



Blockchain-enabled secure and efficient data sharing scheme for trust management in healthcare smartphone network

Rati Bhan¹ · Rajendra Pamula¹ · Parvez Faruki² · Jyoti Gajrani³

Accepted: 4 April 2023 / Published online: 26 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The Internet of Medical Things (IoMT) is an extended genre of the Internet of Things (IoT) where the *Things* collaborate to provide remote patient health monitoring, also known as the Internet of Health (IoH). Smartphones and IoMTs are expected to maintain secure and trusted confidential patient record exchange while managing the patient remotely. Healthcare organizations deploy Healthcare Smartphone Networks (HSN) for personal patient data collection and sharing among smartphone users and IoMT nodes. However, attackers gain access to confidential patient data via infected IoMT nodes on the HSN. Additionally, attackers can compromise the entire network via malicious nodes. This article proposes a Hyperledger blockchain-based technique to identify compromised IoMT nodes and safeguard sensitive patient records. Furthermore, the paper presents a Clustered Hierarchical Trust Management System (CHTMS) to block malicious nodes. In addition, the proposal employs Elliptic Curve Cryptography (ECC) to protect sensitive health records and is resilient against Denial-Of-Service (DOS) attacks. Finally, the evaluation results show that integrating blockchains into the HSN system improved detection performance compared to the existing state of the art. Therefore, the simulation results indicate better security and reliability when compared to conventional databases.

Keywords Blockchain · Internet of medical things (IoMT) · Trust management · Smartphone data security · Secure smartphone data sharing · Healthcare smartphone network (HSN)

✉ Rati Bhan
ratibhan.16dp000126@cse.ism.ac.in

Extended author information available on the last page of the article

1 Introduction

The Internet of Things (IoT) comprises physical objects connected to the Internet via embedded sensors (e.g., camera, gyroscope, and GPS) to perceive, interact, and exchange data. In the modern era of remote healthcare via smartphones, the Internet of Medical Things (IoMT) facilitates different healthcare treatment concepts. In addition, the Covid-19 pandemic has strained healthcare, necessitating the development of tools that can help with online treatment. Numerous successful examples of IoMT that promote remote healthcare include smart insulin pens, oxygen/asthma monitors, connected inhalers, Instant monitoring of Uric Acid (UA) [1], immunological detection of HIV. This paper uses medical biochemical analyzers (Hitachi 7600P), an IoMT device controlled by a smartphone, to measure uric acid and blood sugar to treat patients remotely. It enables them to manage and quickly resolve their medical needs when anything goes wrong. Smartphone-compatible wearable devices, such as smartwatches and biosensors, help monitor a patient's body parameters via internet-based IoMT, enabling observation, diagnosis, and treatment.

Smartphones have become ubiquitous for healthcare institutions, reducing communication costs and delays due to their sensing and app capabilities. Several studies show that smartphone apps are behind the rapid expansion of the mobile health industry. Smartphones facilitate patient medical record collection, and transmission [2]. Additionally, a smartphone sensor-enabled app detects certain cardiac illnesses by monitoring and exchanging patient electrocardiogram (ECG) reports with an IoMT environment [3]. Consequently, these devices form what is known as the healthcare smartphone network (HSN).

The IoMT assists in healthcare transformation into a more intelligent generation, offering real-time interaction and machine-to-machine connectivity. Healthcare professionals can receive reports and be alerted to significant deviations in patients' health. The interdependent ecosystem of the IoT enables multiple components to communicate, such as real-time data collection, physical connections, application control, and data analysis. It is possible to improve the healthcare business through IoMT and HSN. Among peer-to-peer and ad-hoc networks, various trust management solutions have been suggested. However, technological flaws are similar to those found in traditional systems open to scrutiny. According to research by Mohammed et al., [4], and Atlantic Council, the number of known security breaches in healthcare institutions is more than double that in other industries. It implies that safeguarding patients' privacy and sensitive data are one of the most significant issues and challenges [5]. It is the level of trust in a healthcare smartphone network that deals with patient information sent from a centrally trusted server to an app or IoMT device. In addition, trust measures a node's competence in providing necessary services in an IoT network dependability. In general, creating trust in a network has several advantages [6], including the following:

1. Data access control depends on assessing the trust value of IoMT devices and their services. However, providing the same through traditional security techniques is very challenging since it does not overcome such situations.

2. Trustworthy routing that protects critical patient information against malicious nodes can be achieved through trust.
3. By ensuring that the interacting IoMT devices are trusted throughout authentication and authorization, trust makes the services' traditional security model resilient.

Clustering techniques allow the trust management for the IoMT nodes in the HSN network to be a cooperative process rather than an individual duty. IoMT is installed in groups (clusters) that appear to work together to process, collect, and convey acquired information rather than independently performing their responsibilities. As a result, developing and maintaining trust in a clustering system offers several benefits. For example, it assists the member nodes in selecting a trustworthy Cluster Head (CH) inside the cluster. The CH would become capable of identifying malicious nodes in the same way. Furthermore, it aids in selecting trusted route neighbors via which a node will transfer patient records to the CH in multihop clustering. Trust management also aids in identifying alternative trusted CH by which the source node will transfer patient records to the Base Station (BS) in inter-cluster information exchange. To our knowledge, only a comprehensive trust management strategy, Belief-based trust evaluation mechanism for wireless sensor networks (BTEM) [7]) presented by Ganeriwal and Srivastava, has been developed for sensor networks. BTEM solutions have several drawbacks, including IoMT's resource constraints. The proposed work implements a lightweight, reliable Cluster-based Hierarchical Trust Management Scheme (CHTMS) to overcome resource constraints.

1.1 Motivation

Healthcare data include private and sensitive patient records, ailment information, diagnosis data, and doctor recommendations. If the adversary gains access to the healthcare data, the patient data's safety, security, and privacy are compromised [8]. Such sensitive information has a high probability of misuse for financial gain [5]. Mohammed et al. [4] studied data breaches in healthcare institutions during the COVID-19 pandemic and reported volumes of patient information leaked and misused by the supported service provider, causing resources crisis such as lack of oxygen and ventilator. Recent study [8, 9] reports alarming misuse of patient health records. Hence, the security of private IoMT nodes in HSN is paramount. Hence, developing an adequate trust mechanism is critical and mandatory in IoMT and HSN.

1.2 Contributions

Traditionally trust management is handled by a single centralized server in healthcare organizations, a single risk factor that can result in failures. The server may be subject to overwhelming traffic or unexpected events. Since blockchain technology has recently grown and become more accepted, it has been discovered to offer a platform for entirely unknown parties to interact without relying on trusted

intermediaries. Our primary focus is on thwarting insider attacks. Therefore, the proposed blockchain-based trust management system protects HSN against insider attacks significantly. The main contributions of the paper are:

1. We propose a novel CHTMS algorithm, a blockchain-enabled Cluster-based Hierarchical Trust Management technique for detecting malicious insider nodes. The HSN nodes examine the transactions and identify malicious nodes(s).
2. Compared to the state of the art, our approach is a lightweight trust management approach that overcomes the issue of limited resources. We compared memory and communication overhead with the state of the art to manifest the same.
3. Compared to typical trust management systems, which focus exclusively on the trust values of individual nodes, CHTMS examines the trust of a cluster of IoMT devices. This approach is more robust and reliable and has the benefit of using less memory to keep trust values at every node.
4. The proposed approach employs Elliptic Curve Cryptography (ECC) to protect the sensitive health records of patients and is resilient against Denial-of-Service (DOS) attacks.

Security researchers have focused much attention on keeping connected medical equipment trustworthy. Medical devices are integrated with security procedures to ensure their protection. For instance, if an IoMT device in HSN is compromised, attackers can use the infected device to exploit additional IoMT devices. Insider attacks are a significant issue for IoMT and HSN due to their distributed architecture. As previously stated, trust-based interaction is a crucial security strategy for preventing insider attacks [10]. These methods rely on a central server to oversee the trust estimation process and decide. However, most IT people in healthcare institutions must be more security experts. In such a scenario, a central server could be a weak point in the failure of the entire system. In short, it is not easy to know whether users trust the central server regarding security. With the rise in popularity and use of blockchain technology, advanced research started into the combination of trust management and blockchains. It is because blockchain technology allows unknown (or untrusted) nodes to exchange data verifiably without the requirement of any trusted intermediary [11]. Our objective throughout this paper is to propose a trust management system based on blockchain for securing the healthcare industry from insider attacks.

The proposed collaborative system combines energy-efficient data collection with safe data exchange across IoMT nodes using the Hyperledger blockchain. Hyperledger successfully maintains a tamper-proof ledger distributed among IoMT nodes without any requirements for a trusted third central entity, ensuring data security and safety when IoMT nodes share data. The proposed approach validates the security and privacy of the proposed HSN against Denial-of-Service (DoS) attacks. HSN conducted a simulated DoS attack to compare the performance of Blockchain vs. Conventional MySQL databases. The Python script is executed to varying degrees of resource depletion by two distinct levels of DoS attacks: Low-Intensity attacks run ten concurrent processes, while High-Intensity

attacks run twenty concurrent processes. The efficiency of this solution is measured using Transaction Failure Ratio (TFR) and Transaction Successful Ratio (TSR) metrics.

The rest of this paper is organized in the subsequent order: Background and related work about IoMT and HSN are given in Sect. 2. The proposed HSN methodology is explained in Sect. 3. In Sect. 4, performance evaluation and results analysis are covered. Finally, we conclude the paper in Sect. 5.

2 Background and related work

Section 2 discusses the HSN design and its work on preventing insider attacks, including various Intrusion Detection System(IDS) and trust planning and management. This section covers the blockchain background, literature review, and other related research works.

2.1 HSN background

Information technology and communication systems are steadily embraced in the healthcare industry, making communication more efficient between patients and healthcare specialists. The smartphone is among the essential gadgets deployed in numerous healthcare organizations, assisting with cost reduction, data management, and outcome control. It also includes several simple applications that allow users to collect real-time data and consult with healthcare experts. For example, Guo et al. [2] unveiled a smartphone-enabled electrochemical biosensing device that enables healthcare specialists to examine patients' medical records and give accurate, personalized therapy. In addition, Yang et al. [3] demonstrated an IoMT that employed smartphones to capture real-time ECG data from patients and deliver it to the appropriate healthcare institutions for review.

As a result, these Internet-enabled smartphones create a developing network platform known as the Healthcare Smartphone Network, which may be considered one form of the IoT [2]. The high-level design of HSNs is depicted in Fig. 1. The former refers to smartphones used within a healthcare institution. In contrast, the latter also includes smartphones used outside the organization and the equipment patients use daily. Furthermore, patients' smartphones connect with local HSNs over the Internet. So, every device in the HSN is viewed as a (network) node, just like in a traditional network. HSNs, in particular, are predicted to deliver several advantages to patients and healthcare practitioners.

1. The proposed HSN allows interaction between patients and doctors remotely so that it can deal with emergency problems on time.
2. The proposed HSN facilitates remote patient health records sent directly to the concerned doctor periodically using IoMT devices connected to a smartphone in a secured channel. So, the Patients' financial costs and communication delays are reduced.

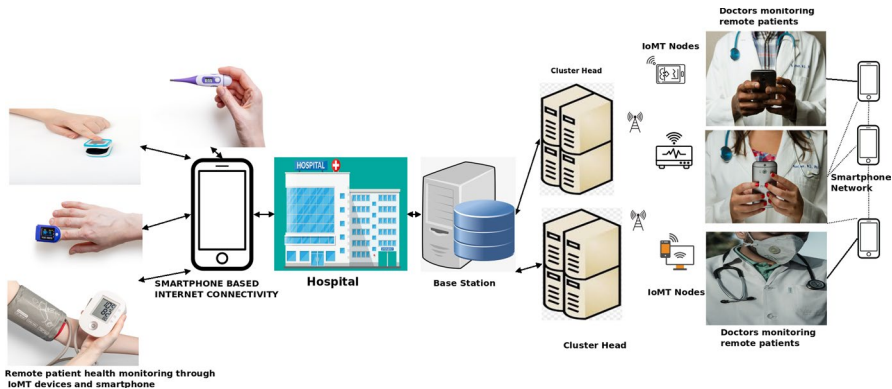


Fig. 1 High-level design of HSNs

3. To improve the management of resources within healthcare organizations, the proposed HSN uses blockchain-based trust management.
4. As a result of the blockchain-based sharing of patient records in a secure channel, we observe that the system continued to perform better under malicious conditions in spite of various insider attacks.

