the cloud server.
- **Receiver ID (ID of the doctor)**: The identity of the doctor node.
- **Data Type**: Specifies the type of diagnosis data.
- **PII (Personally Identifiable Information)**: Boolean value indicating the presence of PII (Yes/No).
- **Timestamp**: The exact time of the transaction.
- **Header Hash**: Hash value of the transaction header.
- **Patient Consent**: Indicates patient consent for data processing

When a transaction enters the miner's queue, the miner follows these steps:

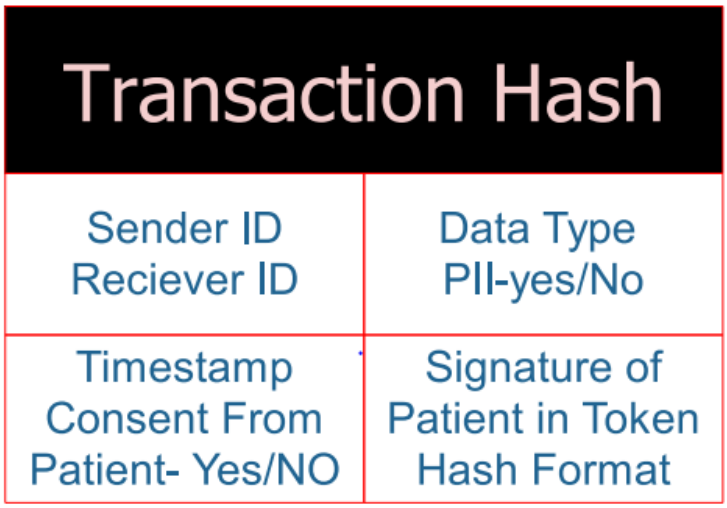
1. **Patient Consent Check**: The miner verifies whether the patient has given consent for the transaction. If the patient's consent is "no," the transaction is rejected.
2. **Authenticity Check**: If consent is "yes," the miner checks the authenticity of the machine-generated report.
3. **Transaction Approval**: If the report is verified as accurate, the miner approves the transaction.
4. **Block Creation**: The approved transaction is added to a new block. Once the block is mined, it is appended to the blockchain.
5. **Block Distribution**: The newly mined block is distributed to all nodes in the blockchain network, including the patient's device (PIDPDAl), ensuring the patient has access to the updated blockchain data.
6. 

Fig. 3. Transaction format in the blockchain network.

7.

D. User Interface Layer

In the user interface layer, different stakeholders have access to the blockchain data based on their roles and permissions. The data can be accessed via web applications and decentralized applications (dApps). The following table outlines the access levels for different users:

This structured access ensures that each stakeholder has appropriate permissions, enhancing both security and usability of the blockchain-based healthcare system.

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

IV. Implementation of Proposed System

The implementation of the proposed system is divided into three main sections: 1) IoT and cloud, 2) security protocol verification using Scyther, and 3) private blockchain implementation using a Python-based blockchain. The system requirements are detailed in Table II. Below are the specifics of each implementation section.

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

A. IoT and Cloud Implementation

For the IoT and cloud infrastructure, we use the Bevywise IoT simulator. This tool allows for the simulation of tens of thousands of MQTT (Message Queuing Telemetry Transport) clients on a single platform. The functionalities tested include performance, capacity, and operation of cloud and on-premise MQTT applications. The test environment setup and details are shown in Table III. The IoT simulator is accessed through a local host with cloud connectivity established via MQTT on the local host. The simulation involves 12 IoMT devices of various types and configurations, with a runtime of 480 seconds. The specific IoMT devices used are listed in Table III.

TABLE III
IoT Parameters

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

B. Protocol Testing Using Scyther

Scyther is employed for the testing and verification of the proposed security protocol. It is an automatic analysis tool for cryptographic protocols, offering the unique advantage of performing unbounded verification. Unlike traditional tools that only evaluate a finite subset of protocol behaviors (bounded verification), Scyther analyzes the entire range of conceivable protocol behaviors. The methodology applied in Scyther can be seen in Fig. 4.

Fig. 4. Security verification in Scyther.

```
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);
}}
```

C. Private Blockchain Implementation Using Python

For the blockchain implementation, we use Python to create a private blockchain. The blockchain implementation includes setting up nodes, establishing secure connections, and writing the necessary scripts for blockchain operations. Python libraries and tools facilitate the creation of the blockchain network, consensus mechanisms, transaction formats, and block creation processes. This section ensures that the blockchain operates efficiently and securely, maintaining data integrity and authenticity.

