Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 2558469, 14 pages
https://doi.org/10.1155/2023/2558469

Research Article

SecureMed: A Blockchain-Based Privacy-Preserving Framework for Internet of Medical Things

Wajid Rafique,[1] **Maqbool Khan**,[2] **Salabat Khan**,[3] **and Juma Said Ally**[4]

[1]*Department of Computer Science and Operations Research, University of Montreal, Canada*
[2]*Department of IT and Computer Science, PAF-IAST, Pakistan*
[3]*College of Computer Science and Software Engineering, Shenzhen University, China*
[4]*Department of Electronics and Telecommunication Engineering, Mbeya University of Science and Technology, Tanzania*

Correspondence should be addressed to Juma Said Ally; juma.ally@must.ac.tz

Received 8 September 2022; Revised 11 October 2022; Accepted 12 October 2022; Published 21 April 2023

Academic Editor: Ruinian Li

The Internet of Medical Things (IoMT) connects a huge amount of smart sensors with the Internet for healthcare service provisioning. IoMT's privacy-preserving becomes a challenge considering the life-saving data collected and transferred through IoMT. Traditional privacy protection techniques use centralized management strategies, which lead to a single point of failure, lack of trust, state modification, information disclosure, and identity theft. Edge computing enables local computation of IoMT data, which reduces traffic to the cloud and also helps in accomplishing latency-sensitive healthcare applications and services. This paper proposes a novel framework (i.e., SecureMed) that uses blockchain-based distributed authentication implemented at the edge cloudlets to enforce privacy protection. In SecureMed, IoMT devices interact with edge cloudlets using smart contracts. It uses trusted edge nodes to implement an authentication algorithm that uses public/private key matching to authenticate IoMT. Experimental evaluation performed using the Pythereum blockchain shows that SecureMed outperforms the traditional blockchain scheme based on latency, bandwidth consumption, deployment time, scalability, and accuracy. Therefore, it can be used to protect the edge-enabled IoMT from privacy attacks and to ensure end-to-end healthcare service provisioning.

1. Introduction

The networking of medical objects (e.g., machines, sensors, and healthcare devices) is transforming traditional healthcare setup into smart healthcare systems [1]. The future of healthcare systems corresponds to the Internet of Medical Things (IoMT) which can operate autonomously by sending controlled messages for healthcare service provisioning. IoMT produces real-world data at an enormous speed. According to the World Healthcare Organization (WHO), there will be a lack of healthcare professionals by the end of 2030, which proclaims the need for improved healthcare solutions to ensure efficient disease diagnosis and treatment [2]. The world population is expected to reach 9.8 billion by 2050 [3]. This rapid growth will also give rise to the elderly population (greater than 65 years to 16%) by 2050. This will cause an increase in the population vulnerable to chronic diseases, which is responsible for 65% of total deaths and accounts for up to 60% of global healthcare expenditure by 2025 [4]. Thus, efficient healthcare service provisioning techniques are required to fulfill future human needs. The COVID-19 pandemic has changed the way medical devices were manufactured in the past and has speed up IoMT production [5] due to high spread and death rate [6]. The development of new intelligent monitoring systems for disease detection and prediction has increased the demand for IoMT, which has also been motivated by the fast development in the Internet of Things (IoT) [7, 8].

IoMT comprises heterogeneous objects that can efficiently be managed by Software-Defined Networking (SDN) [9]. It enhances programmability and provides granular control over massive traffic generated by IoMT. Due to the resource limitation of IoMT, edge computing is widely used to fulfill the resource demands of IoMT. Increasing deployment of IoMT in healthcare raises privacy disclosure issues. IoMT captures highly sensitive data which makes its privacy an important concern, e.g., medical image data, personal information, medical records, and disease diagnosis information [10]. Any unauthorized access can further cause anomalies in diagnosis and treatment [11]. Due to the resource limitations, it is challenging to implement privacy-preserving solutions directly on the resource-limited IoMT [12]. Therefore, network-level security solutions are adopted for IoMT security and privacy. Hence, greater privacy challenges and attack scenarios arise. Furthermore, edge cloudlets are required to support higher mobility, resource limitation, lower latency, and higher connectivity needs of IoMT [13].

To deal with the privacy challenges of IoMT, authentication techniques have been widely used. These techniques are broadly based on agreed-upon decision-making using session keys [14]. Encrypting data and sharing session keys is a feasible choice to secure IoMT [15]. However, it increases communication overhead and computation complexity. To deal with privacy issues, authenticated key agreement (AKA) has been proposed. However, AKA protocol creates a consensus between entities and generates common session keys. It also suffers from deployment challenges over multiple edge cloudlets [16] and mobile crowdsourcing issues [17–20]. Additionally, available techniques suffer from single point of failure issues, unnecessary delays, and weaker defense against attacks [14]. Moreover, IoMT incorporates distributed infrastructure, which increases additional deployment issues. Therefore, it is challenging to provide centralized security mechanisms [21]. These limitations instigate the use of distributed blockchain to ensure the privacy of IoMT [22].

Distributed blockchain ensures the security and privacy of the transactions [23]. One of the most popular implementations of secure blockchain is the cryptocurrency to generate bitcoins. Blockchain manages a ledger where all transactions are verified and signed from the time of creation. The transaction record is always present on all the blockchain nodes making it harder to maliciously modify a transaction. Every node in the blockchain stores some part of the transaction which makes it reliable [24]. Trusted nodes coordinate with each other to authenticate transactions ensuring the privacy of the overall network infrastructure [25]. Blockchain records are safe from unauthorized access because only authenticated nodes have access to the blockchain resources [26]. A new entry in the blockchain is stored on top of the previous blocks. It creates a chaining impact where any malicious activity can be easily detected. If an adversary tries to alter a record, all blockchain nodes analyze their blocks and ultimately detect the location of the intervention [27].

Some previous techniques use blockchain for privacy enforcement of IoMT. Pan et al. [28] propose to alleviate the hacking risk of IoMT devices and efficient management of smart contracts. However, this scheme suffers from increased latency during authentication. Karmakar et al. [29] propose a privacy framework for IoMT. They use a centralized authentication mechanism using encryption keys. However, it suffers from a single point of failure issues. Makhdoom et al. [30] propose a secure data transmission strategy between IoMT and edge cloudlets. Every operation on the public blockchain is stored as a transaction. Since the blockchain uses a distributed strategy to store records where every participant is able to perform the integrity check, they are secured as they are added to the blockchain. Although these approaches utilize blockchain for IoMT privacy, they suffer from scalability issues when deployed in large-scale IoMT. Moreover, using blockchain with IoMT raises the challenge of storing an authenticated local copy of the current ledger in the blockchain that has not been considered in the previous techniques.

Figure 1 shows blockchain implementation on edge cloudlets. Keeping in view the privacy challenges of IoMT, characteristics of blockchain with smart contracts, and to overcome limitations of existing techniques, we intend to deploy distributed blockchain under the centralized management of SDN [31]. The combination of IoMT, SDN, and blockchain ensures a higher level of security and privacy. An IoMT device that needs to offload its data to an edge cloudlet is authenticated using blockchain. Every blockchain in the vicinity of the IoMT device stores a block of the blockchain that authenticates the device. Once the device is authenticated, it can start offloading data and securely access edge resources. We propose a blockchain-based framework (i.e., SecureMed) for the security and privacy-preserving of IoMT. We develop an algorithm to authenticate IoMT using public/private key consensus. This algorithm is implemented on the edge cloudlets which authenticates IoMT. SecureMed provides a privacy-preserving environment for the IoMT to access network resources. The contributions of this paper are as follows.