For instance, an attacker may pretend to be a patient and then, attempt to hack an IoMT device in an HSN. The attacker uses the compromised device (IoMT node) to execute other attacks, including spoofing, Eclipse Attacks, Vulnerability Attacks, DoS, and malware distribution. As a result, develop an effective security procedure to fight insider attacks, such as recognizing malicious nodes. In addition, the system is intended to be dynamic and centralized, allowing for more effective monitoring and management.

2.2 Background on blockchains

Due to the widespread adoption of cryptocurrency, blockchain technology has drawn much interest from academic and commercial communities. The main goals of blockchains are to create a tamper-proof transaction chain and to enable transactions between entities without trust. It is a repository for distributed ledgers comprising data blocks produced through cryptography. A distributed network manages or controls a blockchain that provides traceable, transparent, and immutable data storage while ensuring data integrity. Without the consent of more than half of the participants, the encapsulated data in the block cannot be changed maliciously [12]. Distributed ledgers, smart contracts, a consensus algorithm, and peer-to-peer connectivity are all included in a complete blockchain system. Blockchain is categorized into three types: (i) Permissionless blockchains, also known as public blockchains, (ii) Permissioned blockchains, also known as private blockchains; and (iii) Federated blockchains, also known as Consortium blockchain. The former allows

anyone to participate in the blockchain as a viewer (reader) or contributor (writer) throughout the consensus process.

Suitable blockchain for proposed approach As the proposed HSN approach needed permissionless and permissioned blockchains, the consortium blockchain was considered the most appropriate. Patient and HSN networks (inter-base stations) are connected with the consortium blockchain, while IoMT nodes, BS, and CH (intra-base stations) require permissioned blockchains. The patient can therefore join as a reader using only their smartphone. In contrast, the BS of the HSN network is authorized to act as a writer on a permissionless blockchain. In this way, it maintains the integrity of the medical report. It helps us protect patient data from unauthorized access. The Hyperledger Fabric blockchain source code partially implemented for healthcare smartphone networks is available at GitHub repository [13].

2.3 Related work

The difficulty of identifying malicious nodes in HSNs has drawn research and business interest. This section briefly reviews several studies on employing trust-based security to identify compromised and malicious nodes. Trust and reputation-based security methods are in place to protect against internal assaults. Based on the analysis results, these systems detect malicious communication nodes based on their dependability [14]. Since Internet of things (IoT) devices have limited resources, implementing cryptographic security is challenging in IoT, where the devices are dispersed. Network security, protection from adversaries, and providing integrity, confidentiality, and authentication are highly desirable, with good throughput and minimal delay. Due to malicious nodes in the network, standard security measures can also not correctly identify internal attacks while protecting against external ones. As a result, trust-based security has shown to be more resistant to identifying malicious nodes and more conducive to ensuring reliable data transmission.

Another research suggests a trust-based cross-layer architecture (TruFix) [15], which offers protection against a limited number of network attacks. A more recent proposal for an underwater sound sensor network that employs game theory is the Cloud-based trust evidence generation model (TMC) [16]. Furthermore, the approach resists internal assaults, such as denial of service, jamming, on-off attacks, and bad-mouthing. Still, it could be more reliable regarding message transmission between nodes. Furthermore, the authors have developed a model for trust-based neighbor choice employing an activation function (AF-TNS) [17] that utilizes just direct trust with additive metrics to assess the reliability and retention of trusted neighboring nodes in wireless sensor networks (WSNs). Due to AF-TNS's use of a method to stop misleading information from being spread by malicious nodes against trustworthy nodes, incorrect decisions are made due to the identification of malicious nodes.

Blockchain-based trust management in hierarchical clustering approach The decentralization, immutability, and security of blockchains are advantageous in

addressing flaws in communication networks [18]. Using clustering with the Blockchain has proven beneficial for scalability, energy consumption [19], and authentication [20] that protects security and privacy. Implementing algorithms based on artificial intelligence to identify malware have been linked to adopting blockchain technology [21]. Applying blockchain and trust management to improve security in IoT contexts have been extensively investigated by existing clustering methodologies. Unfortunately, only a few research works, mainly in the IoT context, have simultaneously taken advantage of the employment of both strategies. Tariq et al. proposed methods to identify attacks in an IoT environment [22]. She et al. suggest a method for identifying malicious nodes in Wireless Sensor Networks [23]. According to the authors of BCTrust [24], blockchain and trust-based authentication mechanisms are used in IoT.

Integration of blockchain-based trust management in a healthcare organization

Khatoun et al. [25] suggested a blockchain-based solution for handling healthcare data. This study also extensively highlights patient data management workflows using the blockchain network. It enables many medical industry participants to deliver improved health care and reduce expenses. Srivastava et al. [26] identified numerous advantages and limitations of blockchain-based technologies for monitoring remote patients with IoT devices. The authors assess various cryptographic methods used in Internet of Things (IoT) devices, including security features. Yanez et al. [27] presented a blockchain-based data allocation method for Internet of Things. This innovative approach employs context-aware algorithms for data allocation on-chain to increase the usefulness of fog and cloud environments. Namasudra et al. [28] presented a solution for IoT-based healthcare infrastructure that protected security and privacy in 2022. It improves safety and prevents unauthorized access to medical certificates. Smart contracts deliver the smartphone platform's core functionality, but the solution needs to increase its scalability. Abdellatif et al. [29] created an intelligent and secure healthcare system using edge computing and blockchain technologies. This system regulates healthcare data exchange services between regional healthcare organizations and permits safe patient record sharing. Jeet et al. [30] proposed an encrypted blockchain-based paradigm for IoT-powered medical facilities. This system can block-based the encryption of patient information through a cloud server. The Internet of Medical Things (IoMT) powered by blockchain was proposed by Dai et al. [31]. The proposed approach addresses the security and privacy issues associated with an integrated approach to IoMT and blockchain techniques.

Trust based intrusion detection system An IDS is one of the most widely used mechanisms for securing various networks, including healthcare smartphone networks. Anomaly-based IDS and Rule-based IDS are popular types of IDS. Rule-based IDS identifies future attacks by comparing known events to stored malware signatures [32]. An anomaly-sensitive IDS identifies a discrepancy between the main profile and a previously specified standard profile to identify malicious behavior [33]. If any possible danger is discovered, the IDS will trigger an alert. Alevizos et al. [34] presented a host-based IDS cooperation architecture that allows each IDS node to transmit challenges and assess the trustworthiness of everyone based on its

own experience. The forgetting factor emphasizes the importance of new acquired expertise [35]. Li et al. [36] discovered that not every IDS has the same degree of sensitivity in recognizing all types of intrusions. This is because recognition rate should depend on their signatures and applied machine learning techniques to increase detection performance.

Because IDS rely mostly on centralized or distributed integration, communication methods are unscalable. Therefore, a dynamic detector works with a decentralized location and routing infrastructure to solve node communication problems. Because authors believed that all peers were trustworthy, their strategy was prone to insider or betrayal attacks in which specific nodes became malicious unexpectedly. As a result, we established the concept of intrusion sensitivity and studied how well it performed in calculating trust levels for various IDS nodes. To increase the robustness of IDS, the trust management approach is based on intrusion sensitivity [10]. The lightweight IDS (SNORT mobile version) placed on IoMT nodes and smartphone to aid network monitoring, traffic recording, and security policy enforcement. It interacts with several HSN nodes and a single central server.

2.3.1 Resilient against insider attacks

The robustness of blockchain-based trust management systems is examined against different kinds of insider attacks, covered in the literature study in Table 1. Even though blockchain security has dramatically improved, attackers have a variety of techniques to hack it [11]. The authors discuss several possible attacks and their solutions covered in related studies.

Transaction forgery attack Initially, Liu et al. [18] discuss Transaction Forgery in their research. The data kept on the blockchain is often comprehensive, correct, and consistent. An attacker who attempts to tamper with persistent data must pay a high price. Therefore, it is more susceptible to attacks during data transfer. Furthermore, it can intercept transactions posted by the IoMT node on the blockchain. The attacker can impersonate trusted nodes and submit phony transaction requests to the blockchain network after often accessing the encrypted transaction records.

In the proposed scenario, each HSN node gathers data and makes storage requests as transaction data to store in the blockchain; however, the attacker can intercept the request and manipulate the amount of data collected to be false or incorrect. It is conceivable that the attacker's HSN node is an authentic device inside the blockchain network. So, it tries to inflate the quantity of data acquired by transmitting false data while requesting storage. The computation of the private key of CA is difficult or impossible for an attacker node. Therefore, it cannot alter CA's confirmation message. Instead, the CA will authenticate whether the communication originated from a legitimate device once the attacker delivers the forged storage request. In such a case, verify that the node contains the data specified in the request. If everything checks out, CA will append the digital signature to the valid transaction and send it back to the node. In response to a proposal, the node uploads the transaction data to the blockchain network. It would also check whether it was the actual digital signature of the CA.

Table 1 Comparisons between our proposed HSN approach and existing related research

Year	Author/framework name	Research methodology	Research outcome	Limitations
2017	Umar et al./TruFix [15]	Use Fuzzy logic, Trust-based cross-layer Framework	Sybil, blackhole	Prone to network attacks, inefficient routing process
2017	Jiang et al./TMC [16]	Use game theory, Interaction and recommendation-based trust Computation	Selective forwarding, On-Off, bad mouth	Unreliability for message delivery
2018	AlFarraj et al./AF-TNS [17]	Activation function, direct trust, and additive metric	Isolate malicious nodes	Fail to stop the spread of false info by malicious nodes
2018	Hammi et al./BCTrust [24]	Ethereum Blockchain-based trust model in IoT	Authentication of IoT nodes	Security analysis not covered
2019	She et al. [23]	OPNET, Ganache in WSN Blockchain and cluster-based trust model	Deliver certificate and manage communication in nodes	Detection performance is not satisfied
2019	Srivastava et al. [26]	Blockchain-based encryption in IoT devices	Securely monitor the health of remote patient achieve anonymity	Performance analysis is not covered
2020	Tariq et al. [22]	Ganache Blockchain in IoT, code driven trust model	Analyze node behavior and energy consumption	Take long time to detect greyhole attacks
2022	Namasudra et al. [28]	Implement smart contracts, secure IoT, and smartphone communication	Access medical certificates in IoT based healthcare	Need to improve scalability
2022	Jeet et al. [30]	Encrypt patient data in blockchain block-based encoding over cloud	Secure patient data in IoT-based healthcare	Need to improve system performance, not applicable in non-cloud networking
2022	Bhan et al./Proposed HSN [43]	Encrypt remote patient data in blockchain, direct and recommendation-based trust computation	Exchange patient data in secured trusted HSN, Analyze the performance under DoS attacks, identify malicious nodes in HSN	Simulate the performance, need to implement in real healthcare environment

Eclipse attack in blockchain network In the Eclipse attack, the attacker effectively takes over a peer-to-peer network [37]. In an Eclipse attack, the attacker tries to get control of the node of the intended victim. Every node in a blockchain network depends on connections to many other nodes to get a comprehensive network view. Therefore, an attacker prevents the victim from obtaining complete information about the system. In our proposed scenario, an attacker employs an Eclipse Attack to prohibit non-mining nodes from accepting query and storage requests on the blockchain. It makes it challenging to initiate additional smart contract events, disrupting regular device-to-blockchain interaction. Only two malicious blockchain nodes are required to isolate and affect other nodes for an eclipse attack. By upgrading the software of the blockchain, malicious nodes are no longer able to launch attacks. [11].

Vulnerability attack by smart contract users A blockchain platform with open-source smart contract capability provides visibility to each user. Therefore, it is elementary for attackers to expose a significant flaw in a smart contract. It is presumed that an attacker may use smart contract flaws to tamper with the data that persists on the blockchain. The blockchain's data are stolen and deleted if a more significant vulnerability exists. While building the smart contract, we conducted security tests and code audits to eliminate recursive calling [37], timestamp reliance, arithmetic challenges [38], and return value issues [11].

Majority attack by malicious users When most nodes in a blockchain network turn malicious and cooperate with the attacker, this is when the attacker controls more than 50% of the processing power in the blockchain network. Data in the blockchain cannot be guaranteed to be secure [11]. The attacker can use computational power to tamper with blockchain records. In the proposed HSN blockchain network, the system owns and controls the mining nodes, preventing blockchain majority attacks. The mining nodes are truthful, obey the rules to some extent, and ensure that the attacker cannot gain more than half of the processing power. It ensures that such an assault is prevented as soon as possible.