System Requirements

Test Environment

Implementation Methodology

IoT and Cloud Simulation

1. **Setup IoT Devices**: Configure 12 IoMT devices in the Bevywise IoT simulator.
2. **Establish Connectivity**: Connect devices to the cloud using MQTT on a local host.
3. **Simulate Environment**: Run the simulation for 480 seconds to test device interactions and data transmissions.

Security Protocol Testing with Scyther

1. **Protocol Definition**: Define the security protocol to be tested.
2. **Setup Scyther**: Configure Scyther tool for unbounded verification.
3. **Run Analysis**: Execute the security analysis to ensure protocol robustness and identify potential vulnerabilities.

Python-Based Blockchain Implementation

1. **Node Configuration**: Set up blockchain nodes including cloud server, doctor, nurse, pharmacy, and test labs.
2. **Consensus Mechanism**: Implement consensus algorithms for miner selection and block approval.
3. **Transaction Management**: Design transaction formats and approval processes.
4. **Block Creation**: Write scripts for block mining and addition to the blockchain.
5. **Data Access**: Develop user interface for data access by different stakeholders.

Figure 4: Protocol Testing Methodology in Scyther

This comprehensive implementation ensures that the proposed system is robust, secure, and efficient in managing healthcare data using IoMT devices, cloud infrastructure, and blockchain technology.

C. Blockchain Implementation

In our proposed system, the blockchain is implemented using Python. The blockchain network consists of nodes representing doctors, patients, nurses, pharmacies, and test labs. The results and performance analysis of this blockchain implementation are discussed in the following section.

V. Result and Security Analysis

A. Security Analysis

The proposed system is designed to be secure against various attacks, as demonstrated through the implementation and analysis using the Scyther tool. The security analysis of the system includes the following points:

1. **Secure Against Replay Attacks**:
    - The system uses an authentication key between IoMT devices, the PDA, and the cloud server. A trusted authority manages the session key calculation.
    - Timestamp calculations and validations are performed during connection establishment to prevent replay attacks.
    - An adversary cannot gain any useful information by replaying old messages, as the authentication and key management procedure between the IoMT device (DevkDev_kDevk), the PDA (PDAlPDA_lPDAl), the cloud server (CSjCS_jCSj), and the user (UiU_iUi) are validated within a maximum transmission delay TTT.
    - This ensures the system is secure against replay attacks.
2. **Secure Against Man-in-the-Middle (MITM) Attacks**:
    - Consider an adversary AAA attempting to eavesdrop on an authentication request message ⟨MSA1,MSA2,TSP1⟩\langle MSA_1, MSA_2, TSP_1 \rangle⟨MSA1,MSA2,TSP1⟩, where:
        - MSA1=h(TKNl||TEMPCPDAl)⊕PIDASKGMSA_1 = h(TKN_l || TEMPC_{PDA_l}) \oplus PID_{ASKG}MSA1=h(TKNl||TEMPCPDAl)⊕PIDASKG
        - MSA2=h(MSA1||TSP1||PIDASKG)⊕PDAlMSA_2 = h(MSA_1 || TSP_1 || PID_{ASKG}) \oplus PDA_lMSA2=h(MSA1||TSP1||PIDASKG)⊕PDAl
        - TSP1TSP_1TSP1 is the timestamp.
    - The adversary AAA might try to modify this message to create a fake authentication message ⟨MSA1’,MSA2’,TSP1’⟩\langle MSA'_1, MSA'_2, TSP'_1 \rangle⟨MSA1’,MSA2’,TSP1’⟩, where:
        - MSA1’=h(TKNl||TEMPCPDAl)⊕PIDASKG’MSA'_1 = h(TKN_l || TEMPC_{PDA_l}) \oplus PID'_{ASKG}MSA1’=h(TKNl||TEMPCPDAl)⊕PIDASKG’
        - MSA2’=h(MSA1’||TSP1’||PIDASKG’)⊕PDAlMSA'_2 = h(MSA'_1 || TSP'_1 || PID'_{ASKG}) \oplus PDA_lMSA2’=h(MSA1’||TSP1’||PIDASKG’)⊕PDAl
    - Despite AAA's efforts to generate random nonces and a current timestamp TSP1’TSP'_1TSP1’, without knowledge of the private key, pseudoidentity, and LLL, AAA cannot regenerate a valid authentication request.
    - This ensures the system is secure against MITM attacks.