(i) We propose a blockchain-based framework (i.e., SecureMed) for privacy-preserving in IoMT

(ii) We design an authentication-enforcement algorithm based on a mutual consensus of public/private keys to ensure the privacy of IoMT

(iii) We perform experimentation using Pythereum which is Python-based Ethereum implementation, mininet, and floodlight controller to evaluate SecureMed

This paper is organized as follows. Section 2 presents the related work. Section 3 discusses the blockchain mechanism for IoMT privacy including signature verification algorithms. Section 4 discusses the system model of SecureMed. Section 5 discusses the authentication mechanism carried out on edge cloudlets. Section 6 discusses the evaluation of the proposed framework while Section 7 summarizes this research and provides future research directions.

2. Related Work

IoMT sensors and actuators collect real-world medical data which is used for autonomous service provisioning in healthcare [32]. Adversaries can exploit IoMT capability to

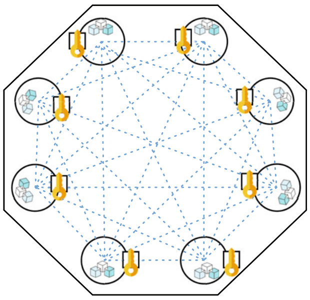

Figure 1: An example of devices, where two adversaries are shown attacking the infrastructure.

generate and transfer massive amounts of data to generate devastating attacks [33]. IoMT users are mostly unaware of the secure usage of smart objects. For example, most of the time, the default password of an IoMT device is not changed. The Mirai botnet is one of the popular examples of the vulnerabilities of IoMT [34]. Privacy-enforcement solutions for IoMT have been devised by numerous previous studies [15]. For an effective understanding of the previous work, we have divided the existing literature into authentication approaches, privacy-preserving at edge cloudlets, and hash-based approaches.

2.1. Authentication Approaches. Garg et al. [36] propose an authentication scheme by transmitting security keys over the network. They use security managers to secure IoMT data. Authentication keys are securely transmitted that are decrypted upon receiving at the destination. Blockchain is used for distributed key management. A dynamic transaction collection period helps in eliminating longer key transfer time during handover. Kumar et al. [40] propose a blockchain-based method for IoMT security against privacy leakage and data theft attacks. They use a public blockchain to ensure trustworthiness using authentication keys. Karmakar et al. [29] propose a security framework for IoMT. They use a centralized authentication mechanism using encryption keys. Storing an authenticated local copy of the current ledger in the blockchain becomes challenging. Therefore, IoMT should be equipped with lightweight solutions to download the blockchain's authentication information. However, solution installation on the resource-limited IoMT is challenging. In this regard, Danzi et al. [41] propose a method to aggregate the authenticated blockchain data after different intervals which alleviates the communication cost. Although authentication approaches provide better results, they create extra overhead of data aggregation and add latency to the overall performance of the system.

2.2. Privacy Preserving at the Edge. Edge computing provides storage capabilities to the resource-limited IoMT [35]. However, utilizing the centralized cloud becomes challenging as it increases the latency of transferring data to and from the cloud. In this situation, edge cloudlets provide computation resources at the edge of IoMT and eliminate the overhead of transferring computations to the central cloud. Edge cloudlets decrease latency, reduce extra bandwidth consumption, and alleviate battery power usage. Pan et al. [28] propose to alleviate hacking risks of IoMT devices by efficient management of smart contracts. Edge cloudlets are linked with IoMT devices using a secure blockchain currency/coin system which creates a secure ecosystem for service provisioning. IoMT produce highly sophisticated data related to state information, management, and control [39]. It is crucial for real-time decision-making and any malicious activity can put human lives in danger. It can be exploited by hackers maliciously in the absence of security mechanisms [27]. Makhdoom et al. [30] propose a secure data transmission strategy between IoMT and edge cloudlets. Data is stored only on authenticated nodes. Every operation on the public blockchain is stored as a transaction. Since the blockchain uses a distributed strategy to store records, every participant is able to perform an integrity check. Hence, they are secured as they are added to the blockchain. Although these approaches utilize edge computing for IoMT privacy, they suffer from scalability issues when deployed in large-scale IoMT.

2.3. Hash-Based Approaches. Xu et al. [42] propose a nonrepudiation scheme for service provisioning. They use blockchain for service publishing and authentication. They propose a homomorphic-hash-oriented method that ensures on-chain evidence. Rahman et al. [43] propose a secure hash-based framework using blockchain. They use a smart artificial intelligence-based hash to process data and to store transactions on the edge using the authentication mechanism provided by the blockchain. Ali et al. [26] secure transaction record management using blockchain. They store multimedia and high-end data off-chain to implement shared economy services. However, data processing using artificial intelligence services adds complexity and reduces overall system performance. Moreover, the classification of compute-intensive and non-compute-intensive tasks adds a higher amount of overhead to the system.

We conclude that the fundamental problems of IoMT privacy have not been solved or critically explained in the existing literature. There is still a lack of research that uses distributed blockchain to enforce the privacy of IoMT under the centralized governance of SDN. IoMT includes miniature sensors, wearable devices, and smart pills that need to interact with localized edge networks to perform transactions. Data offloaded by IoMT to the edge could be attacked by adversaries creating higher-security risks that have not been adequately explored in previous studies. Additionally, the current literature also suffers from scope issues. Available solutions only consider one part of the problem of securing a specific aspect of IoMT. There is a lack of literature available that considers the whole IoMT ecosystem for security. Therefore, keeping in view the above discussion, there is still a need to improve the privacy of IoMT using a blockchain-based decentralized storage mechanism. Table 1 shows a comparative analysis of SecureMed, which demonstrates that it has better characteristics including preserving privacy, security, and scalability for the large-scale deployment than the compared techniques. SecureMed provides a solution for the privacy and security challenges of IoMT using SDN, edge computing, and blockchain. It is lightweight, secures overall IoMT, and is scalable for large-scale healthcare services provisioning.

Table 1: Comparative analysis of SecureMed with the available solutions.

Reference	Blockchain	SDN	IoMT	Privacy	Security	Scalability	Authentication
Mukherjee et al. [35]	✓	×	×	✓	✓	×	✓
Garg et al. [36]	✓	✓	×	✓	×	×	×
Rakovic et al. [37]	✓	✓	×	×	×	✓	×
Pan et al. [28]	×	✓	×	×	×	×	√
Yuan et al. [38]	✓	×	×	✓	✓	×	×
Mohanty et al. [27]	✓	×	×	✓	✓	×	✓
Zhong et al. [39]	✓	×	×	✓	✓	✓	✓
SecureMed	✓	✓	✓	✓	✓	✓	✓

3. Blockchain for IoMT Security

Blockchain could be categorized into public and private networks. A public blockchain is a peer-to-peer (P2P) decentralized network that has no private ownership and a consensus is developed among all the participants to operate under a common goal. A private blockchain has more control and agreed-upon regulations to operate. The private blockchain is considered centralized with extended privacy. Each participant in the blockchain approves, maintains, and updates incoming traffic. All members ensure that records are kept in order which invokes a higher level of data privacy and validity.

Figure 2 shows the block structure of the blockchain. Nodes in the blockchain comprise endpoint workstations that contain an entire copy of the ledger that is maintained and updated independently. The most vital aspect of blockchain is the transaction which consists of a record in the blockchain. The block is a structure to record the transactions that are distributed in the whole network where the sequence of blocks is denoted as the chain. Every block has a hash of the current block and a hash of the previous which contains the data. Nodes that perform block verification before adding it to the blockchain structure are denoted as miners. Moreover, every blockchain block contains rules to secure the network. Table 2 shows the notations used in this paper.

Every block in the blockchain contains a block-hash which is generated using cryptography algorithms like SHA-256 [44]. Whenever a block is created, the hash is automatically added to it. Malicious attempts to alter the block provoke incorrect information in the subsequent blocks which renders the whole blockchain erroneous. A proof-of-work is used to slow down the block creation process where miners require a reward. Every joining node in the block gets a complete copy of the blockchain system where the existing nodes verify the information in the newly created block to synchronize with the new blockchain status.