IoMT nodes failure IoMT nodes may behave differently during the proposed HSN network scenario when collecting and exchanging dispersed data. The HSN protocol provides a set of operations for a group of nodes, ensuring that each node will accept the result of processing. Nevertheless, over the long-term usage of such IoMT devices, someone might develop software or hardware issues that prevent them from functioning correctly, like network service failure, device downtime, and network communication failure, among other things. There are two types of difficulty in this category. The first is that data are lost or delayed. The other is considerably more problematic because malicious nodes in the network can communicate fake data to other users, referred to as *The Byzantine Generals Problem* [39]. Alternatively, an attacker may pose as an authorized or lawful IoMT if it knows about the failed devices. It then fraudulently reports the absolute quantity of information, causing data insecurity or inconsistency. Blockchain consensus algorithms, such as Paxos and PBFT [39], are also used in the proposed HSN to successfully tackle the IoMT device malfunction. So, it will avoid or solve these kinds of potential difficulties.

Other possible attacks sybil, blackhole, On-Off, bad mouth, Sybil Attacks: Attackers create several aliases on individual nodes to gain control of a network, known as virtual Sybil nodes. Each node along the routing path has a distinct identity and is located in a different location. Consequently, attackers drain communication resources among network nodes and cause uncertainty. In TruFiX [15], the reputation of the virtual nodes was suggested as a possible solution. Despite the forced fairness process, the non-malicious nodes surrounding A1 were selected to deliver the packets due to congested network conditions. During a bad-mouthing attack, a neighboring node will make inappropriate recommendations. Consequently, the attacker damages the reputation of the target node. While On-off attacks sometimes cause nodes to behave favorably or adversely differently. Because of this, the attacker can earn the targeted trust before their subsequent attack. The author of TMC's [16] controls bad-mouthing and On-off attacks by employing trust assessment with energy constraints and additional metric-based node analysis.

Security resilience of proposed HSN The literature survey reveals in-depth analyses of numerous trust models that can fend against attacks employing direct and indirect trust and other metrics for calculating trust. To manage cluster-based IoMTs, we developed the Clustered Hierarchical Trust Management System (CHTMS). Describing the pre-defined objective of malicious nodes shows that the trust management system is resilient at the nodes, cluster, and BS levels if it can prevent malicious nodes from achieving their goals. The three phases of trust assessment in the suggested architecture are at the IoMT node level using the concept of sliding time window [40], Cluster Head (CH), and Base-Station (BS) level. The direct and indirect interactions of nodes are considered when calculating total trust. Indirect trust is based on recommendations from peers, and direct trust is based on the success and failure of data transmission between nodes. Uncertainty and trust make up the final degree of trust at the base-station level. The cumulative trust for the specific cluster level is determined at the BS level. The forgetting component is incorporated into the proposed scheme to emphasize the significance of newly acquired experiences. Further, we investigate the effects of different forgetting factors on the processing of trust based on the results of the most recent study [41]. Compared to the value of forgetting factor 0.9, forgetting factor 0.8 has a higher security performance.

The suggested CHTMS can identify malicious IoMT nodes having resilience against insider attacks, according to observations of the related literature. However, we implemented Denial-of-Service (DoS) assaults and observe at a particular level while increasing the throughput and network stability. The associated works outlined above and listed in Table 1 serve as partial inspiration. In the context of these problems, the CHTMS proposal enhances the performance, throughput of the network, and attack detection. In addition, it identifies the malicious IoMT node accountable for misleading reporting.

Throughout this paper, we focus on defending against insider attacks, and upgrading intrusion detection systems (IDSs) are outside the scope of our investigation. Furthermore, our proposed method can also be applied in other fields, despite being evaluated in the healthcare sector. Using blockchain technology to establish trust in organizations is the purpose of this study to contribute to the current state of the art.

3 Proposed methodology

This section presents the concept of a blockchain-based healthcare smartphone network maintaining a hierarchical trust management system. The proposed HSN collaborative system combines energy-efficient data collection with safe data exchange across IoMT nodes using the Hyperledger blockchain. The Hyperledger blockchain successfully retains a tamper-proof ledger distributed by the involved IoMTs without any requirements for a trusted third central entity, ensuring data security and integrity during IoMTs communication. In our proposed HSN, the malicious IoMT nodes are identified at the cluster level, with the help of trust computation of individual IoMT nodes and integration of SNORT IDS to prepare a blocklist of malicious nodes. In the next step, we analyze the security features of the proposed model against various insider attacks. Additionally, analyze HSN's robustness when undergoing DoS attacks by finding the rate of failed and successful transactions. In addition, compare the envisioned HSN with other performance and security analytics solutions. Table 2 shows a description of the notations used in the proposed HSN.

3.1 Smartphone-powered IoMT devices

Recent studies focused on applications of smartphone-powered biomedical devices with biochemical, immunological, and genetic characterization [42]. Instant monitoring of Uric Acid (UA) [1], immunological detection of HIV, prostate-specific antigen (PSA), and protein bio-markers demonstrated utilizing smartphones as tiny electrochemical analyzers at the point of care. This research presented an IoMT device (Medical dongle) powered by a smartphone and functions as a miniature electrochemical analyzer to measure uric acid and blood sugar using a disposable test strip. Using OTG wire, the medical dongle stays connected to the smartphone. The biochemical data are stored on a smartphone and sent to HSN. The purpose of the HSN platform is to accept patient records captured by a medical dongle and deliver expert medical services. Through the built-in HSN platform, the concerned doctor will immediately access the relevant health data of their patients and remotely start prescribing.

To calculate the reliability of the test strip, UA and blood glucose levels are measured by a hefty medical biochemical analyzer (Hitachi 7600P) at values of 401 mol/L and 5.1 mmol/L, respectively. The low concentration level (UA below 200 μ mol/L and glucose below 6.0 mmol/L). The range of medium concentration level (for UA 201 μ mol/L to 435 μ mol/L and for glucose 7.0 mmol/L to 10.0 mmol/L). The high concentration level (UA above 436 μ mol/L and glucose above 11.0 mmol/L). The UA and glucose test strips integrate with the medical dongle to ensure consistency and reliability. The current generated by the electrochemical reaction measurements by the IoMT device (medical dongle) using glucose and UA test strips was also presented in Fig. 2. The current signal mapped to get summarized test value, and coefficient variance in IoMT results is below 3% and 5%, respectively, for the glucose and UA test, shown in Tables 3 and 4, respectively. These test

Table 2 Defining abbreviations and their meanings

Abbreviation	Meaning/values
IoMT	Internet of Medical Things
HSN	Healthcare Smartphone Networks
CHTMS	Clustered Hierarchical Trust Management System
DoS	Denial Of Service
BS	Base Station
CH	Cluster Head
BTEM	Belief-based Trust Evaluation Mechanism
TFR	Transaction Failure Ratio
TSR	Transaction Successful Ratio
IDS	Intrusion Detection System
AF-TNS	Trust-based Neighbor Selection using Activation Function
WSN	Wireless Sensor Network
UA	Uric Acid
STW	Sliding Time Window
$T_{A,B}$	Trust Value between node A to node B
$S_{A,B}$	Number of successful communication between node A and B during time Δt
$F_{A,B}$	Total number of failed communication between node A and B during Δt time
C1	Average value of half of all trusted nodes
C2	Average value of one-third of all untrusted nodes
X_a	Set of trustworthy nodes in the list of nodes A
Y_a	Set of untrustworthy nodes in the list of nodes A
N	Total nodes containing trustworthy, untrustworthy, and unclear nodes
f	20
g	15
$ X $	Total number of trusted nodes in CH
M_{ts}	Cluster head's trust state matrix
$S_{ch,1}$	Trust state of node one at the CH
$PT_{a,b}$	Trust value using a time window based on previous transactions
$BSS_{a,b}$	Trust value between node A to B, suggestion from their BS
$S_{bs,ch}$	Number of successful conversations between BS and CH during Δt time
$F_{bs,ch}$	Total number of failure interactions between BS and CH during Δt time
G_n	Number of clusters/groups in HSN network
TSV_{ch_i}	Computed trust state vector

findings show that the smartphone-powered IoMT devices accurately measure UA and blood sugar levels and promise to use remotely in healthcare situations.

3.2 Scheme for trust management

Our premise is that benign nodes often communicate successfully and submit genuine suggestions. On the other hand, malicious nodes attempt as many failed

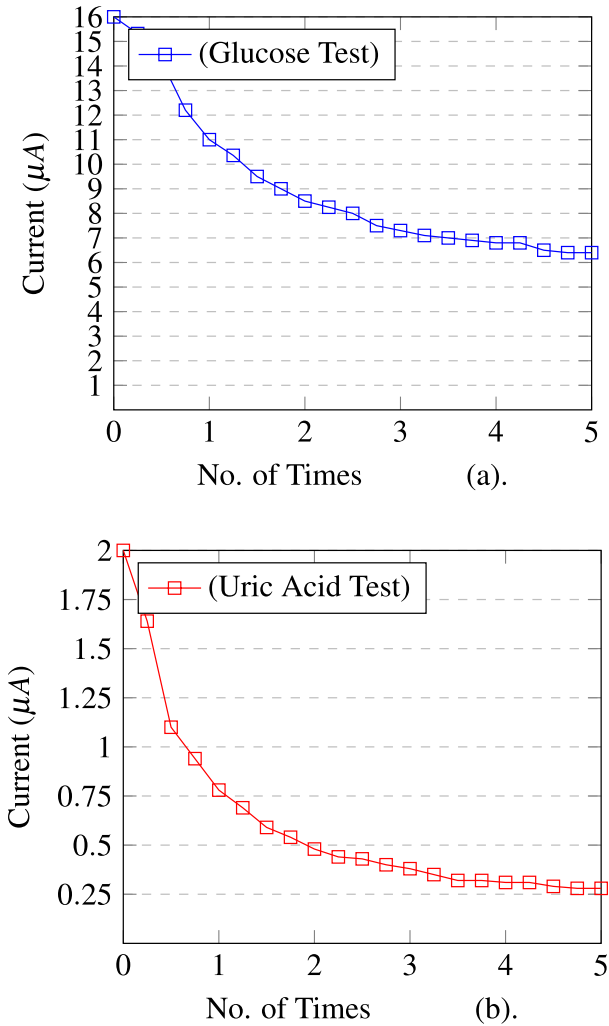


Fig. 2 Chronoamperometric curves represent the reading of Current (μA) generated during testing of **a** Blood Glucose and **b** Uric Acid; from the blood sample taken from a healthy person measured by smartphone-powered IoMT medical dongle (Hitachi 7600P)

Table 3 Level of glucose measured by smartphone-powered IoMT medical dongle

Level of glucose	Value	Coefficient variance (%)
Low	4.1 ± 0.1	3.0
Medium	8.6 ± 0.2	2.5
High	13.0 ± 0.3	2.5

Table 4 Level of uric acid measured by smartphone-powered IoMT medical dongle

Level of uric acid	Value	Coefficient variance (%)
Low	165 ± 6	3.0
Medium	287 ± 7	2.0
High	570 ± 10	2.0

communicate as possible and make fake suggestions regarding benign nodes. The distinction between benign and malicious nodes is arbitrary. From the perspective of one node, a node may be benign, but from the perspective of another IoMT node, it may be malicious. Section 3 focused on capturing the activity of malicious nodes and simulating how they try to gain an unfair benefit in the trust model. We demonstrate the robustness of the protocol to face such malicious activity. We have applied this analysis to higher clusters in a modular manner. Therefore, we must start by understanding the malicious and unfair advantage concept. Both of these characteristics define a malicious node. A malicious node's purpose while dealing with other nodes is to have as many failed interactions as possible to retain the following objectives:

- Achieves a higher trust value than the actual computed trust value; more precisely, it enters the trusted zone while its actual position is in the untrustworthy or uncertain zone.
- It tries to lower the trust value of the benign node, and
- If feasible, raise the trust value of a cooperating malicious node.

The HSN communication is based on two topologies in the proposed HSN model shown in Fig. 1.

1. The first is an intragroup topology, in which trust management is centralized inside the hospital. The only communication considered here is between one BS and its associated CHs within a single hospital (over the hospital's intranet).
2. The other is an intergroup topology, in which distributed trust management occurs among hospitals. Here, communication is considered between one BS to another BS of another hospital for multiple hospitals (over internet connectivity among hospitals).