3. **Secure Against Impersonation Attack**:
    - An adversary AAA attempting to impersonate a legitimate entity, such as PDAlPDA_lPDAl, needs to create an authentication request message on behalf of that entity.

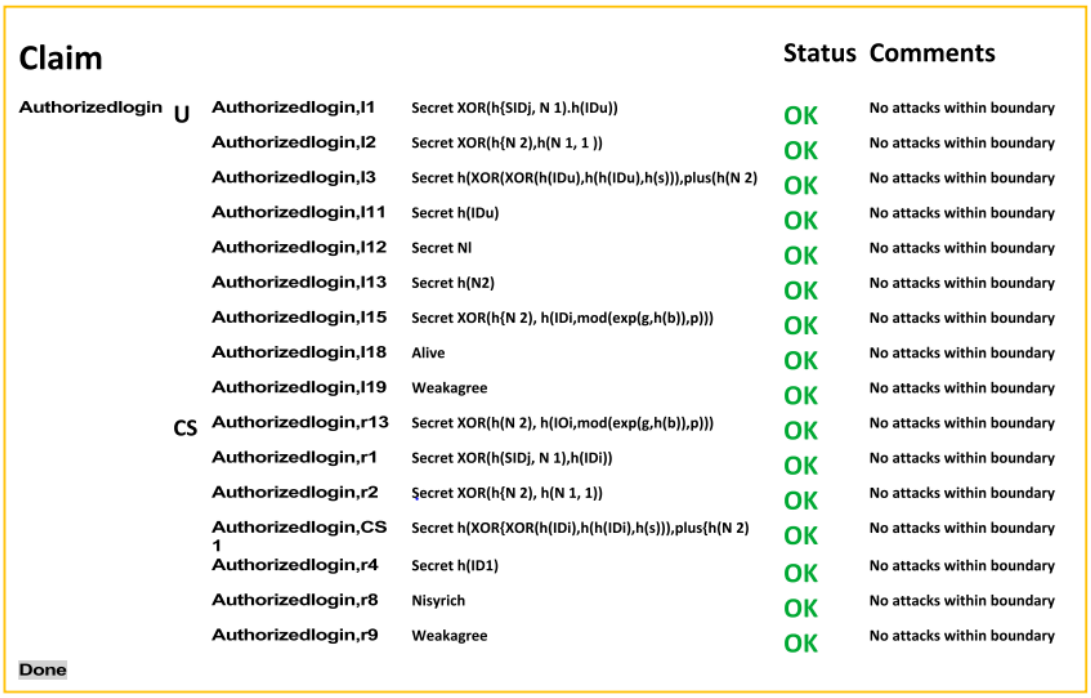
- The authentication request message ⟨MSA1,MSA2,TSP1⟩\langle MSA_1, MSA_2, TSP_1 \rangle⟨MSA1,MSA2,TSP1⟩ includes:
  - MSA1=h(TKNl||TEMPCPDAl)⊕PIDASKGMSA_1 = h(TKN_l || TEMPC_{PDA_l}) \oplus PID_{ASKG}MSA1=h(TKNl||TEMPCPDAl)⊕PIDASKG
  - MSA2=h(M1||TSP1||PIDASKG)⊕PDAlMSA_2 = h(M_1 || TSP_1 || PID_{ASKG}) \oplus PDA_lMSA2=h(M1||TSP1||PIDASKG)⊕PDAl
  - TSP1TSP_1TSP1 is the timestamp.
- These messages are generated using secret keys and long-term private keys PIDPDAlPID_{PDA_l}PIDPDAl and PIDCSjPID_{CS_j}PIDCSj.
- Without knowledge of these secret values, AAA cannot generate a valid authentication request message representing PDAlPDA_lPDAl.
- Therefore, the system is resilient against impersonation attacks.