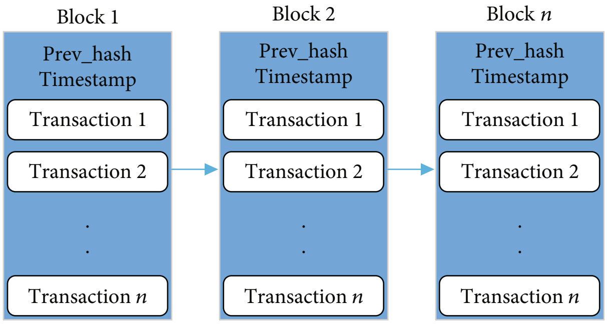

Figure 2: The configuration of blocks in the blockchain.

The elliptic curve digital signature algorithm (ECDSA) employs the elliptic curve (computed using (1)) and a finite field to create a signature such that the other party can verify it [45]. However, the user who verifies the transaction using signatures needs to verify it again. Blockchain uses public and private keys for authentication. The pubic key depends on the "order" which is computed by the total number of instances when a number is continuously added to itself to a level when its slope becomes vertical. Hence, the public key consists of a number between 1 and the "order." Moreover, the public key is computed by employing the scalar multiplication formula as given in (2) where the base point falls on the elliptic curve. The selection of base points is performed in a way that the "order" denotes a higher prime value.

3.1. Signature Verification Using Private Key. A string of characters comprising letters and digits is used to denote the address. For instance, "2DzYBRO6YiFulZbCiBNZB-DICTBEno6a9." Private and public keys are present at every node that is chosen using the ECDSA algorithm. The signature is computed prior to the computation of the private as well as public keys using Algorithm 1.

$$y^2 = x^3 + ax + b, \tag{1}$$

$$Q = d \times G. \tag{2}$$

3.2. Signature Verification Using Public Key. Steps given in Algorithm 2 are used to verify the signatures.

The steps given in Algorithms 1 and 2 demonstrate the computation of signatures using the private key and subsequently verifying it using the public key.

3.3. The Working Principle of Blockchain. A block in the blockchain consists of the block header and body fields. Every block header in the block contains Merkle root, a hash function of the previous block, data, and timestamp. The blockchain contains three blocks connected sequentially. The body contains transaction data of the blockchain where

Table 2: Notations used in the formulation of blockchain.

Symbol	Explanation
G	Base point
n	Order
d	Private key
Q	Public key
k	A random integer
x	The x-axis point on the elliptic curve
y	The y-axis point on the elliptic curve
a	A random real number
b	A random real number
t_i	The i_{th} blockchain transaction
auth_{aj}	Validation certificate
v_i	The i_{th} IoMT
pk_{vi}	Public key of authenticator
sk_{mj}	Private key of authenticator
pubkey_{vi}	Public key of the requesting node
T_{pealert}	Notification for blockchain
T_{admit}	Admit transaction
T_{del}	Delete transaction
T_{permit}	Assign permission to IoMT

the header connects to the chains of the blocks. Timestamp inspects the time when the block was created whereas Merkle root ensures the transaction integrity which may change if the transactions of the previous block were configured maliciously. Any change in the block entry causes changes in the hash function; the impact of which is propagated sequentially forming a fork. Changed blocks are not aligned with the status of all the other blocks which identify the exact location where the block was changed. Hence, data is secured from adversaries.

3.4. Attack Scenario. IoMT contains attack vulnerabilities due to lack of computation resources (e.g., battery, storage, and processing). Attackers can utilize the resource-limited nature of IoMT and launch devastating attacks such as man-in-the-middle, link flooding, DDoS, and privacy-invading attacks. IoMT utilizes autonomous transactions for data collection, transfer, and storage where any adversary can maliciously modify transactions causing huge data and revenue loss. Hackers maliciously impersonate and inflict fake data, spoof transactions, and sniff data without identity reveal. Any attack on the healthcare control system can cause severe consequences including threats to human lives, medical diagnosis, and treatment discrepancies.

4. System Model

SDN has been widely used in controlling heterogeneous devices. SDN provides flexibility to control network infrastructure using a centralized controller. The controller has a centralized view of the network; therefore, it can make efficient routing decisions and direct traffic towards desired paths [46]. SecureMed can fulfill the heterogeneous demands of IoMT for traffic routing using optimized paths. It can direct offloading traffic from IoMT to the nearest edge cloudlets. The architecture of SecureMed is shown in Figure 3. The system comprises three layers including, application, control, and infrastructure layer. Edge cloudlets and IoMT reside at the infrastructure layer which also contains forwarding devices including routers, switches, and gateways. Blockchain is implemented at the edge cloudlets as it contains sufficient resources to handle blockchain transactions. Moreover, it is challenging to provide central cloud services at lower latency. Therefore, edge cloudlets can provide offloading services to the devices where each cloudlet contains a block of the overall blockchain. SecureMed will be implemented as an application at the application layer of SDN. Figure 3 shows the SecureMed application at the application layer of the architecture. It could be observed from the figure that the blockchain has been shown at the edge cloudlet layer. The bottom layer contains the IoMT devices. The IoMT devices interact with the blockchain using smart contracts which authenticate them and allow computation on edge cloudlets. The detailed workflow of SecureMed is described in the next section.

4.1. SDN-Oriented IoMT. We utilize the centralized management characteristics of SDN to efficiently manage network traffic and implement requirements of security and privacy. The dynamic flow rule installation strategy of SDN makes it an efficient choice to manage the IoMT network. SDN enables the programmability of IoMT which can be utilized to manage the network resources in a desirable way. IoMT devices merely act as the forwarding components in the network. Different flow rule installation strategies are used including reactive, proactive, and hybrid. The proactive strategy installs flow rules at the network infrastructure prior to the arrival of traffic flows, whereas the reactive strategy installs the flow rules after the arrival of the flows. Both strategies have certain advantages and drawbacks; hence, the hybrid strategy is adopted by most of the networks. In this strategy, some flow rules are proactively installed, whereas other traffic is handled upon the arrival of the flows reactively.

We use public/private blockchain to ensure the security and privacy of IoMT implemented on the edge cloudlets. Edge cloudlets create, append, and delete blocks based on the rules of the proposed framework. Edge cloudlets are responsible to act as authenticating nodes that decide whether to allow or block an IoMT transaction.

4.1.1. Application Layer. This layer contains services to manage the IoMT network. Customized applications can be developed to control network traffic in a desirable way. This plane contains blockchain management services including configuration, hash, key, and transaction management. This layer communicates with the control layer using the representational state transfer (REST) application programming interface (API). It facilitates application development to

```
Require: variable $t$, $l$, $w$, $n$, $u$, $v$, $G$, $Q$, $x$.
Ensure: verification status of signatures
1:  An integer $k$ is selected having the value from the range of 1 and $n - 1$.
2:  Scalar multiplication is performed to compute the point $(x, y) = k \times G$.
3:  Compute the value of $t = x|n$.
4:  if $t == 0$ then
5:      go to step 1.
6:  else if $l == 0$.
7:      go to step 1.
8:  end if
9:  Compute the value of $l = (z + t \times d)/k|n$.
10: return Extract the value of the blockchain signature $(t, l)$.
```

Algorithm 1: Computing the blockchain signature.

```
Require: variable $t$, $l$, $w$, $n$, $u$, $v$, $G$, $Q$, $x$.
Ensure: verification status of signatures
Ensure variables $t$ and $l$ contain the values between 1 and $n - 1$.
Compute the value $w = l^{-1}|n$.
Compute the value $u = z \times w|n$
Compute the value $v = t \times w|n$.
Compute value of the point $(x, y) = uG + vQ$.
if $t = x|n$ == true. then
    Signature==valid.
else if $t = x|n$!== true.
    Signature == invalid
end if
return Validity status of the transaction
```

Algorithm 2: Signature verification using blockchain.

control the blockchain in a user-defined way. Customized blockchain hashes can be developed at the application layer which enables granular control over IoMT. This layer acts as the platform for the customized applications to effectively provide services from the heterogeneous IoMT. These services include security, privacy, reliability, mobility, and virtualization. Network managers can build applications by acquiring requirements from the network administrators. These applications can guide the controller to perform actions on the IoMT transaction using REST API.