Each IoMT node (i.e., IoMT devices and smartphones) in the HSN network calculates the trust values for every other node (only for group members) separately. Then, a node assigns one of the three possible state options based on the trust values. The possible states include 1) trusted nodes, 2) untrusted nodes, or 3) uncertain nodes. For mathematics brevity and to provide an appropriate level of granularity, the 3-state approach was chosen. Following that, each node transmits the trust state of all the group nodes to the CH installed at the individual Departmental Information Service Center. After that, centralized trust management takes control. The CH

recognizes the malicious node(s) and alerts the BS located at the Hospital Information Service Center of the same and receives the trust status of every group member.

Additionally, upon request from the BS, each CH sends the trust values of the neighboring CHs. The BS assigns one of the group's three potential states when it has all the data. Upon request from a particular CH, the BS will provide an update on the current condition of certain CHs. Our grouped trust management model comprises three main phases: (i). Computation of trust first at the node level (ii). Computation of trust at the level of CH and (iii). Last calculation of trust at the BS level.

3.2.1 Computation of trust at the node level

In HSN, the nodes may be any IoMT devices or smartphones. Peer recommendations or time-based prior contact at the node level calculate a trust value, as shown in Fig. 3. For instance, when node A wants to interact with node B, it first checks whether A has any prior history of interacting with B within a particular period. If the answer is yes, node A makes the right decision based on the last communication; if the answer is no, node A uses the peer recommendation method.

Trust evaluation based on past communication using a time-based approach

The trust calculation indicates the degree of confidence in the node's reliability at each node. As a result, the authors use the concept of a Sliding Time Window (STW) [40]. It takes relative time into account and mitigates the effect of network conditions on the cumulative trust calculation. This is because the communication protocol is often preceded by timestamps, every node that delays packet delivery by utilizing the sliding timing window.

The timing window (Δt) divided into multiple time units determines previous communication success and failure rates. Prior communication took place at each point in time. It logs the units contained within the timing window. When a unit of time passes, the window slides to the right for a one-time slot. As a result, the window keeps forgetting its experiences. The window length could be reduced or increased based on scenarios derived from network analysis. Figure 4 illustrates the HTMS time window scheme. The time window Δt is made up of four units. The initial unit of (Δt_1), indicating the number of successful and unsuccessful communication/transactions, is two and one, respectively. Throughout the entire Δt_1 interval, the number of success and failure attempts is ten and four. Following the passage, Δt_1 , the new time interval Δt_2 decreases the values associated with prior communication that occurred during the very first unit Δt_1 ($S=2, F=1$) and considers only the values of the last three units of (Δt_1) plus the value of a recently added unit ($S=5, F=2$) on the right.

$$\begin{aligned}
 T_{A,B} &= \left[100 \left(\frac{S_{A,B}}{S_{A,B} + F_{A,B}} \right) \left(1 - \frac{1}{S_{A,B} + 1} \right) \right] \\
 &= \left[\frac{100(S_{A,B})^2}{(S_{A,B} + F_{A,B})(S_{A,B} + 1)} \right]
 \end{aligned}
 \tag{1}$$

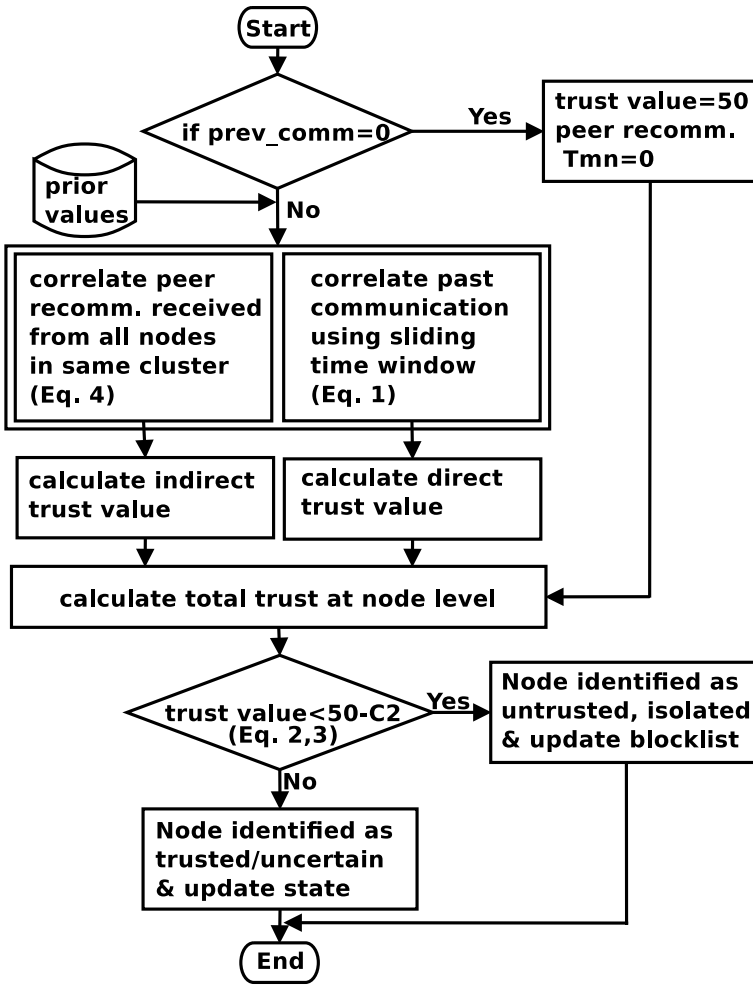
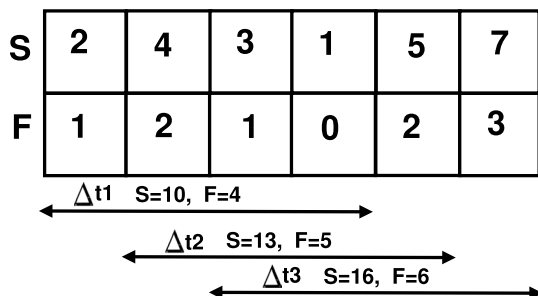


Fig. 3 Flow diagram of computation of trust at the node level

Fig. 4 Sliding time window-based scheme for trust computation among HSN network entities



Along with the time window based on past communication, the trust value $T_{A,B}$ of node B to node A, the range defined as 0 to 100 (round the nearest integer),. $S_{A,B}$ is the weighted sum of the number of successful communication between node A and B during time Δt . $F_{A,B}$ is the total number of failed communication between node A and B during Δt time. The expression $(1 - \frac{1}{S_{A,B}+1})$ in Eq. 1 represents the rise in trust value over time, based on the number of successful communication attempts. This method was chosen over a linear function because it very slowly reaches one as the number of successful interactions increases; thus, it would take significantly longer for a node to increase its trust value for another node. Likewise, it takes a long time to decrease the trust value of a node in the same cluster formed at each departmental level. Thus, the equation 1 is designed to mitigate the effects of a declaration of a few wrong communications caused by network traffic problems. The trust value rises when the number of successful intercommunication increases but remains modest if communication failure is also relatively high. For instance, if six unsuccessful interactions and five successful, the trust value is 4.5. Following is the calculation of the trust value to judge the trust state of a node (N_j).

$$N_j(T_{A,B}) = \left\{ \begin{array}{ll} \text{trusted} & 100 - C1 \leq T_{A,B} \leq 100 \\ \text{uncertain} & 50 - C2 \leq T_{A,B} < 100 - C1 \\ \text{untrusted} & 0 \leq T_{A,B} < 50 - C2 \end{array} \right\} \tag{2}$$

C1 denotes half of all trusted node average values, and C2 represents one-third of all untrusted node average values, calculated in Eq. 3. The robustness of a node is directly affected by the use of half and one-third of average values in assessment.

$$C1 = \left[\frac{1}{2} \left(\frac{\sum_{i \in X_a} T_{a,i}}{|X_a|} \right) \right] \quad C2 = \left[\frac{1}{3} \left(\frac{\sum_{i \in Y_a} T_{a,i}}{|Y_a|} \right) \right] \tag{3}$$

X_a is the set of trustworthy nodes in the list of nodes A. Y_a is the set of untrustworthy nodes in the list of nodes A. N represents the total containing trustworthy, untrustworthy, and unclear nodes. Initially, all nodes' trust values are 50 while f and g are set to 20 and 15, respectively, to maintain the constraint: $f > g > 1$. By doing so, we can avoid ambiguous zones between trusted and untrusted zones.

Evaluation of trust value based on peer recommendations Let us say, a group consists of n nodes. In addition, every node keeps a trust value for all the other nodes in a group. When a node needs peer recommendation, it requests all group member nodes, excluding untrusted ones. For example, node M wants to compute the trust value of node N, represented as $T_{m,n}$. $|X|$ is the total number of trusted nodes in CH. The updated trust value $T_{m,n}$ is calculated in Eq. 4.

$$T_{m,n} = \left[\frac{\sum_{i \in X_a} T_{m,i} * T_{i,n}}{100 * |X|} \right] \tag{4}$$

$T_{m,i}$ is the recommender's trust value. A recommender's weighted value is also multiplied by the trust value sent by all the recommenders. While $T_{i,n}$ is the trust value sent by all the trusted nodes for node N. Node N's trust value shall not exceed the range of trust values among node N and all the recommender nodes of group X.

3.2.2 Computation of trust at the cluster head level

Each node transmits the trust state of all the group nodes to the CH installed at the Departmental Information Service Center. Authors believe that the CH has more computing power and memory.

Computation of CH group's trust state CH requests nodes for the trust states of all other group members to determine the overall trust value of nodes in a cluster, as shown in Fig. 5. This is because trust states are employed rather than actual trust values. First, the communication cost is reduced because only a simple state is conveyed to the CH. Second, a particular node's trust constraints differ from other nodes. For example, a given trust value may belong to the trustful zone for one node. Although for other nodes, it may only belong to the uncertain zone. As a result, calculating the global trust status of a collection of nodes is more practical and efficient when the trust state is used. For example, the group contains n nodes and one CH. The CH will transmit the request packet to the group regularly. As a result, everyone in the group conveyed their computed trust status (S) to the CH. There are three

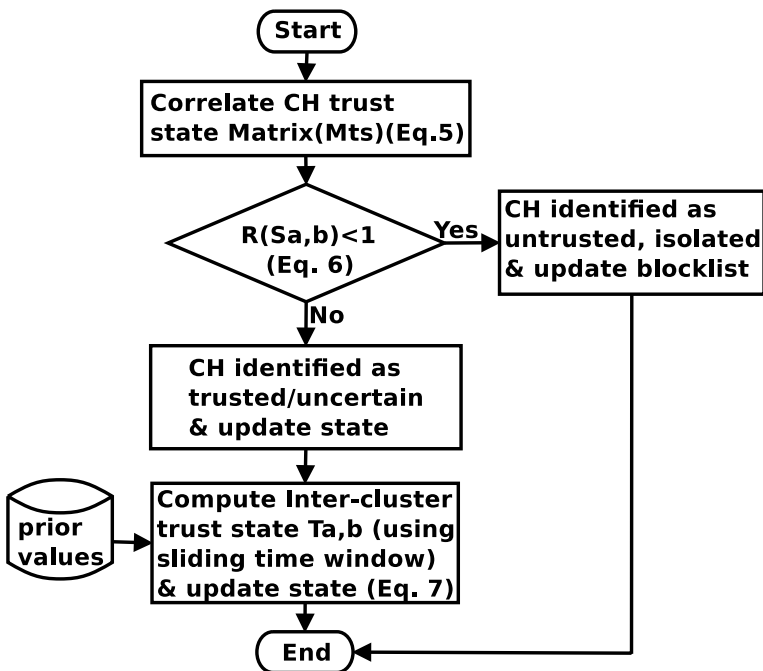


Fig. 5 Flow diagram of computation of trust at the cluster head level

states that S can be in: trustworthy, untrusted, and uncertain. CH will maintain all these trust state values in matrix M_{ts} represented in Eq. 5.

$$M_{ts} = \begin{bmatrix} S_{ch,1} & S_{1,1} & \dots & S_{1,n} \\ S_{ch,2} & S_{2,1} & \dots & S_{2,n} \\ \dots & \dots & \dots & \dots \\ S_{ch,n} & S_{n,1} & \dots & S_{n,n} \end{bmatrix} \tag{5}$$

Where M_{ts} denotes the cluster head’s trust state matrix, the trust state $S_{ch,1}$ indicates the trust state of node one at the CH. In contrast, $S_{1,1}$ to $S_{1,n}$ represent the trust state of node one sent by the remaining n nodes in a group. Based on the computation of relative differences in the represented matrix of trust states of the CH (M_{ts}), assigned a global trust status to a node n, i.e., $S_{ch,n}$. As a result, the CH declares a random variable R, S_n is the sum of n random variables, and $\sqrt{\frac{n}{3}}$ is the standard deviation. The CH defines the behavior of random variables $R(S_{a,b})$ as per equation 6.