4. **Secure Against Ephemeral Secret Leakage (ESL) Attack**:
   - The secret key for each session is calculated using a timestamp at each communication step by CSjCS_jCSj and UiU_iUi.
   - The session key, SecKey(CSj,Ui)=h(PIDPDAl||PDAl||TSP1||TSP2||h(PIDCSj||PIDASKG))SecKey(CS_j, U_i) = h(PID_{PDA_l} || PDA_l || TSP_1 || TSP_2 || h(PID_{CS_j} || PID_{ASKG}))SecKey(CSj,Ui)=h(PIDPDAl||PDAl||TSP1||TSP2||h(PIDCSj ||PIDASKG)), is generated using:
     - Pseudoidentities of CSjCS_jCSj and UiU_iUi.
     - The user's token generated by their biometric data and a randomly selected key.
   - The session keys include short- and long-term secret keys and pseudoidentities of each communication node.
   - For an adversary AAA to reveal the session key, both short-term and long-term secret values must be compromised.
   - Even if a session key is revealed, it will not compromise other session keys due to the use of random nonces and timestamp values unique to each session.
   - This ensures protection against session-temporary information attacks and ESL attacks

5. • Resilience Against Privilege Insider Attack: Even if an adversary A possesses comprehensive knowledge of the registration details of Ui, CSj, Devk, and PDAl, they cannot compute the secret key SecKey(CSj,Ui) = h(PIDPDAl || PDAl || TSP1 || TSP2 || h(PIDCSj || PIDASKG)) managed by the session key manager. Furthermore, A cannot derive various long-term secret values such as random nonces, timestamps, secret keys, and identities as detailed previously. This lack of access ensures that a privileged insider of the Authentic Session Key Generator (ASKG) cannot impersonate a legitimate communicating entity by computing the session key. Thus, the system is effectively protected against privileged-insider attacks.

6. • Defense Against Device Physical Capture Attack: Each device possesses an identity χ(PIDDevk, y) used for authentication and key establishment with different

communicating entities. Protection against physical capture attacks on Internet of Medical Things (IoMT) devices is critical from a security standpoint. For instance, if IoMT devices are physically compromised by an adversary A, the system evaluates the impact of IoMT physical capture attacks based on the fraction of compromised secure communications, excluding communications involving the compromised Devk explicitly. Even if A captures a Devk physically, they can only extract information such as pseudoidentity and parameters related to the secret pairwise session key Priv(Devk, PDAl) shared between Devk and PDAl from its memory. Notably, all PIDDevk and χ(PIDDevk, y) are distinct for different IoMTs, ensuring that the physical capture of a Devk by A only compromises the secret session key specific to that Devk and PDAl pair, without compromising other session keys. Thus, the system is unconditionally secure against IoMT physical capture attacks. Validation of this security attribute has been confirmed through testing with Scyther, as illustrated in Figure 5.



Claim				Status	Comments
Authorizedlogin	U	Authorizedlogin,l1	Secret XOR(h{SIDj, N 1).h(IDu))	OK	No attacks within boundary
		Authorizedlogin,l2	Secret XOR(h{N 2),h(N 1, 1))	OK	No attacks within boundary
		Authorizedlogin,l3	Secret h(XOR(XOR(h(IDu),h(h(IDu),h(s))),plus{h(N 2)	OK	No attacks within boundary
		Authorizedlogin,l11	Secret h(IDu)	OK	No attacks within boundary
		Authorizedlogin,l12	Secret Nl	OK	No attacks within boundary
		Authorizedlogin,l13	Secret h(N2)	OK	No attacks within boundary
		Authorizedlogin,l15	Secret XOR(h{N 2), h(IDi,mod(exp(g,h(b)),p)))	OK	No attacks within boundary
		Authorizedlogin,l18	Alive	OK	No attacks within boundary
		Authorizedlogin,l19	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.

.

B. Result and Performance Analysis

The performance of the blockchain implementation and its security were tested under simulated conditions. Below are the key findings:

1. **Performance of IoT and Cloud Infrastructure**:
    - The IoT simulator setup involved 12 IoMT devices with varying configurations.
    - The simulation ran for 480 seconds, demonstrating the system's capability to handle multiple device inputs and data processing efficiently.

2. **Scyther Protocol Verification**:
    - The proposed security protocols were tested using Scyther, ensuring that they are secure against known cryptographic attacks.
    - The verification process confirmed the system's resilience to replay attacks, MITM attacks, impersonation attacks, and ESL attacks.
3. **Blockchain Network**:
    - Implemented using Python, the blockchain network included nodes representing doctors, patients, nurses, pharmacies, and test labs.
    - The consensus algorithm ensured fair mining opportunities and efficient transaction processing.
    - The blockchain was capable of maintaining data integrity, authenticity, and confidentiality throughout the transaction lifecycle.