4.1.2. Control Layer. The controller is the brain of SecureMed which deals with the intelligence of the whole network. Policies of the overall network are maintained at this plane where a central database ensures the placement of nodes and path selection for IoMT transactions. The controller implements virtual instances according to the size of the underlying network. These instances are devised to serve a maximum number of forwarding devices and to enforce Quality of Service (QoS) requirements. Additionally, distributed controllers are used to enforce fault tolerance and to avoid the single point of failure of a central controller. SecureMed uses southbound OpenFlow protocol to guide network traffic according to the flow rules. The controller has a bird-eye view of the whole network including knowledge of traffic paths. Therefore, the controller forwards the network traffic effectively. Actual decision-making to implement network policies, cloudlet accessibility, and security of the blockchain is implemented on this layer. Smart devices need to take spontaneous decisions for traffic forwarding or implementing network policies that are governed by the controller. Multiple controllers are used to provide seamless services to IoMT where the network operating system (NOS) employs a hypervisor that performs slicing of the network components for virtualization. NOS creates virtual instances of the physical controller and places VMs at the remote controllers.

4.1.3. Infrastructure Layer. This layer contains physical hardware including switches, wearables, ingestion devices, and edge cloudlets. Different communication technologies are used at this plane to provide services where P2P is employed by the IoMT to communicate. Base stations and edge cloudlets use IEEE 802.11 for interaction. A long-range communication protocol, namely, LTE-A, is employed by the cloudlets to communicate with the global controller. This layer communicates with the edge cloudlets and IoMT devices for seamless privacy enforcement. Here, edge cloudlets provide offloading capabilities to the resource-limited IoMT including smart pills in body sensors and wearables. The infrastructure layer comprises two sublayers including the edge cloudlet layer and the device layer. Edge cloudlets

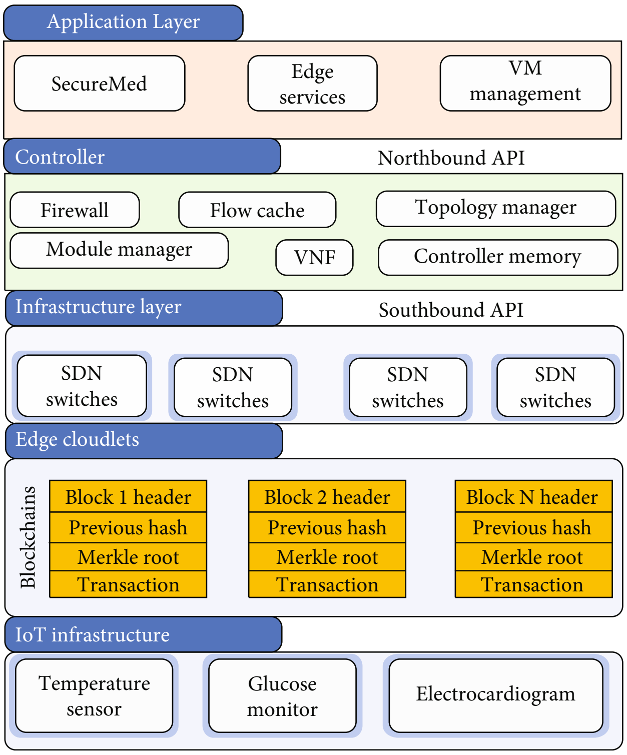

Figure 3: The basic system architecture.

are employed to perform resource-intensive computing tasks as well as implement blockchain blocks. Each cloudlet acts as a block in the overall blockchain to ensure the privacy of IoMT.

4.1.4. Blockchain on Edge Cloudlets. Blockchain acts as a distributed database that stores an incremental list of blocks connected together to maintain a distributed ledger. IoMT interacts with the blockchain using smart contracts. However, it creates more overhead in solving computationally complex problems in the blockchain. For example, the gas consumption during Ethereum smart contract deployment increases with the number of smart contracts. Therefore, we execute smart contract deployment at the edge cloudlets. The public key of IoMT establishes privacy where transactions are identified by the public key and compared with the corresponding private key maintained in the authenticating nodes. If both keys are matched, the transaction is successfully executed. The responsibility of authentication is given to the edge cloudlets. Authenticating nodes identify the transaction and broadcast the public key throughout the network. The transaction is successfully executed upon matching the public/private keys. Any IoMT requesting to join the network sends a request to the blockchain that is forwarded to the authenticating nodes. Edge cloudlets act as the authenticating nodes that ensure the authenticity of the IoMT based on the registration ID of the IoMT. Every authenticating node acts as a block in the blockchain to ensure trust. These nodes act as data trustees which saves the data and blocks malicious traffic from adversaries.

4.2. Working Principle. Figure 4 shows system configuration along with the detailed representation of the components of SecureMed. Edge cloudlets provide continuous resources to the resource-limited devices to perform compute-intensive tasks. Each cloudlet is managed by a distributed controller. Each distributed controller is managed by the central SDN controller. Seamless and secure service is implemented using SecureMed. Blockchain is incorporated with edge cloudlets providing internal security. Cloudlet resources are securely provided to the requesting IoMT. Due to resource limitations, mining is performed on edge cloudlets. A device's past behavior, resource consumption, and abnormal behavior are considered to enforce privacy. If any device is involved in the aforementioned malicious activities, it is added to the blocklist that is continuously monitored to ensure privacy. Data of all devices in the network is stored on the blockchain which ensures continuous surveillance of the whole network. Moreover, cloudlets employ VMs to accomplish service requests from resource-limited IoMT devices. The workflow of SecureMed is described in steps 1 through 5 in the following. During step 1, an IoMT device interacts with the blockchain using the smart contract through an SDN IoMT gateway. In step 2, the IoMT device also sends verification/validation requests to the edge cloudlets. Upon receiving the request, the edge cloudlets interact with the blockchain using a smart contract to verify the received request from the IoMT device using step 3. Blockchain transactions are immutable; therefore, blockchain compares both the transactions (i.e., one submitted by IoMT and the second submitted by edge cloudlets) in step 4. If both transactions are verified, the edge cloudlets perform the computation and send the results to the medical application servers, which use this information for the relevant medical tasks.

4.2.1. Request/Response Mechanism. The request-response mechanism of SecureMed offers a secure transaction system for IoMT. IoMT requests a resource from the cloudlet and the request is transferred to the blockchain which ensures data integrity using the public key. Edge cloudlets then analyze the relevant private key associated with the public key. If both public and private keys are matched, the transaction is successfully conducted. Results are transferred to the requesting IoMT after verification. Additionally, the response is also generated in the same manner which is first authenticated by the blockchain in the same manner as carried out during the request phase. The response is then delivered back to the requesting station when a device is validated.