$$R(S_{a,b}) = \left\{ \begin{array}{l} \text{trusted} \quad \text{when } S_{a,b} > 1 \\ \text{uncertain} \quad \text{when } S_{a,b} = 1 \\ \text{untrusted} \quad \text{when } S_{a,b} < 1 \end{array} \right\} \tag{6}$$

Computation of intergroup trust state During intergroup communication, every CH keeps track of their previous interactions with other groups in the same way as individual nodes maintain track of all other nodes. A group’s trust values are computed based on previous interactions or information from the BS. Assume CH A wishes to determine the trust value $T_{a,b}$ with another CH B. It determines trust value either using a time window based on previous transactions $PT_{a,b}$ (when $PT_{a,b} \neq \phi$) or a suggestion from their BS $BSS_{a,b}$ (when $PT_{a,b} = \phi$), the value of $T_{a,b}$ calculated in Eq. 7.

$$T_{a,b} = \left\{ \begin{array}{l} \left[\frac{100(S_{a,b})^2}{(S_{a,b}+F_{a,b})(S_{a,b}+1)} \right] \quad \text{if } PT_{a,b} \neq \phi \\ BSS_{a,b} \quad \text{if } PT_{a,b} = \phi \end{array} \right\} \tag{7}$$

3.2.3 Computation of trust at the base-station level

The BS keeps a record of previous interactions with Cluster heads in the same way as every node does, as shown in Fig. 6. The BS periodically computes the trust value between every CH, represented as $TV_{bs,chi}$ during the time window Δt in Eq. 8.

$$TV_{bs,ch} = \left[\frac{100(S_{bs,ch})^2}{(S_{bs,ch} + F_{bs,ch})(S_{bs,ch} + 1)} \right] \tag{8}$$

The number of successful conversations between BS and CH during Δt time is $S_{bs,ch}$. In contrast, the total number of failure interactions between BS and CH during Δt time is $F_{bs,ch}$. The BS broadcasts request packets to all the Cluster Heads regularly.

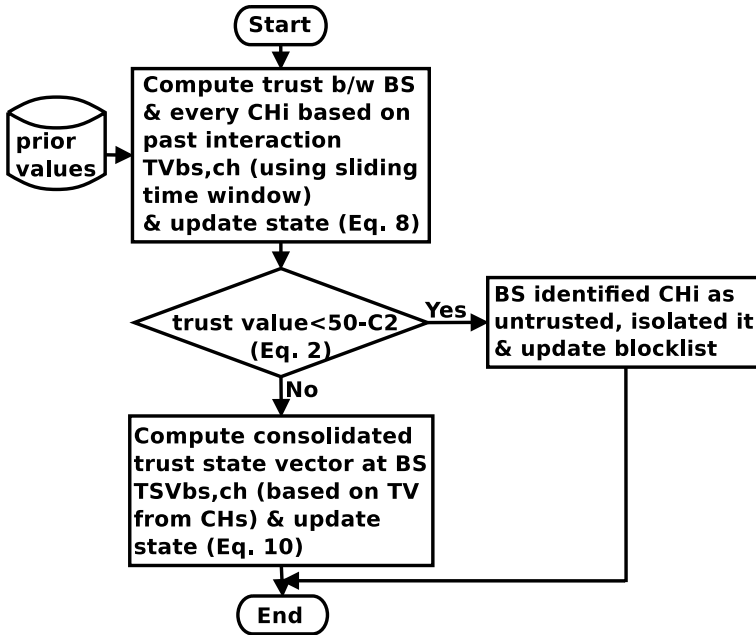


Fig. 6 Flow diagram of computation of trust at the base-station level

In response to BS requests, all cluster heads send their computed trust state vector TSV_{ch_i} to the BS, calculated as per equation 9. Now BS prepares a consolidated Trust State Vector (TSV) and computes cluster-wise (CH_i) trust state TSV_{bs,ch_i} as given below:

$$TSV_{ch} = (TV_{ch_1} + TV_{ch_2} \dots + TV_{ch_n}) \tag{9}$$

$$TSV_{bs,ch_i} = \left[\frac{TVS_{ch}(TV_{CH_i,CH_j})}{CH_n - 1} \right] \tag{10}$$

In the Eq. 9, $TSV_{bs,ch}$ is the trust value between BS and CH, TV_{ch_i,ch_j} is the trust value between two different clusters, namely CH_i and CH_j , while CH_n represents the total number of clusters in HSN.

3.2.4 Blockchain-based centralized trust management in proposed HSN

Healthcare management requires a single central server, the high-level design shown in Fig. 1; however, it could become a weak point of failure. Because blockchains allow multiple nodes to interact in a distributed fashion without a central authority, so blockchain is becoming popular gradually. By merging blockchains, it proposes a trust management system. The blockchain-enabled trust management strategy is

depicted in Fig. 8, which divides the HSN into two essential layers: the HSN and the blockchain.

1. **HSN Layer:** This feature enables HSN nodes (MySQL database) to communicate with the centralized base server in the usual way. The central server responsible to manage/control entire HSN network, so it could be single point of failure as shown in Fig. 8. But it helps a healthcare institution keep its existing foundation while lowering the cost of deployment. In order to deploy blockchain-enabled trust management, the overall design needs to be modified.
2. **Blockchain layer:** This layer creates a private blockchain (Hyperledger Fabric) that permits each node to submit undesirable or malicious node characteristics. Since each node has access to the blockchain, it can examine the trust value of malicious nodes. It can immediately update its local blocklist, gain further information, and transmit messages to the target node. At the same time, updating the list quickly in traditional architecture is challenging.

Centralized trust management The BS is responsible for assessing trust at a different level in the HSN system and malicious node detection. There are three key components in the Base Station and IoMT node: (i) the communication component, (ii) the trust computation component, and (iii) the allow-listed and block-listed HSN nodes.

1. **Communication Component:** This component is managed connections between nodes and the server and transmits essential information to a central server. It assists in gathering essential data from nodes to aid in the trust computation process, which is vital to the interaction's success. It also assists in updating its blocklist and forwarding the revised blocklist to the relevant HSN nodes.
2. **Trust Computation Component:** This component primarily assists in calculating HSN node trust levels based on the obtained data and peer recommendation, identifying malicious nodes, and constructing the blocklist.
3. **Allowlisted of HSN Nodes:** It keeps the updated list of allow-listed nodes of HSN. Some security measures, in particular, ensure the list is updated dynamically and correctly.
4. **Blocklist of HSN Nodes:** Most components include a list of prohibited smartphone or HSN nodes based on trust values computed on the server side. The list will be dynamic based on healthcare management's response to decreasing the effect of false positives [10].

The blockchain-enabled trust management strategy has two advantages: 1) Monitoring blockchains assist in rapidly updating the blocklist among HSN nodes. 2) It permits more capable nodes to communicate with potentially anomalous nodes and learn more about their traffic status by checking blockchains. Furthermore, per

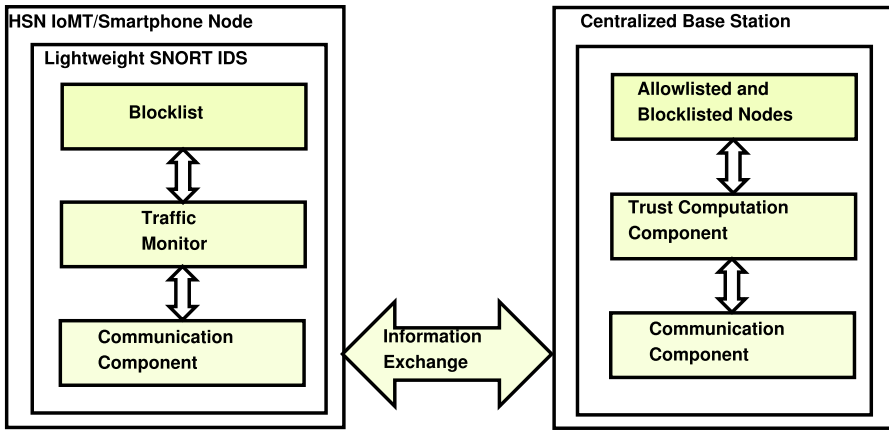


Fig. 7 Integration of Lightweight SNORT IDS in HSN Node (IoMT/Smartphone) and Communication with Centralized Base Station

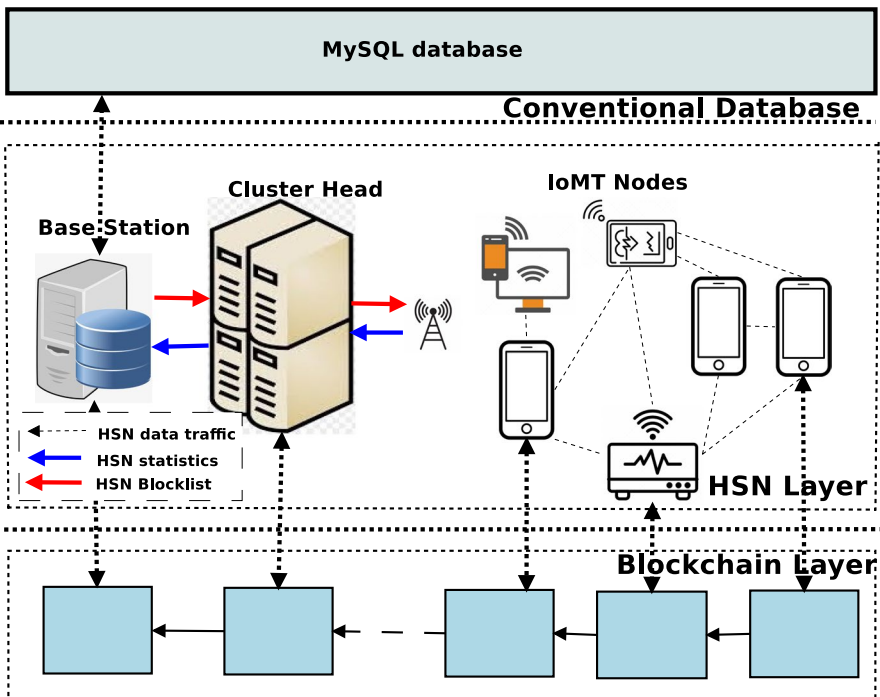


Fig. 8 HSN trust management based on hyperledger blockchain and MySQL database

the trust computation Eq. 3, a malicious node is discovered quickly by lowering the threshold value.

Dynamic update the list of blocklisted nodes $T_{a,b}$ represents a trust value between nodes A and B, calculated using the Eq. 4. However, an HSN node stopped for a malicious packet, leading to a high false-positive rate caused by accidents or careless operations. As a result, the dynamic blacklist design method reduces the impact of error rates. For example, the following judgment with a $T_{a,1}$ threshold:

1. If the trust value of blocked node $T_{a,b}$ is less than threshold T , it withdrew from the blacklist.
2. otherwise, the node is kept on the blacklist.

Integration of SNORT IDS in HSN IoMT node IDS technique includes sophisticated interaction between several HSN nodes and a single central server, as shown in Fig. 7. Placed lightweight IDS (SNORT mobile version) on the phones to aid network monitoring, traffic recording, and enforcement of security policies. It has three components (i) Communication Component, (ii) Traffic Monitor, (iii) Blocklist.

1. Communication Component: This component is similar to the centralized base-station component responsible for managing connections between nodes and the server and exchanging information with a central server to aid in the trust computation process. It keeps the blacklist updated.
2. Traffic Monitor: Inspecting traffic, transferring data, and recording data to the communication component is required.
3. Blocklist: It includes a list of prohibited smartphone or IoMT nodes based on trust values computed on the server side. The list is updated dynamically based on healthcare management's response to decreasing the effect of false positives.

3.3 Blockchain-based secured data transmission among HSN IoMT nodes

The IoMT-based data collection and output strategy are based on storage and query requests, which feed inputs to the blockchain-enabled storage system after each cycle of data gathering. IoMT encrypts gathered data through its private key and submits it to the blockchain as a storage request since each IoMT has a legitimate connection with it. IoMTs query the blockchain for data sharing, and the blockchain returns the results. We create a centralized, safe, and trustworthy storage system that establishes consensus on trustless ground and ensures that data/records have never been tampered with.