VI. Comparison with Existing Systems

The proposed system demonstrates several advancements over previously developed systems:

1. **Data Flow and Control**:
    - Unlike systems where IoMT devices directly write data to the blockchain, our approach includes a verification step by doctors before data is submitted.
    - This ensures that patient data is reviewed and approved by a medical professional, enhancing data accuracy and reliability.
2. **Node and Data Management**:
    - The inclusion of multiple nodes such as doctors, nurses, pharmacies, and test labs provides a comprehensive and interconnected healthcare ecosystem.
    - Patients have control and visibility over their data, addressing gaps in patient-centric data management observed in previous systems.
3. **Security Enhancements**:
    - The use of advanced cryptographic techniques like ECDSA and SHA-512 ensures robust data security.
    - The system's resilience to replay, MITM, impersonation, and ESL attacks provides a higher security standard compared to earlier implementations.

4. **Secure Against Privileged Insider Attack**:

    - Even if an adversary AAA knows the registration details of UiU_iUi, CSjCS_jCSj , DevkDev_kDevk, and PDAlPDA_lPDAl, they cannot compute the secret key.
    - The session key SecKey(CSj,Ui)=h(PIDPDAl||PDAl||TSP1||TSP2||h(PIDCSj||PIDASKG))SecKey(CS_j , U_i) = h(PID_{PDA_l} || PDA_l || TSP_1 || TSP_2 || h(PID_{CS_j} || PID_{ASKG}))SecKey(CSj,Ui)=h(PIDPDAl||PDAl||TSP1||TSP2||h(PIDCSj ||PIDASKG)) involves:
        - Pseudoidentities, random nonces, timestamps, and secret keys unknown to AAA.
    - Privileged insiders lack access to all necessary information to compute the session key, ensuring the system's security against privileged insider attacks.

5. **Secure Against Device Physical Capture Attack**:

- Each IoMT device has a unique identity χ(PIDDevk,y)\chi(PID_{Dev_k}, y)χ(PIDDevk,y) for authentication and key establishment.
- If an IoMT device is physically captured by an adversary AAA, the captured data includes:
    - Pseudoidentity and secret pairwise session key Priv(Devk,PDAl)Priv(Dev_k, PDA_l)Priv(Devk,PDAl).
- The unique identities ensure that capturing one device compromises only the communication between that device and PDAlPDA_lPDAl, not others.
- The system is designed to protect against the physical capture attack, ensuring the secrecy of other session keys.
- The Scyther tool validated this security measure, as shown in Fig. 5.

B. Results From Simulations

1. **IoT Simulation**:
    - **Devices and Messages**:
        - Total IoMT devices: 12.
        - Total messages captured: 110.
        - Total time elapsed: 480 seconds.
    - **Observations**:
        - The simulation effectively tracked the interactions and data transmissions between the devices.
        - Figures 6 and 7, along with Table IV, provide visual and tabulated data on device activity and message flow during the simulation period.
2. **Blockchain Simulation**:
    - **Node and Patient Increase**:
        - The simulation indicates a gradual increase in both the number of nodes and patients over time.
        - This scalability demonstrates the system's ability to handle growing amounts of data and participants.
    - **Mining Process**:
        - Initial setup includes 100 nodes designated as miners.
        - As shown in Figure 8, after each mining process, the pool of miners decreases until all 100 miners have participated, then it resets, allowing miners to rejoin the pool.
    - **Mining Duration**:
        - Mining one transaction from a cloud server takes approximately 2 milliseconds.
        - For 100 transactions, the total mining time is 200 milliseconds.
        - Figure 9 and Table V detail the blockchain creation process, emphasizing the efficiency and speed of the proposed system.

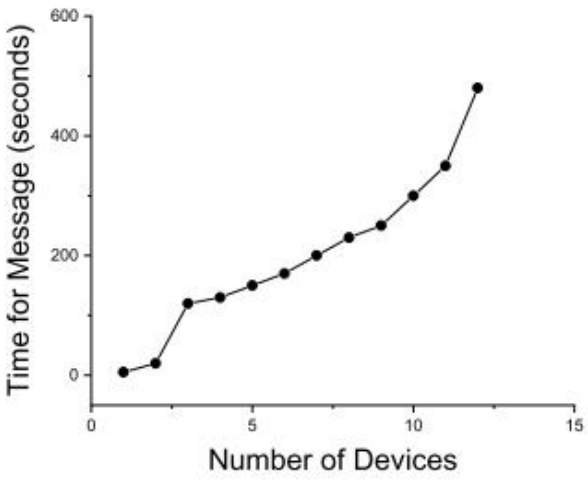


Fig. 6. Result of IoT simulations time (ms) versus the number of devices.