5. Authentication Mechanism

Public key infrastructure- (PKI-) based authentication techniques use a centralized certification authority. It suffers from a single point of failure, lack of fault tolerance, and reliability issues. To provide a better strategy, we propose a decentralized approach using blockchain. This technique utilizes the public key of the IoMT to ensure privacy. To safeguard the whole ecosystem, verification of IoMT identity, its validity, and a technique to revoke the public key

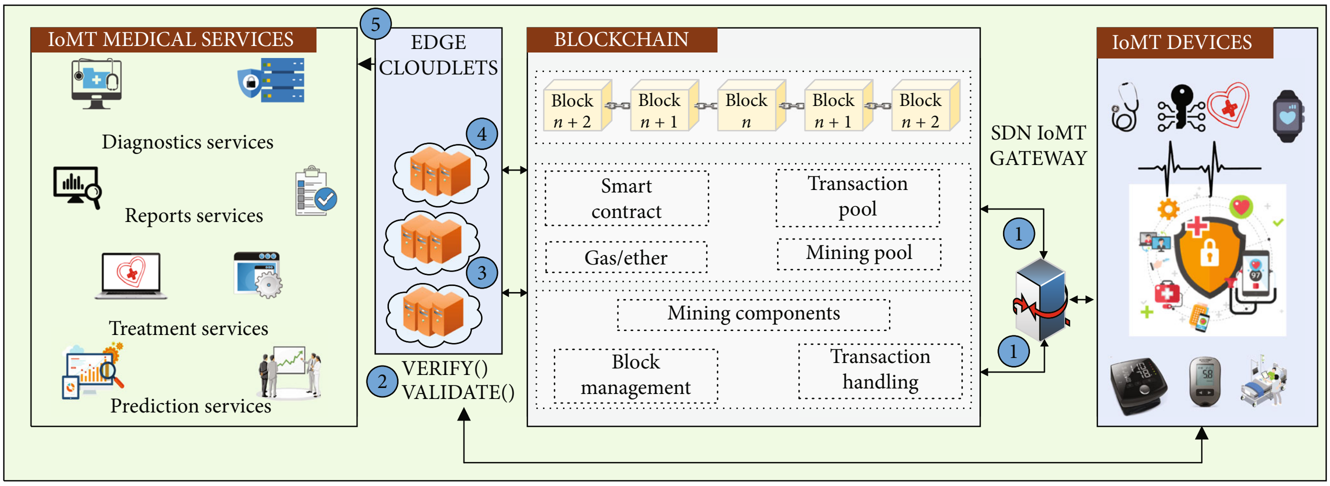

Figure 4: A workflow of SecureMed starting by IoMT data collection to the provisioning of services.

must be established. Due to its decentralized nature, blockchain eliminates the overhead of third-party validation for security. Thus, single point of failure could be eliminated which makes the system dependable and reliable. Information about IoMT admission or revocation is transferred to the permissioned blockchain nodes denoted as the authenticating nodes. Transaction authentication is enforced on the edge cloudlets as they are widely deployed around IoMT. Moreover, there is a central controller that analyzes the transaction's credentials and permits the transaction in the network. Once the transaction is authenticated, public and private keys of the transaction are employed to provide permission to IoMT. Additionally, authenticating nodes have the authority to permit or reject the transaction. In case an IoMT behaves in an unprecedented way, the controller forwards a notification to the blockchain, and action is performed by the authenticating nodes according to the predefined rules. The admission/revocation information stored in the blockchain is utilized by IoMT to authenticate each other without posing a major overhead. Algorithm 3 shows steps to ensure authentication.

IoMT devices are denoted by $(v_1,v_2,v_3\ldots,v_n)$, and every IoMT joining the network must generate a public/private key. The public key is available to other IoMT, and the private key is secretly stored. The public key is used for message exchange and to maintain the integrity of communication. It is also utilized to identify whether an IoMT is a member of the blockchain or not. Authenticating nodes allocate the validation certificate denoted as the auth_{aj} for the IoMT that it is legal. This certificate is delivered to the blockchain using a registration transaction that follows the format given in

$$\text{Certificate} = \text{certID}, \text{regID}, \text{sig}\left(\text{auth}_{aj}, \text{cert}\right). \qquad (3)$$

5.1. IoMT Device Admission. There are multiple edge-based authenticating nodes in the network that deploy Algorithm 3 to reach a consensus in the network. An admission request from the controller is accepted based on the source node's authentication status. If the number of certificates reaches 10 for a particular IoMT, one of the authenticating nodes declares the public key of v_i valid. The admission transaction is given in (4). The controller analyzes the authenticity of the IoMT before inserting its ID into its local blockchain.

$$\text{Admission} = \text{pubkey}_{vi}, \text{auth}, \text{sig}\left(\text{sk}_{mj}, \text{pk}_{vi}\right), \qquad (4)$$

5.2. IoMT Device Authentication. While v_i is added to the blockchain, it can communicate with other IoMT and share critical information. However, to ensure the integrity of the messages, v_i needs to sign the message using its private key. Therefore, every message receiving IoMT authenticates the sender by inspecting the public key of v_i. The public key is analyzed whether it is aligned with the private key that was used to assign the message before sending it. This is different from the traditional PKI-based systems where the sender and receiver need to maintain a digital certificate [47]. This has been managed by a lookup function in SecureMed which is faster compared to traditional cryptography solutions.

5.3. IoMT Device Revocation. For the revocation process, every IoMT v_j detecting the misbehavior of v_i notifies the blockchain. The format of the misbehavior notification is given in (5). In (5), pubkey_{vi} denotes the public key of the malicious IoMT while sk_{vj} denotes the private key of the IoMT that has reported the misbehavior. Authenticating nodes accept the request and add the ID to the blockchain only if they originated from a valid IoMT. These transactions are further considered to revoke an IoMT or not.

$$\text{Revocation} = \text{pubkey}_{vi}, \text{misbehavior}, \text{sig}\left(\text{sk}_{vj}, \text{pk}_{vi}\right). \qquad (5)$$

A distributed revocation protocol is executed by the authenticating nodes. It is based on business rules set and enforced by authenticators. Authenticators decide to revoke an IoMT device v_i only if they receive more than $n = 10$ authenticated misbehavior transactions for v_i. When v_i is ready to be revoked, one of the authenticating nodes

```
Require: set of IoMT $\{v_1, v_2, v_3,.,.,.,v_m\}$, a set of authenticating nodes $\{e_1, e_2, e_3,.,.,.,e_n\}$
Ensure: authenticate transactions in IoMT
for ∀ requesting IoMT $\in\{v_1, v_2, v_3,.,.,.,v_n\}$ do
    Generate (pubKey, pvtKey) for $v_i$
    Broadcast pubkey in IoMT
    Authorize $v_i$ based on $v_{\text{rec}}$
    $e_i$ = push $t_{\text{permit}}$ to blockchain <pubKeyIsValid, app(pubKey$_{aj}$, pubKeyIsValid$_{vi}$) >
end for
for ∀ the transactions $\in\{t_1, t_2, t_3,.,.,.,t_n\}$ do
    if src$_{t_i}$ == src$_{e_j}$ && $t_i$ == 10% transactions then
        Check pubKeyIsValid == True
          Generate $T_{\text{permit}}$
      end if
      Broadcast pubkey in IoMT
      $e_i$ = push $T_{\text{admit}}$ to blockchain <pubKeyIsValid, app(pubKey$_{aj}$, pubKeyIsValid$_{v_i}$) >
      end for
  for ∀ IoMT $\in\{v_1, v_2, v_3,.,.,.,v_n\}$ do
      Check pvtKey is present for pubKey
      Send message to $v_i$ using its pubKey to validate
      if misbehavior == True && $T_{\text{alert}}$==10% transactions then
          Generate $T_{\text{alert}}$ to the blockchain <pvtKey$_{v_j}$, pubKey$_{v_i}$ >
          $T_{\text{del}}$ = <pubKey, deleted, sig(pvtKey$_{v_i}$, pubKey$_{v_i}$) >
      end if
  end for
  return Transaction $T_{\text{admit}}$, $T_{\text{del}}$
```

Algorithm 3: IoMT privacy enforcement using blockchain.

broadcasts a message to the network given in (6). Here, pk_{vi} and sk_{mj} are the public and private keys of the authenticators. Once the revocation request is received, other authenticators add it to the blockchain after analyzing the authenticity of its source.

$$\text{Broadcast} = \text{pubkey}_{vi}, \text{revoked}, \text{sig}\left(\text{sk}_{mj}, \text{pk}_{vi}\right). \tag{6}$$

6. Experimental Evaluation

We perform extensive experiments to demonstrate the efficacy of the proposed framework. The comparison metrics, network model, experimental setup, and results are given in this section. In this section, we also discuss the experimental setup and performance evaluation of SecureMed. Table 3 shows experimental parameters used to evaluate SecureMed. The experimental setup contains 30 IoMT devices generating requests spread over 500 m^2 area. A request is generated when an IoMT device sends a computation request to the edge cloudlets. The request is first transferred to the blockchain using a smart contract. Upon successful authentication, the request is sent to the nearest edge cloudlet in the vicinity of the IoMT. Figure 5 shows the experimental setup of SecureMed.