Hyperledger provides a more robust ecosystem than Ethereum and allows the development of smart contracts and business logic to be relatively straightforward. Furthermore, whereas the Ethereum platform is a public blockchain for open-source applications, Hyperledger is a private consortium blockchain meant for storing sensitive data, making it more appropriate for our experiment. Our simulation environment is developed in Python, and Hyperledger has already offered a Python interface for communicating with the blockchain. Blockchain networks are built using

the Hyperledger blockchain, which meets our needs. In Algorithm 1, we employ pseudocode to describe the safe data exchange process across IoMT nodes.

Algorithm 1: Blockchain-Based Secured Data transmission Among HSN IoMT-Nodes

```

1 Setup a private blockchain and deploy  $N$  Hyperledger blocks;
2 Initialize a smart contract on the Hyperledger blockchain;
3 Start block mining;
4  $\mathcal{I} \leftarrow$  set of HSN IoMT nodes;
5 for each  $IoMT_i$  in  $\mathcal{I}$  do
6   Define  $ECC\_Curve(a, b, SubGroup(p, g, n, h), secp256k1)$  to calculate  $(puK_i, prK_i)$ 
7   BS store  $keyRecords(i, puK_i)$ 
8 end for
9 for each  $IoMT_i$  in  $\mathcal{I}$  do
10   $d_i^t \leftarrow$  data collected from  $IoMT_i$  at timestamp  $t$ 
11  Compute cryptographic hash on collected data  $H(d_i^t) \leftarrow SHA256(d_i^t)$ 
12   $Dsign_i \leftarrow dsignECDSAsecp256k1(H(d_i^t), prK_i)$ 
13  Send  $(d_i^t, Dsign_i)$  to the BS.
14  if  $verifyECDSAsecp256k1(d_i^t, Dsign_i, puK_i)$  is valid then
15    if  $IoMT_i$  is trusted then
16       $Dsign_{BS} \leftarrow dsignECDSAsecp256k1(H(d_i^t), prK_{BS})$ 
17      Return  $(d_i^t, Dsign_{BS})$  to  $IoMT_i$ 
18    end if
19  end if
20  if  $IoMT_i$  receives digital signature  $(Dsign_{BS})$  from BS then
21    Initiate blockchain transaction  $(d_i^t, Dsign_{BS}, puK_i)$ 
22    if  $verifyECDSAsecp256k1(d_i^t, Dsign_{BS}, puK_{BS})$  is valid then
23      Wait for majority confirmation and commit transaction to blockchain
24    end if
25  end if
26 end for

```

Algorithm 1 depicted step by step process of workflow in our proposed approach to the healthcare system. Deploy the Hyperledger blockchain in the HSN network and create a Hyperledger block for each node. Later deploy the smart contract in the blockchain and start the block mining process. From steps 4 to 7 in the algorithm, use modern asymmetric key cryptography named Elliptic Curve Cryptography (ECC). Calculate the public key (puK_i) and private key (prK_i) for each IoMT node(i) from set \mathcal{I} [43] using ECC. ECC employs fewer keys and signatures for almost the same degree of security as RSA and enables rapid key generation, key agreement, and digital signatures. It is considered that ECC cryptography is a natural contemporary successor to the RSA cryptosystem, as per a comparison study by Zeinab et al. As per the [44], the smaller key size of 256 bits of ECC algorithms is equivalent to the width of an RSA algorithm with 256 bits of the key size of 3072 bits, so lightweight ECC algorithms are much more secure and are easily computed by low-powered computers. In this way, the ECC algorithm can be effectively run on each IoMT device. The BS stores the public key (puK_i) of each IoMT node(i).

After calculating the public key (puK_i) and private key (prK_i) for each IoMT node(i), In steps 9 to 12 in the algorithm, We collect the temporal data (d_i^t) from each IoMT node(i) and compute cryptographic hash value utilizing Secure Hash Algorithm (SHA) version 2 and produce 256-bit hash digest value $H(d_i^t)$. The IoMT nodes compute digital signature $Dsign_i$ using Elliptic Curve Digital Signature Algorithm(ECDSA) secp256k1 curve along with the private key (prK_i). In IoMT, the node sends data (d_i^t) and digital signature ($Dsign_i$) to the BS. From steps 13 to 16 in the algorithm, BS will verify the digital signature($Dsign_i$) based on the public key of the IoMT node(puK_i). If the digital signature($Dsign_i$) is valid, then, BS also checks the level of trust based on their previous communication. If IoMT node(i) is trusted, then, BS computes digital signature($Dsign_{BS}$) for hash value($H(d_i^t)$) of data(d_i) using BS private key(prK_{BS}) and returns it to the requested IoMT node. Following the algorithm steps 19 to 22, the IoMT node initiates the transaction using the digital signature of the BS ($Dsign_{iBS}$), along with the data (d_i) and the public key (puK_i). At last non-mining nodes verify and validate the transaction and commit it when they get confirmation from a sufficient number of non-mining nodes.

4 Result analysis

We configured the experimental environment for blockchain and the comparison database MySQL. The proposed solution for collecting data by IoMTs is explained in Sect. 3.3 and transferring data storage queries back and forth by Java. Section 3.3 explains private blockchain functioning in a healthcare smartphone network and proposes a blockchain-based data sharing approach using Hyperledger Fabric v2.2.

4.1 Implementation detail

The Hyperledger Fabric v2.2. is running on Ubuntu 20.04.3 LTS for x86_64 architecture, Intel Core 3.40 GHz i7 – 6700 CPU, and 16 GB of RAM, with all Fabric client nodes operating on the same computer using P2P networking. Fabric Peer v2.2 will generate a Java 11 Virtual Machine (VM) with the x86_64 architecture (OpenJDK version 11.0.4 11). JVMs, Docker images, and other supported tools are only tested on x86_64 standard architecture. Set the `CORE_CHAINCODE_JAVA_RUNTIME` environment variable to the base address of the Docker image. The Java Libraries (version v2.1.0) will establish a connection to the peer while operating. It is referred to as 'Fabric Peer Connectivity.' Execute the Python script that installs Binaries, Samples, and Docker Images on the same system to install all the prerequisites. Then, run the test-network script to verify Docker Desktop 2.5.0.1. Because the Fabric CA server is not part of a cluster, SQLite was chosen as the default database with the `fabric-ca-server`. DB file is in the CA server's root folder (home directory). The Certificate Signing Request (CSR) is configured to create Elliptic Curve X.509 certificates and keys (ECDSA). More identities can be enrolled and registered on the CA server. The CA server is configured using LDAP and must have a registered bootstrap identity to connect with the LDAP server. Now the CA

server starts listening to port 7054. Three Fabric nodes are included in our blockchain experiment. One node accepts query and storage requests from IoMT nodes, and the remaining two nodes can mine threads.

4.2 Configuration detail

Prepare the Genesis block configuration file during the initialization process, which defines a separate blockchain network ID. In hexadecimal, the difficulty is initialized to 0×0400 (1,024 in a decimal number system). $1/1024$ means that, on average, 1 out of every 1,024 hash computations will be successful. Reset the gasLimit to $0 \times \text{FFFFFFFF}$, which is the maximum number of calculations a block can handle. The coinbase parameter of the 160-bit address receives rewards upon successful mining. The mixhash and nonce parameters are 256-bit and 64-bit hash values, respectively. These are used in tandem to determine if a block has been cryptographically mined and is legitimate. The parent hash parameter is the Keccak 256-bit hash value of the parent's full block header. It has both nonce and mixhash in it. The timestamp parameter provided at the start of the block returns the Ubuntu OS time(t) output. The timestamp aids in verifying the chain's block order. The extraData parameter is a 32-byte extra space that may be used as needed. In the private chain, we subsequently create numerous Hyperledger nodes.

Each Hyperledger node launched will start a Javascript Console, and built-in objects execute tasks like mining, transactions, and querying blocks. Then, the Solidity programming language is used to create a smart contract that contains the initialization of the IoMT node accounts. This smart contract includes the storage and query functions for the data gathered. On the Hyperledger blockchain, the smart contract's information is accessible to all Hyperledger nodes and can interact with them. Therefore, it needs a storage system to share data easily. Any IoMT with blockchain access has unrestricted access to all data records. Although, it is expected that the data kept on the blockchain cannot be recorded arbitrarily since it leads to malicious IoMT nodes storing fraudulent and forged data. Hyperledger controllers can check data quality and establish a Central Authority to oversee the validity and data ownership of all IoMTs and maintain their public keys. The Certificate Authority (CA) of Hyperledger Fabric offers the following features:

1. To the user registry, CA connects with LDAP or registers identities.
2. Enrollment Certificates are issued (ECerts).
3. Certificate renewing and cancellation.

There are two ways of interacting with a Hyperledger Fabric CA server: via the Hyperledger Fabric CA client or through one of the Fabric SDKs. Therefore, we communicate with the CA server in one of two ways: (i) using the CA client or (ii) using the Fabric SDKs. The CA client is implemented at the CH level in the proposed HSN.

4.3 Analysis of trust value in normal and malicious condition

The involvement of trust in healthcare organizations should be primarily focused on evaluating the efficacy of the proposed trust management system. The Sensor Network Simulator and Emulator (SENSE) [45] is used for emulation and a simulation for HSN networks to analyze the trust value in normal and malicious conditions. We implement the rule-based open-source, lightweight version of mobile Snort IDS in HSN. This study compares and contrasts two similar trust management schemes: EHDTE [46] and BTEM [7]. The former used a trust-aware algorithm and intelligent trust management system to recognize insider malicious IoMT nodes.

4.3.1 The trust value in normal condition

Foremost, this research examines network traffic in its normal state. The aggregate trust levels of HSN IoMT nodes are shown in Fig. 9. Set the value of the forgetting factor to 0.8 per recent research based on its impact analysis by Zhang et al. [41]. Once the central server finished collecting data, the trust values seemed to become stable, i.e., extremely close to one, after some time. However, achieving one according to the proposed algorithm 1 is extremely difficult due to communication delays and insider DoS attacks.

4.3.2 The trust value in malicious condition

The analysis aims to see how well our blockchain-enabled trust management strategy performs under malicious conditions. We randomly chose three IoMT nodes to transmit malicious packets to other IoMT nodes in HSN to initiate internal attacks. The malicious traffic provoked by DoS assaults software [47] can transmit various modified packets. Adjust the forgetting factor to 0.8 per recent research based on its impact analysis by Zhang et al. [41].

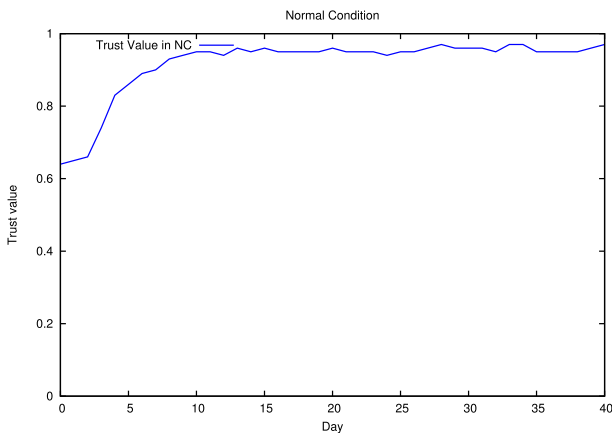


Fig. 9 Average trust value in normal condition

Comparison of trust value with other trust management schemes We compare the trust value of our proposed solution with two comparable trust management methods (EHDTE [46], and BTEM [7]) during DoS malicious attacks. The proposed challenge-based trust model evaluated the satisfaction between desired replies and gained information to detect suspicious nodes in HSN. The malicious IoMT nodes and the central server were placed in the same experiment environment. Insider DoS attacks began when trust values in HSN stabilized. Repeat the test three times to limit the influence of unexpected circumstances. Figure 10 depicts the aggregate trust levels of malicious IoMT nodes. The following are the main points to consider.