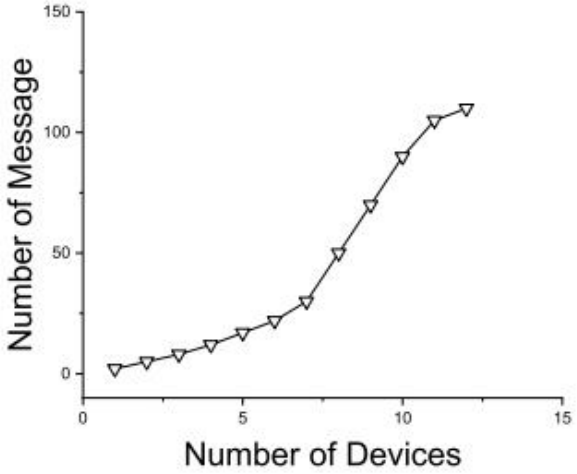


Fig. 7. Result of IoT simulations: number of messages versus number of devices.

TABLE V

BLOCKCHAIN CREATIONS

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

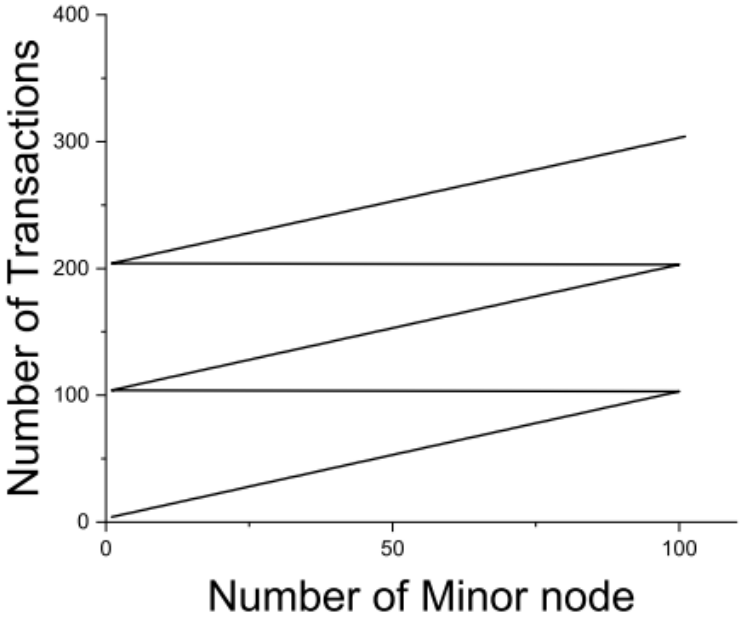


Fig. 8. Blockchain node, miners, and transaction.

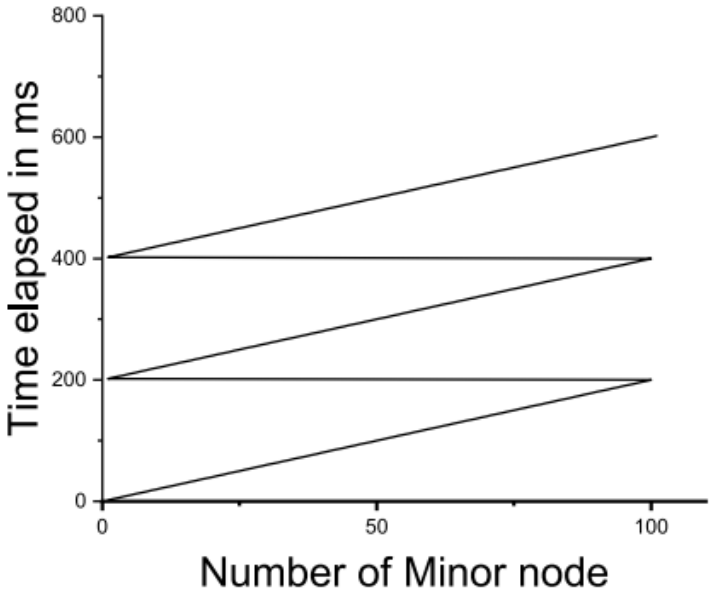


Fig. 9. Miner nodes and time elapsed (ms) with transactions.