6.1. Experimental Setup. All the experiments in this study are based on extensive real-time simulations conducted using a laboratory desktop computer having Intel(R) Xeon(R) CPU E3-1225 v5 @ 3.30 GHz and Windows 10 Enterprise 64-bit OS with 16 GB memory. We consider an IoMT paradigm to evaluate SecureMed for privacy enhancement using mininet-WiFi and floodlight controller to simulate the SDN paradigm. We compare our results with the traditional blockchain approaches. We use the Pythereum tester tool for the traditional and SecureMed blockchain implementation. The reason behind choosing Pythereum was that it provides a lightweight solution for blockchain creation. It works by creating a novel blockchain containing a genesis block while creating a test state by passing a genesis state. Finally, the transaction is processed by exploiting the private key to the given address and data. Mininet-WiFi was running on the VirtualBox-Ubuntu OS, whereas, the floodlight controller was running on the windows machine. The network was remotely connected with the Floodlight OpenFlow controller. We configure each IoMT to generate a request to join the network, which was either accepted or rejected by the central controller based on Algorithm 3. We used 30 IoMT devices to simulate the network, and the simulation area corresponds to 500 m^2.

We used Floodlight open-source controller to simulate the controller in our network, which is a Java-based OpenFlow controller that interacts with the data plane using OpenFlow protocol. Floodlight controller offers flexibility to develop applications using Java programming language to control network traffic desirably. We employed iPerf and Wireshark to measure the time between transactions and performance measurement. IPerf is an open-source tool that can be used to measure network statistics whereas Wireshark can detect different network parameters by

Table 3: Evaluation parameters.

Parameter	Value
Simulation area	500 m^2
Number of IoMT	30
Network controller	Floodlight
Performance parameter measurement	"Wireshark, iPerf"
Traffic simulator	Pythereum
Network simulator	Mininet-WiFi
Traditional blockchain	PoW-based

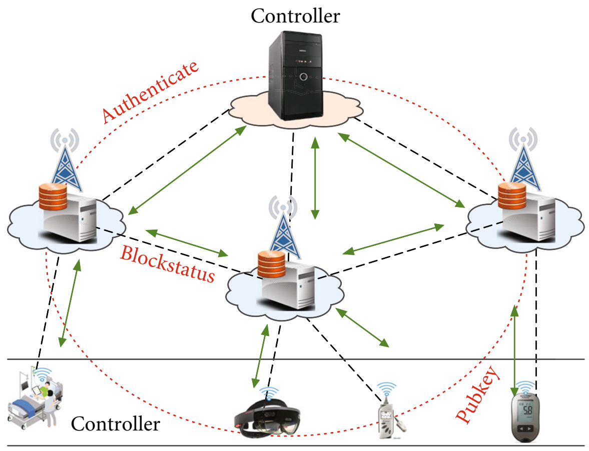

Figure 5: The experimental model showing blockchain implemented on edge cloudlets.

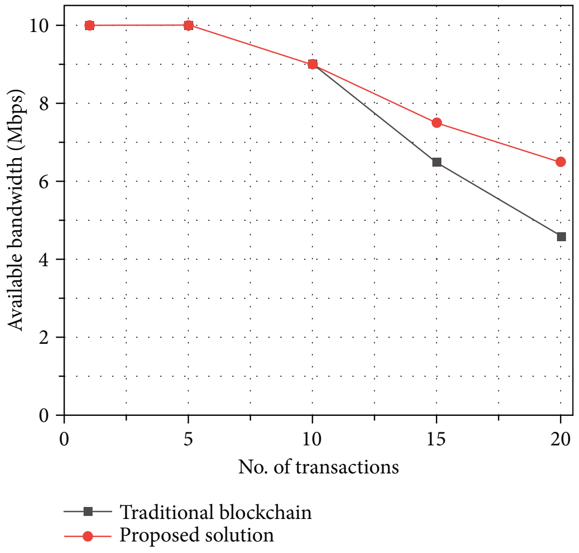

Figure 6: Number of transactions vs. available bandwidth.

monitoring the network traffic at a microscopic level. Ethernet interfaces in mininet can be detected in the Wireshark, which can be used to analyze traffic flows on different paths. Based on the extracted statistics, we draw graphs to demonstrate the effectiveness of SecureMed. We also implemented a traditional blockchain approach and compare the performance of SecureMed, which does not consider the IoMT constraints and poses time overhead during implementation. Traditional blockchain uses proof-of-work as a consensus algorithm to ascertain the validity of transactions and to compute new blocks in the IoMT paradigm. We use average response time, latency, available bandwidth, deployment time, and detection accuracy as performance parameters. The evaluation results are discussed in the next section.

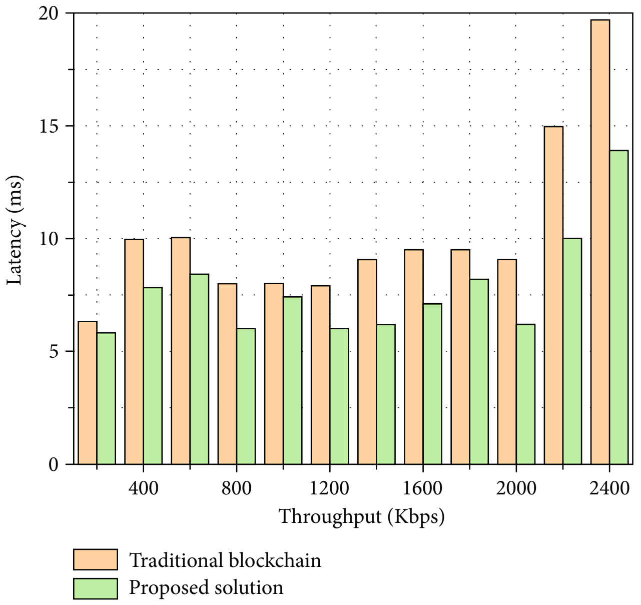

Figure 7: Latency vs. throughput.

6.2. Evaluation and Results. We evaluate SecureMed using real-world parameters and exploited an IoMT paradigm by running the setup and measuring the network statistics that were then used to perform the evaluation. We measure network traffic generated by the interaction of IoMT, edge cloudlets, and the controller during the system execution using Wireshark and iPerf. After gathering the statistics, we perform expert data analysis in Python programming language to measure different statistical parameters that were used for the evaluation. In Wireshark, we separated traffic flows on different links and then measured different evaluation metrics, which were then plotted on the graphs.

6.2.1. Available Bandwidth. Figure 6 shows that the available bandwidth decreases due to the increase in the number of transactions. We measure the available bandwidth using the proposed and traditional blockchain solutions, which can be observed in Figure 6. It shows that the available bandwidth remained the same until 10 transactions for both SecureMed and traditional blockchain. However, it started to decrease continuously with the increase in the number of transactions. The figure shows that at 20 transactions, the available bandwidth using SecureMed was 6.5 Mbps whereas, for the traditional solution, it was around 4.2 Mbps. It shows that SecureMed worked efficiently to consume lesser bandwidth and provided efficient services.