1. According to most trust management methods, the trust values of malicious IoMT nodes started falling once the attack was launched. It was because of the forgetting factor that highlights the behavior of the IoMT node. The reputation of EHDTE Trust model drops quicker than BTEM Trust model.
2. EHDTE and BTEM Both trust models reduce the reputation level slower than our proposed cluster-based hierarchical trust management scheme (without a blockchain-based trust management approach). It happens because the hierarchical cluster's trust model calculates trust values based on packet status, which is quite sensitive to (malicious) traffic status. On the other hand, BTEM trust model experience a communication delay since it must first collect input from target nodes before evaluation.
3. When comparing our proposed blockchain-enabled trust model to the conventional hierarchical cluster-based trust management scheme. We discovered that the proposed method enhances the efficiency of IoMT malicious node detection in HSN contexts one day earlier than its original (cluster-based hierarchical trust management scheme, without blockchain). However, the proposed model faster degrades the trust value of malicious IoMT nodes below the threshold of 0.8. Our technique uses the blockchain to allow HSN IoMT nodes to update their blocklist

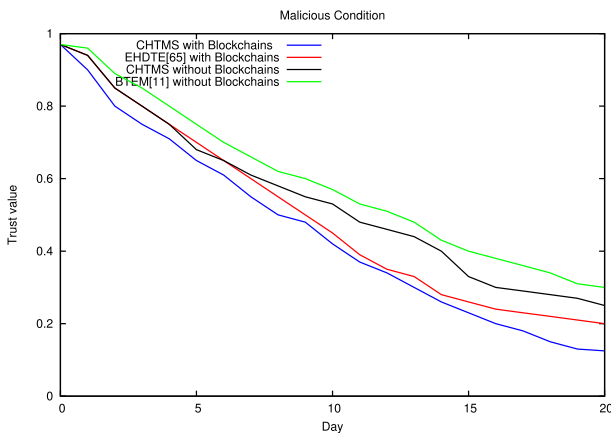


Fig. 10 Average trust value in malicious condition

faster than the original system, and adjacent nodes interact with suspicious nodes to get additional traffic updates.

Furthermore, our experiment results show that our proposed blockchain-based trust management scheme provides better malicious IoMT node detection performance than the original scheme. The other two comparable trust management schemes are based on the pattern of malicious nodes' reputation in the HSN environment. Therefore, the scalability of the proposed approach is relatively high in healthcare environments.

4.4 System performance analysis undergoing DoS attacks

The comparative analysis of Blockchain vs. Conventional MySQL database performance was recorded during HSN's simulated DoS attacks. We randomly chose two, three, and four IoMT nodes to transmit malicious packets to other IoMT nodes in HSN to initiate internal attacks. The modified Python script of the IDS testing tool [47] is executed to generate different types of airjack beacon malicious traffic in the network. During DoS attacks, the resource consumption is sending queries to MySQL clients at port no 3306 and communicating with three Fabric blockchain nodes (two mining nodes and the other just a transaction receiving node). In HSN, simulate malicious activity by frequently querying the Fabric nodes to get the most recent block information. On the other hand, querying MySQL for the total amount of entries in the table causes it to be delayed and yields invalid storage transactions. We experimented with two distinct levels of DoS attacks: Low-Intensity and High-Intensity DoS attacks. The Low-Intensity DoS attacks use ten concurrent processes on ten connected IoMT devices, while High-Intensity DoS attacks use twenty concurrent processes on twenty connected IoMT devices.

Both blockchain and MySQL nodes utilized the PTB-XL ECG dataset consisting of 21837 diagnostic 12-lead ECGs, each lasting 10 s, from 18885 patients. PTB-XL is an extensive electrocardiography dataset freely accessible on the Kaggle platform [48]. It is a vital diagnostic technique for determining a patient's heart state through IoMT-enabled ECG. The Schiller AG devices collect the waveform data that makes up the PTB-XL ECG dataset [48]. At the Physikalisch-Technische Bundesanstalt (PTB), the records were vetted and transformed into a systematically structured database. THE PTB-XL ECG dataset followed SCP-ECG standards while creating the ECG statements to annotate the recordings.

MySQL takes advantage of the same data exchange module and execution platform that Blockchain nodes do. As a baseline, we use MySQL database version 8.0 on Ubuntu 20.04.3 LTS for x86_64 architecture, with an Intel Core 3.40 GHz i7-6700 CPU and 16 GB of RAM, as a comparison to our blockchain-enabled solution. The Python script is executed after every time slot. All IoMTs submit the recorded data to MySQL by sending queries to clients at port no 3306 and communicating with three Fabric blockchain nodes (two mining nodes and the

other is just a transaction receiving node). In HSN, we simulate malicious activity by frequently querying Fabric nodes to get the most recent block information. To uniquely identify patients and ECG records, we constructed a table with two columns (ecgID and patientID). We also query the complete list of records using MySQL's COUNT(*) function and get the total number of entries in the table. The efficiency of this solution is measured using the two following metrics.

1. A transaction failure ratio (TFR) is calculated by querying the MySQL database and Hyperledger blockchain against the total transactions in each timeslot. We compute the TFR ratio of unavailable data whose data transaction requests may be blocked/delayed or canceled. TFR is calculated as $TFR = 1 - \Delta T/S$, where ΔT is the total number of currently processed transactions in the database or blockchain. S is the total number of transaction storage requests, calculated as $S = M * ts$. M is the number of IoMT nodes in HSN, and $ts = 1, 2, \dots TS$. TS is the number of transaction storage requests by each IoMT. The range of TFR is (0 – 1). When all the transactions are completed and stored in the database or blockchain, TFR is 0. When all the transactions are blocked/delayed or canceled and unavailable/stored in the database or blockchain, TFR is 1.
2. Transaction Successful Ratio (TSR) is the ratio between the inserted data ΔI and the number of data storage requests found by counting the total number of transaction queries. TSR is computed when the Hyperledger blockchain or MySQL database does not process queries, and IoMT nodes stop submitting requests $TSR = \Delta I/S$. In this case, $S = M * TS$. The range of TSR is (0–1). When all the transactions are completed and stored in the database or blockchain, TSR equals 1. When all the transactions are blocked, delayed or canceled and unavailable or stored in the database or blockchain, TSR is 0.

Overall, the TFR and TSR ratios indicate the outcomes of the system operating under two malicious conditions (low-intensity and high-intensity attacks). The TFR is computed immediately after processing each batch of ten transactions, while the system remains operational because it accepts and executes ongoing data requests. On the other hand, TSR is computed whenever all the system tasks are finished, i.e., when the IoMT nodes have performed the diagnostic process. The storage system has stopped handling requests. In some situations, if a transaction request fails while calculating TSR, it may be denied or canceled, so it is considered TFR unless the reverse is true. TFR indicates the system's transient reaction characteristic, whereas TSR represents robustness. Figure 11, demonstrates the impact of DoS attacks on the immediate transaction failure ratio by modifying the number of IoMT nodes (two, three, and four) and the intensity of attacks (low and high). The following remarks are based on our observations.

1. When the count of storage transactions grows by IoMT nodes, some parts may not be retrieved from the entire data, regardless of its storage mechanism. In the event of a DoS attack, some data storage queries will be restricted, no matter if the data is stored in a conventional database or on a blockchain.

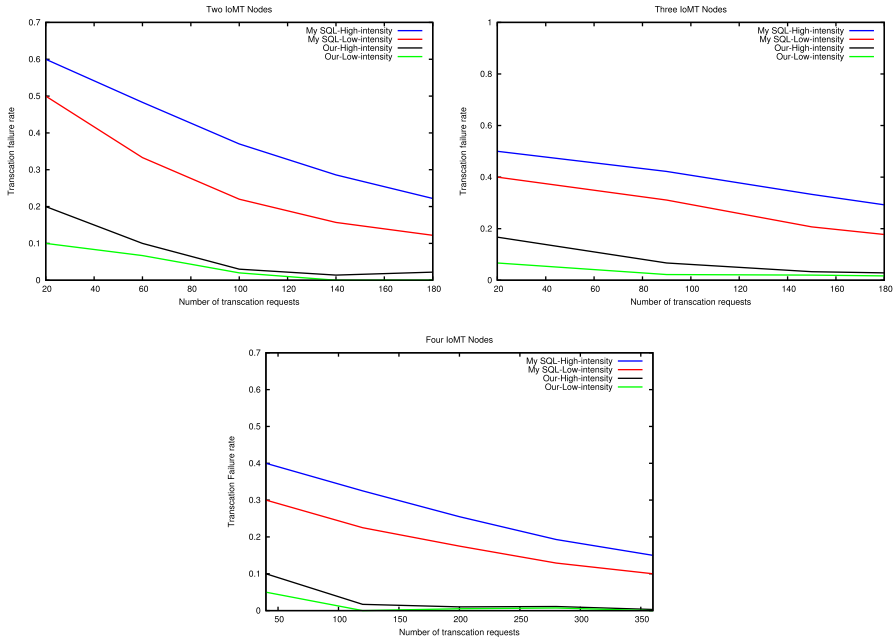


Fig. 11 Two, three and four IoMT nodes undergoes DoS attacks impacts the immediate TFR

2. During DoS attacks, increasing the number of IoMT nodes and intensity level of attacks, the upper limit of TFR is 0.3 and 0.7 for conventional databases and blockchain, respectively. In this way, blockchain has lower TFR than a conventional database, with the exact causes as discussed in the analysis of DoS attacks.
3. As demonstrated in Fig. 11., during low-intensity DoS attacks, the TFR is 0.1 in blockchain and 0.5 in database. During high-intensity DoS attacks, the TFR increases by 0.2 in the blockchain and 0.63 in the database due to increased blocked requests and the intensity of DoS attacks.
4. The number of IoMT nodes increased from two to four at any moment, and the rate of TFR drop is high; with the low-intensity DoS attacks, it dropped from 0.2 to 0.05 and 0.6 to 0.45 for blockchain and database, respectively. However, the same drop rate is also found in high-intensity DoS attacks. As discussed during the analysis of DoS attacks, more IoMT nodes yield excessive transaction requests, resulting in a lower TFR.

Last, Fig. 12. Demonstrates how conventional databases and blockchains perform under DoS attacks. Our observations are given below.

1. The TSR of the blockchain is not affected by DoS attacks; it always lies on the top. Conversely, the TSR of the database affected by DoS attacks improves whenever the number of transaction requests increases. The reason is the same as discussed in the TFR analysis. The less data exposed in DoS attacks increase the transaction request count in the IoMT multimode structure. It leads to a sluggish improve-

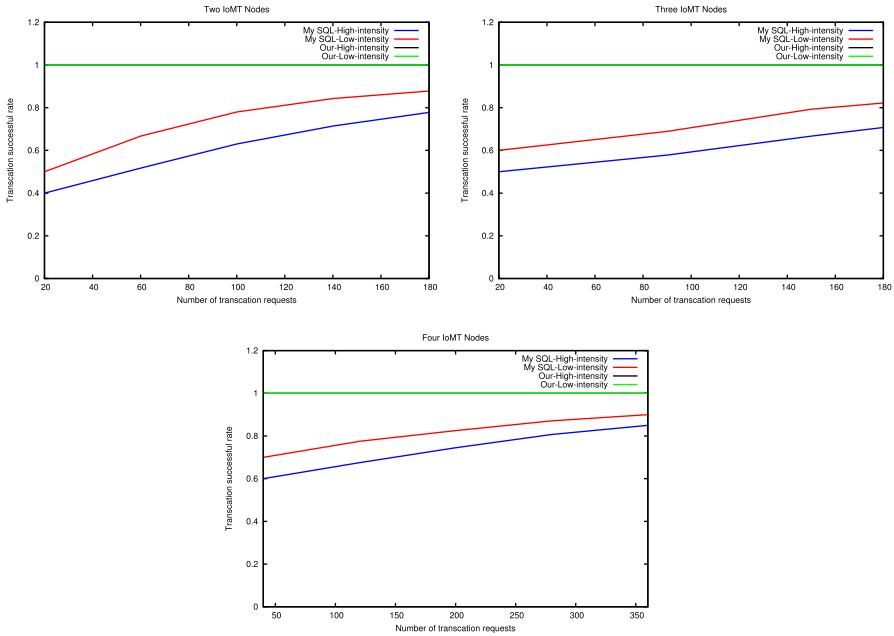


Fig. 12 Two, three and four IoMT nodes undergoes DoS attacks impacts the immediate TSR

ment in the TSR of the database. At the same time, the TSR of the blockchain remains unaffected.

2. When the number of IoMT nodes increases, the corresponding number of database storage requests also increases, so the TSR of the database improves. At the same time, the TSR of the blockchain remains unaffected by any intensity of DoS attacks.

4.4.1 Security resilience analysis

To examine the proposed CHTMS scheme’s resilience to attacks, we divide the IoMT nodes into reliable and unreliable nodes. It is the case that reliable nodes communicate successfully and always provide genuine recommendations. On the other hand, malicious nodes strive to engage in as much unsuccessful communication as possible while sending false recommendations regarding trustworthy nodes. The distinction between healthy and compromised nodes is subjective. A node may be an exemplary node in the perception of one node but a compromised node in the perception of another.