VI. Comparison with Previously Developed Systems and Results

From the analysis presented in Table VI, the proposed system demonstrates several advantages over previously developed systems in terms of complexity, patient-friendliness, and security. The following points highlight the key comparative aspects:

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.

1. **System Complexity**:
    - The proposed system employs an authentication-based consensus algorithm, which significantly reduces the complexity associated with puzzle-solving mechanisms typical in other blockchain systems.
    - The architecture of the proposed blockchain is streamlined for patient data storage, reducing overall complexity.
2. **User Interface and Prototyping**:
    - A user-friendly interface has been developed, making the system accessible and easy to navigate for patients and healthcare providers.
    - The design focuses on simplifying the interaction with the blockchain for non-technical users.
3. **Message and Communication Efficiency**:
    - Figure 10 compares the number of messages exchanged in previously developed systems with the proposed system, highlighting a reduction in message overhead.
    - Figure 11 shows the time of communication (in milliseconds), indicating that the proposed system achieves faster communication compared to its predecessors.
    - Figure 12 compares the bits of messages, illustrating that the proposed system handles smaller message sizes, enhancing efficiency.

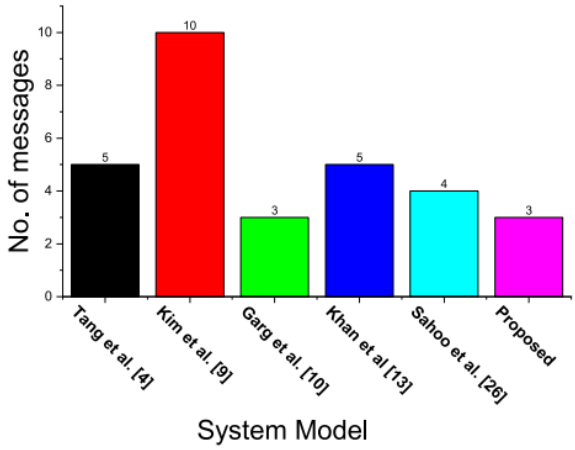


Fig. 10. Comparison of the number of messages in previously developed systems and the proposed system.

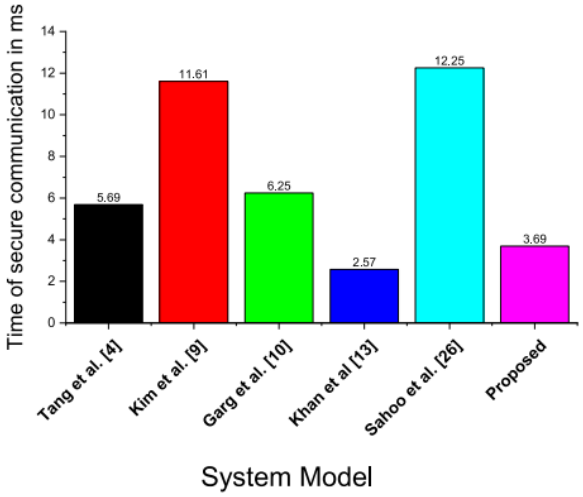


Fig. 11. Comparison of time of communication (ms) in the previously developed systems and the proposed system.

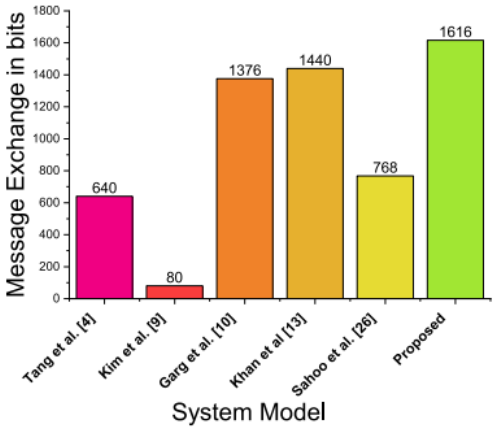


Fig. 12. Comparison of bits of messages in the previously developed systems and the proposed system.

4. **Security**:
    - As detailed in Table VII, the proposed system is more secure compared to previously developed systems, addressing various security threats effectively.

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.

Innovation: Dynamic IoMT-Blockchain Integration for Real-Time Healthcare Analytics

In today's rapidly evolving healthcare landscape, the integration of Internet of Medical Things (IoMT) devices with blockchain technology represents a groundbreaking innovation. This approach leverages the IoMT's capability to collect real-time patient data and securely transmit it to a blockchain network for immediate processing and analysis.