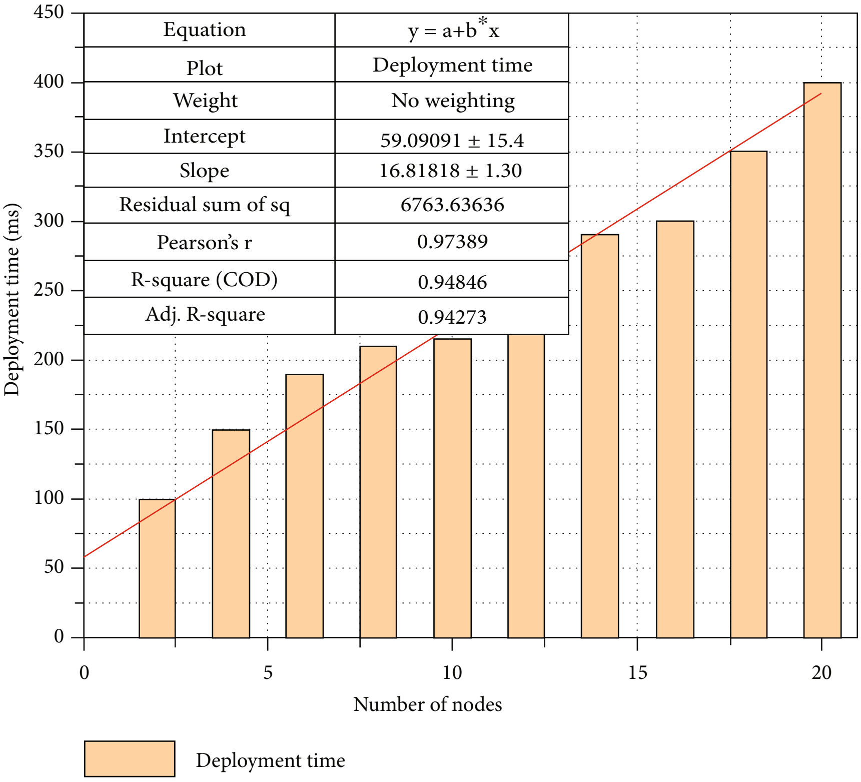

Figure 8: Number of nodes vs. deployment time.

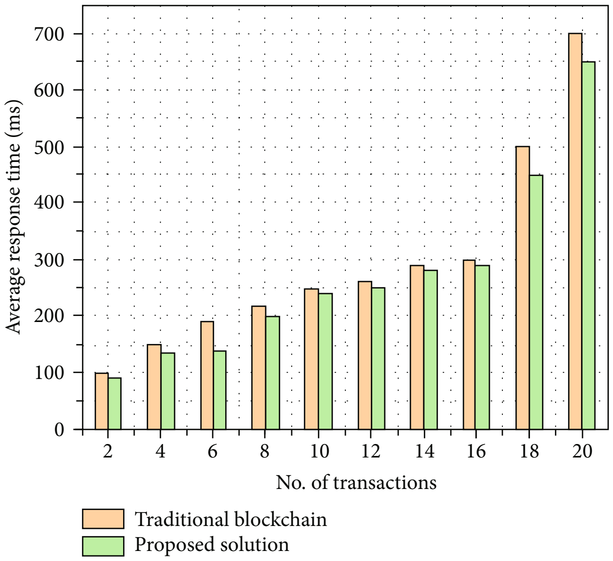

Figure 9: Average response time Vs. number of transactions.

6.2.2. Latency vs. Throughput. We measure latency with reference to the throughput of traditional blockchain and SecureMed. Figure 7 shows that the increase in the network throughput is resulting in an increase in the latency of the system. The latency increased for both traditional blockchain and SecureMed. However, SecureMed showed an improved performance compared with the traditional blockchain. As can be observed from the figure that at 2000 Kbps throughput, the overall latency of SecureMed was around 6.2 ms whereas for the traditional blockchain, the latency was around 9.05 ms. It shows that SecureMed helps in lowering down the latency compared to traditional blockchain.

6.2.3. Deployment Time. This experiment shows that, with the increase in the number of nodes, the deployment time has also been proportionally increased. Figure 8 shows that the deployment time is continuously increasing with the increase in the number of nodes in the network. This linear increase in the deployment time represents that SecureMed is scalable and can be utilized in large-scale paradigms. The miners' block generation process in the blockchain has been performed periodically, which produces a slight overhead on the network.

6.2.4. Response Time. Further, we send packets from one host to another and measure the response time using the proposed and traditional blockchain solutions. Figure 9 shows that the average response time has been increased, which could have been caused by block creation and chaining. However, SecureMed worked efficiently having a lower response time compared to the traditional blockchain. As it could be observed from the figure that at 20 transactions, the response time was around 650 ms, whereas at the same number of transactions, the average response time for the traditional blockchain was around 700 ms. This experiment shows that SecureMed consumes lesser time in packet processing using an enhanced authentication algorithm compared to traditional blockchain.

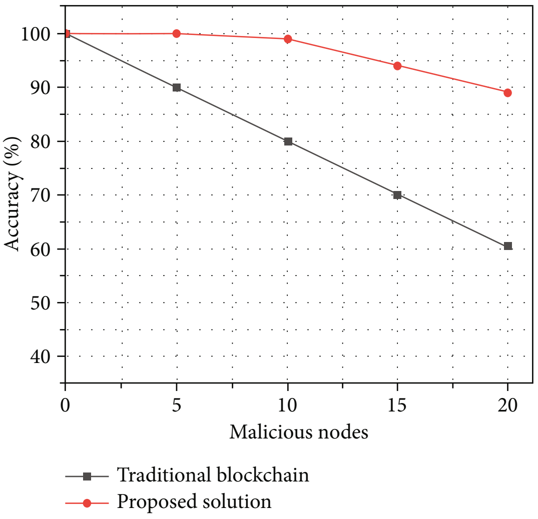

Figure 10: Accuracy vs. number of malicious nodes.

6.2.5. Detection Accuracy. Figure 10 shows the detection accuracy of SecureMed while increasing the adversarial nodes in the network compared to the traditional blockchain scheme. It can be observed that with the increase in the adversarial nodes, the detection accuracy was lowering down. Although the accuracy has a downward trend with the increase in the malicious nodes. However, SecureMed performed better than the traditional blockchain. It can be observed in the figure that the accuracy continuously dropped for the traditional technique. However, the detection accuracy of SecureMed was not deterred even at around 6 malicious nodes, whereas at the same number of malicious nodes, the accuracy was continuously lowering down for the traditional blockchain. This shows that SecureMed significantly enhances attack detection accuracy and offers defense against adversaries.

The experimental evaluation shows that SecureMed provides efficient results in lowering down latency, enhancing detection accuracy, lowering down response time, and efficiently managing bandwidth consumption, compared to traditional blockchain. Moreover, SecureMed is scalable and could be used for large-scale service provisioning.

7. Conclusion and Future Research Direction

IoMT uses sensors and actuators to control healthcare infrastructure autonomously, supported by edge cloudlets to perform computation at the edge. Data produced by IoMT is critical, and any anomaly causes discrepancies in disease diagnosis and treatment. IoMT could be attacked in a variety of ways including MiTM, key fob, relay, information spoofing, data loss, and password theft attacks due to lack of security. We proposed the SecureMed framework for the privacy enforcement by deploying blockchain at the edge of IoMT. In the SecureMed, blockchain offers decentralized privacy by maintaining a privacy-preserving ledger for IoMT. A decentralized trust-management scheme has been used on the edge cloudlets to enforce trust using the previous interactions of IoMT devices with the edge cloudlets. Trusted edge cloudlets implement an authentication algorithm to enable privacy-preserving service provisioning to the IoMT. We postulate that traditional privacy-preserving techniques such as PKI suffer from single point of failure and complex computations. Therefore, SecureMed employs the decentralized blockchain to enforce the privacy of the edge-enabled IoMT. The decentralization using edge cloudlets makes SecureMed fault-tolerant, reliable, and trustworthy for efficient end-to-end IoMT service provisioning. Edge cloudlets are utilized to implement blockchain that acts as authenticators to permit transactions, which provides efficient privacy. We performed a simulation-based evaluation using real-world parameters, and the results demonstrate that SecureMed is efficient in ensuring privacy during IoMT service provisioning.