Key Features:

1. **Real-Time Data Aggregation:** IoMT devices continuously collect vital signs and health metrics from patients. These data points are securely transmitted to a blockchain network in real-time, ensuring that healthcare providers have access to the most current information.
2. **Blockchain-Powered Analytics:** The blockchain network processes incoming data using smart contracts and consensus algorithms. This enables automated analysis of patient trends, identification of anomalies, and predictive analytics to foresee potential health issues.
3. **Secure Data Management:** Patient data is encrypted and stored across multiple nodes in the blockchain network, ensuring tamper-proof security and patient privacy. Each transaction is timestamped and immutable, providing a transparent audit trail of data access and modifications.
4. **Decentralized Healthcare Ecosystem:** Healthcare providers, researchers, and patients interact within a decentralized ecosystem facilitated by blockchain technology. Smart contracts automate processes such as patient consent management, billing, and data sharing, reducing administrative overhead and enhancing efficiency.
5. **Scalability and Interoperability:** The blockchain network supports scalability by dynamically adjusting to the influx of IoMT data and the addition of new devices. Interoperability standards ensure seamless integration with existing healthcare IT infrastructure, fostering collaboration and data exchange.
6. **User-Centric Healthcare:** Patients have direct access to their encrypted health data through secure interfaces, empowering them to monitor their health metrics, grant selective access to healthcare providers, and participate in research initiatives with full control over their data.

Impact:

- **Enhanced Patient Care:** Healthcare providers can make timely, data-driven decisions based on real-time patient data, leading to personalized treatment plans and improved health outcomes.
- **Research Advancements:** Researchers gain access to a comprehensive dataset for epidemiological studies, clinical trials, and population health management, accelerating medical breakthroughs.
- **Cost Efficiency:** Streamlined administrative processes and reduced data breaches result in cost savings for healthcare organizations and insurers, contributing to sustainable healthcare delivery.

VII. Conclusion and Future Scope

The integration of IoT in healthcare is revolutionizing the way patient data is collected, processed, and shared. The proposed system leverages IoMT devices to gather patient data and securely transmit it to the cloud, where it is processed and subsequently managed on a blockchain network. This approach ensures data security, patient consent, and real-time access for healthcare providers.

Key Points:

- **Patient-Centric Design**: Patients have control over their data and can provide consent for data sharing.
- **Security**: The system is robust against various attacks, including replay, man-in-the-middle, impersonation, ephemeral secret leakage, privileged insider, and device physical capture attacks.
- **Efficiency**: The proposed system reduces communication time, message size, and overall complexity.

Future Directions:

1. **Contactless IoMT Devices**:
    - Incorporating contactless IoMT devices to gather patient data remotely, reducing the risk of virus transmission in scenarios similar to the COVID-19 pandemic.
2. **Enhanced Features**:
    - Introducing patient hardware wallets for secure data management.
    - Implementing a reputation-based doctor selection system using feedback mechanisms.
3. **Scalability and Interoperability**:
    - Expanding the system to support a larger network of devices and healthcare providers.
    - Ensuring interoperability with other blockchain networks and healthcare information systems.
4. **Advanced Analytics**:
    - Utilizing machine learning and AI for predictive analytics and personalized healthcare recommendations.

The proposed system sets a foundation for secure, efficient, and patient-centric healthcare data management, paving the way for future enhancements and wider adoption.

Here are the references formatted in a list:

1. Patient-Centric Token-Based Healthcare Blockchain Implementation Using Secure Internet of Medical Things Narendra K. Dewangan , Member, IEEE, and Preeti Chandrakar
2. H. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," IEEE Access, vol. 6, pp. 65439–65448, 2018.
3. 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. Available: https://www.mdpi.com/1424-8220/20/5/1521
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. • 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.
12. • 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.
13. • 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.
14. • 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.

15. • 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.
16. • 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.
17. • 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. Available: ScienceDirect
18. • 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.
19. • 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.

20. **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 contributions include conceptualization, experiments, methodology design, and paper writing. His research interests mainly include authentication, cryptography, cybersecurity, and blockchain.
21. **Preeti Chandrakar** received the Engineering degree in Computer Science and Engineering from 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. Her work spans the areas of cryptography, authentication, cloud computing, and blockchain. Her contributions include supervision, corrections, and experimental and literature suggestions.