This work is a step towards efficient end-to-end IoMT service provisioning, ensuring security, and enforcing privacy of services in the heterogeneous IoMT. We are planning to extend this research by performing experiments on a physical testbed. Additionally, implementing SecureMed on different blockchain types and comparing the results to establish the best-performing solution will be an effective future area of research.

Data Availability

Experiments were performed based on simulations, and no publicly available dataset was used.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research has been performed as part of the employment contract mentioned against each author’s name.

References

[1] J. E. da Rosa Tavares and J. L. Victória Barbosa, “Ubiquitous healthcare on smart environments: a systematic mapping study,” *Journal of Ambient Intelligence and Smart*, vol. 12, no. 6, pp. 513–529, 2020.

[2] E. Schwartz, “The number of health care workers the world will be short by 2030,” 2022, January 2021, https://www.projecthope.org/the-global-health-worker-shortage-7-numbers-to-note/02/2020/.

[3] T. Kaneda, “PRB projects 2050 world population at 9.8 billion, youth population to reach 1.4 billion,” 2017, January 2022, https://www.prb.org/2017-world-population-data-sheet/.

[4] Center of Disease Control and Prevention, “Health and Economic Costs of Chronic Diseases,” January 2021, https://www.prb.org/2017-world-population-data-sheet/.

[5] P. D. Waggoner, “Pandemic policymaking,” *Journal of Social Computing*, vol. 2, no. 1, pp. 14–26, 2021.

[6] R. Kumari, S. Kumar, R. C. Poonia et al., “Analysis and predictions of spread, recovery, and death caused by covid-19 in India,” *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 65–75, 2021.

[7] C. Chi, Y. Wang, X. Tong, M. Siddula, and Z. Cai, "Game theory in internet of things: a survey," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12125–12146, 2021.

[8] T. Li, C. Li, J. Luo, and L. Song, "Wireless recommendations for internet of vehicles: recent advances, challenges, and opportunities," *Intelligent and Converged Networks*, vol. 1, no. 1, pp. 1–17, 2020.

[9] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, "SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)," *Computer Communications*, vol. 160, pp. 697–705, 2020.

[10] Q. Huang, Y. Zhou, L. Tao et al., "A chan-vese model based on the Markov chain for unsupervised medical image segmentation," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 833–844, 2021.

[11] F. Wang, M. Zhu, M. Wang et al., "6g-enabled short-term forecasting for large-scale traffic flow in massive iot based on time-aware locality-sensitive hashing," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5321–5331, 2020.

[12] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[13] K. R. Sollins, "Iot big data security and privacy versus innovation," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1628–1635, 2019.

[14] M. Ma, D. He, H. Wang, N. Kumar, and K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.

[15] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new-type of blockchain for secure message exchange in vanet," *Digital Communications and Networks*, vol. 6, no. 2, pp. 177–186, 2019.

[16] Y. Zhou, X. Long, L. Chen, and Z. Yang, "Conditional privacy-preserving authentication and key agreement scheme for roaming services in vanets," *Journal of Information Security and Applications*, vol. 47, pp. 295–301, 2019.

[17] F. Li, Y. Wang, Y. Gao, X. Tong, N. Jiang, and Z. Cai, "Three-party evolutionary game model of stakeholders in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 4, pp. 974–985, 2021.

[18] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 2058–2080, 2021.

[19] W. Wang, Y. Wang, P. Duan, T. Liu, X. Tong, and Z. Cai, "A triple real-time trajectory privacy protection mechanism based on edge computing and blockchain in mobile crowdsourcing," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2022.

[20] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[21] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, "Fast anomaly identification based on multi-aspect data streams for intelligent intrusion detection toward secure industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6503–6511, 2021.

[22] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing iots in distributed blockchain: analysis, requirements and open issues," *Future Generation Computer Systems*, vol. 100, pp. 325–343, 2019.

[23] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: a blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.

[24] C. Hebert and F. D. Cerbo, "Secure blockchain in the enterprise: a methodology," *Pervasive and Mobile Computing*, vol. 59, article 101038, 2019.

[25] J. H. Ryu, P. K. Sharma, J. H. Jo, and J. H. Park, "A blockchain-based decentralized efficient investigation framework for iot digital forensics," *The Journal of Supercomputing*, vol. 75, pp. 4372–4387, 2019.

[26] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *Journal of Systems Architecture*, vol. 99, article 101636, 2019.

[27] S. N. Mohanty, K. Ramya, S. S. Rani et al., "An efficient lightweight integrated blockchain (elib) model for iot security and privacy," *Future Generation Computer Systems*, vol. 102, pp. 1027–1037, 2020.

[28] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "Edgechain: an edge-iot framework and prototype based on blockchain and smart contracts," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4719–4732, 2019.

[29] K. K. Karmakar, V. Varadharajan, U. Tupakula, S. Nepal, and C. Thapa, "Towards a security enhanced virtualised network infrastructure for Internet of medical things (iomt)," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pp. 257–261, Ghent, Belgium, 2020.

[30] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, article 101653, 2020.

[31] C. Catlett, P. Beckman, N. Ferrier et al., "Measuring cities with software-defined sensors," *Journal of Social Computing*, vol. 1, no. 1, pp. 14–27, 2020.

[32] S. Din, A. Paul, and A. Rehman, "5g-enabled hierarchical architecture for software-defined intelligent transportation system," *Computer Networks*, vol. 150, pp. 81–89, 2019.

[33] N. Miloslavskaya and A. Tolstoy, "Internet of things: information security challenges and solutions," *Cluster Computing*, vol. 22, no. 1, pp. 103–119, 2019.

[34] C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[35] A. Mukherjee, P. Deb, D. De, and R. Buyya, "Iot-f2n: an energy-efficient architectural model for iot using femtolet-based fog network," *The Journal of Supercomputing*, vol. 75, no. 11, pp. 7125–7146, 2019.

[36] N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. 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.

[37] V. Rakovic, J. Karamachoski, V. Atanasovski, and L. Gavrilovska, "Blockchain paradigm and Internet of things," *Wireless Personal Communications*, vol. 106, no. 1, pp. 219–235, 2019.

[38] L. Yuan, Q. He, F. Chen et al., "Csedge: enabling collaborative edge storage for multi-access edge computing based on blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1873–1887, 2021.

[39] L. Zhong, Q. Wu, J. Xie, J. Li, and B. Qin, "A secure versatile light payment system based on blockchain," *Future Generation Computer Systems*, vol. 93, pp. 327–337, 2019.

[40] R. Kumar and R. Tripathi, "Towards design and implementation of security and privacy framework for internet of medical things (iomt) by leveraging blockchain and ipfs technology," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 7916–7955, 2021.

[41] P. Danzi, A. E. Kalør, Č. Stefanović, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight iot clients," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, 2019.

[42] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A blockchain-based nonrepudiation network computing service scheme for industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3632–3641, 2019.

[43] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and iot-based cognitive edge framework for sharing economy services in a smart city," *IEEE Access*, vol. 7, pp. 18611–18621, 2019.

[44] H. E. Michail, G. S. Athanasiou, G. Theodoridis, A. Gregoriades, and C. E. Goutis, "Design and implementation of totally-self checking SHA-1 and SHA-256 hash functions' architectures," *Microprocessors and Microsystems*, vol. 45, pp. 227–240, 2016.

[45] Z. Wang, H. Yu, Z. Zhang, J. Piao, and J. Liu, "Ecdsa weak randomness in bitcoin," *Future Generation Computer Systems*, vol. 102, pp. 507–513, 2020.

[46] F. Wang, G. Li, Y. Wang et al., "Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city," in *ACM Transactions on Internet Technology*, Association for Computing Machinery, New York, NY, USA, 2022.

[47] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Misbehavior detection and efficient revocation within vanet," *Journal of Information Security and Applications*, vol. 46, pp. 193–209, 2019.