To examine this idea more precisely, record the behavior of untrustworthy nodes, and simulate how they try to gain an unfair advantage in the CHTMS trust model. By applying the same approach to a higher level of cluster hierarchy, CHTMS’s trust management approach addresses the resilience to the malicious behavior of a

compromised node. A malicious node proceeds by exploiting unfair benefits and compromised behavior. When dealing with other reliable nodes in the same cluster, a malicious node is to have as much failed communication as possible while maintaining specific goals:

1. Keep itself high trust value than the original computed trust values; more significantly, keep itself in the trusted zone while its proper location is in the unsure or untrustworthy zone.
2. Good node's trust value decreased and forwarded to neighbor and upper level.
3. The trust value of the compromised node is always kept high.

When it prevents compromised nodes from achieving their pre-specified objectives, trust management at the node level seems resilient. To ensure the HSN's security, specify the compromised nodes' boundaries. More specifically, the trusted node must keep the number of successful communications equal to or greater than failed communications.

4.5 Analyze communication cost in HSN

In the worst-case scenario, each IoMT node wants to interact with all the IoMT nodes in their group. Each IoMT group wants to interact with all the groups in the HSN network. Suppose the HSN network comprises $|M|$ groups, and the average group size is η . When an IoMT node A intends to communicate with other IoMT nodes B via intragroup communication, then, IoMT node A will request a maximum of $\eta - 2$ peer recommendation. In response, an IoMT node A gets $\eta - 2$ replies. The greatest communication overhead for each IoMT node A is $2(\eta - 1)(\eta - 2)$ to communicate with all IoMT nodes in the group. The highest intragroup communication cost (IC_{intra}) in proposed CHTMS is $2\eta(\eta - 1)(\eta - 2)$ if all IoMT nodes intend to connect with every IoMT node in HSN.

When an IoMT group A intends to communicate with other groups B , it will submit the strongest single peer recommendation request to BS during intergroup communication. As a result, every request requires a two-packet communication cost. The highest communication cost can be $2|N| - 1$ packets if group A tends to communicate with every group $|M|$ in HSN. The highest intergroup communication cost (IC_{inter}) of the CHTMS is $2|N||N| - 1$ when every group tends to communicate with every group in HSN. As a result, the CHTMS's highest communication cost (CC) of HSN is:

$$\begin{aligned}
 CC &= [No. \text{ of groups}(N) * \text{Intragroup communication cost}(IC_{intra})] \\
 &\quad + \text{intergroup communication cost}(IC_{inter}) \\
 &= [|N| * 2\eta(\eta - 1)(\eta - 2)] + 2|N||N| - 1 \\
 &= 2|N|(\eta(\eta - 1)(\eta - 2) + (N - 1))
 \end{aligned} \tag{11}$$

Table 5 Intra-cluster communication cost of different trust management schemes in worst case

Trust management scheme	Communication cost
AF-TNS [17]	$2 N (\eta(\eta - 1)^2 + (N - 1)^2)$
BTEM [7]	$4 N (\eta(\eta - 1) + (N - 1)(N - 2))$
EHDTE [46]	$2 N (\eta(\eta - 1)(\eta - 2) + (N - 1))$
CHTMS	$2 N (\eta(\eta - 1)(\eta - 2))$

4.5.1 Comparison of communication cost with other schemes

The communication cost of different trust management techniques for a large number of IoMT nodes ($N * \eta = 100$ nodes) form N number of clusters with similar cluster sizes ($\eta = 10$) is shown in Table 5. Compared to state-of-the-art approaches, CHTMS imposes less communication cost even when clusters inside the HSN network grow. It was also found that CHTMS is appropriate for large-scale HSNs with small cluster sizes (10 IoMT in each cluster). The crucial point concerning the BTEM scheme [7] is whether it displays the results of one transaction per node. For instance, when IoMT node A and IoMT node B want to communicate, they initially share four packets. The trust is computed again after finalizing the transaction when node A wishes to begin another transaction with B. As a result, the BTEM scheme's communication costs will rise quarterly with each transaction. In the CHTM scheme, an IoMT node A wishes to initiate the new transaction with IoMT node B only when the first transaction is completed; no extra communication cost occurs since IoMT node A already computes the trust by considering the history of the previous transaction(s).

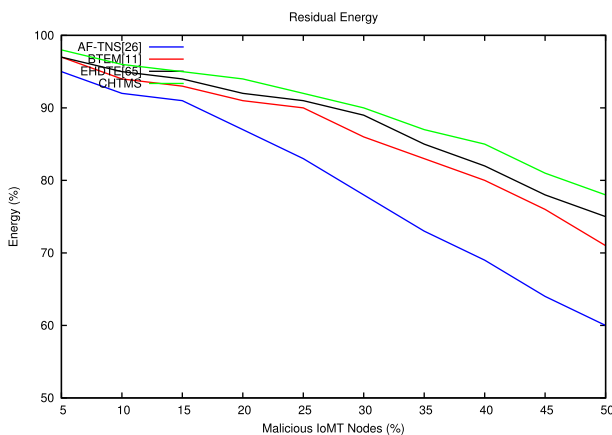


Fig. 13 Comparison of residual energy with other schemes

4.5.2 Comparison of residual energy and throughput with other schemes

The energy a node uses will determine how long the network will continue functioning. The network's lifespan will be reduced in proportion to the amount of energy consumed. Figure 13 depicts the malicious node's effect on the node's residual energy consumption. The findings demonstrate that CHTMS is superior to other approaches of a similar nature in terms of throughput and consuming less energy that is preserved than other comparative schemes like AF-TNS [17], EHDTE [46], and BTEM [7]. This is because CHTMS considers the blockchain-based trust management system of IoMT nodes in the HSN network. Figure 14 depicts a comparison of the average throughput. The influence of throughput is examined by the number of transactions completed per unit of time. The percentage of malignant nodes ranges from 5 to 50 percent.

4.6 Analyze CPU utilization in HSN

Due to the advent of trust management procedures in HSN, it is realistic to expect additional computational effort on both the IoMT Node and Base-Station server sides.

1. Interactions with the HSN server, including monitoring of packet status, blocklist updating, and communication between IoMT nodes and Base-Station, are the significant causes of CPU load at the Base-Station server.
2. The leading causes of forging the CPU load at the smartphone and IoMT node side are interactions with the HSN IoMT node. These interactions include communication with HSN IoMT nodes, trust measurement, security policy implementation, blocklist construction and updating, and blockchain data retrieval.

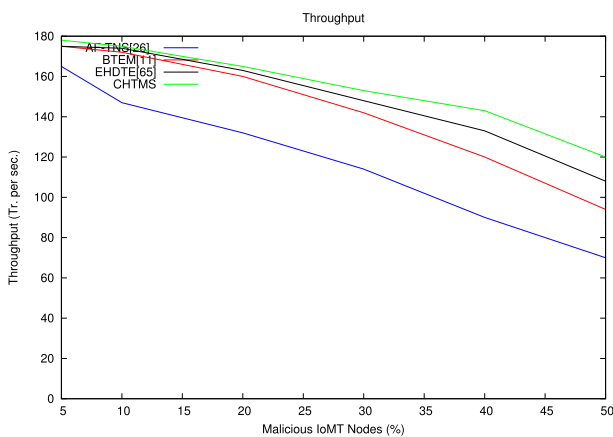


Fig. 14 Comparison of throughput with other schemes

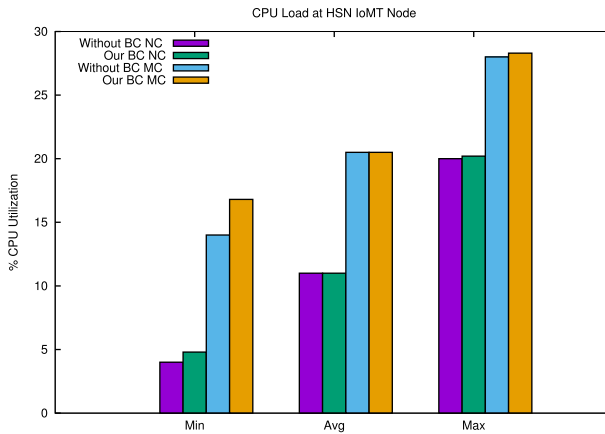


Fig. 15 CPU Utilization at HSN IoMT node

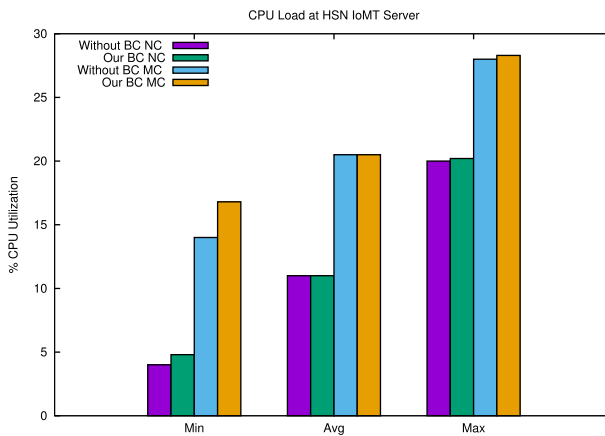


Fig. 16 CPU Utilization at HSN base-station server

In this observation, Figs. 15 and 16 exhibit CPU utilization at IoMT nodes and the Base-Station server in healthcare organizations under Normal and Malicious Conditions, covering the highest, lowest, and average CPU utilization.

The following are critical observations of the results.

1. It is clear that the CPU demand was significantly higher under the Malicious Condition (MC) than in the normal state. As per observation, the average CPU utilization of the Base-station server is 20.3 percent in normal conditions, but it rises to 30.6 percent during malicious conditions. However, in this case, malicious

traffic can generate more significant interaction between smartphones, HSN IoMT nodes, and the Base-station server.

2. The CPU load on the Base-Station server was significantly more than the smartphone and IoMT node side. The result represents 20.3 percent and 11 percent CPU loads for HSN's Base-Station server and smartphone/IoMT nodes. The Base-Station server is responsible for various services, including data gathering, trust computation and assessment, blocklist production, and updating.
3. It has been found that our blockchain-based trust management technique boosts CPU utilization compared to the original trust management scheme without blockchain involvement. The average CPU utilization is 11 percent and 20.5 percent at the smartphone/IoMT node level, in normal conditions and malicious conditions, respectively, while 20.3 percent and 30.6 percent at the Base-Station server level, in normal conditions and malicious conditions, respectively.
4. The CPU workload between our blockchain-enabled technique and the original scheme (without blockchain) slightly increases initially, less than 1 percent in the case of average CPU utilization at both the IoMT node and BS-Server level. Therefore, we adapted the blockchain-enabled technique because it does not significantly increase CPU workload compared to many other security-related benefits over the original scheme.

5 Conclusion and future directions

The extensive usage of IoMTs in HSN has become a prominent target for attackers, who can compromise multiple healthcare resources inside the network. Moreover, insider attacks pose a significant danger to IoMT, necessitating effective trust management strategy implementation in healthcare. We proposed an algorithm for accurately collecting data from IoMT devices and exchanging patient data securely across IoMT nodes and smartphone devices. We integrate the Hyperledger blockchain into HSN, backed by a robust and lightweight CHTMS. It reduces the cost of trust analysis for identifying insider malicious/compromised IoMT nodes and updating blocklists across HSN more efficiently. Our proposed CHTMS scheme provides better security, stability, and more resistance to malicious attacks such as DoS and DDoS attacks. We find that the performance of the proposed approach is better in terms of TFR, TSR, CPU workload, and communication cost compared to state-of-the-art approaches.

Limitations and future research directions: To preserve patient health records privacy, we need to hide IoMT node identities. However, establishing and maintaining a trusted anonymized healthcare environment is still challenging. In the proposed HSN simulation environments, we implemented high-intensity DoS attacks employing 20 nodes, while in reality, the number of IoMT nodes may be more.

Acknowledgements We acknowledge the support of the Indian Institute of Technology (ISM), Dhanbad, India, in this work. In appreciation of the continuous support provided to the authors by the department of CSE at IIT (ISM) in Dhanbad, India, the authors would like to extend their heartfelt gratitude to the department.

Author Contributions (Author A[1]. Performed experimental work, wrote the manuscript, and prepared figures used in the manuscript. The other authors are B[2], C[3], and D[4]. Carefully review the manuscript.)

Funding Not applicable.

Data availability We used the PTB-XL ECG dataset publicly online, accessible on the Kaggle platform [48].

Declarations

Conflict of interest The authors reported no potential competing interest.

References

- Zhang M, Cui X, Li N (2022) Smartphone-based mobile biosensors for the point-of-care testing of human metabolites. *Mater Today Biol* 100254
- Guo J (2017) Smartphone-powered electrochemical biosensing dongle for emerging medical IoTs application. *IEEE Trans Ind Inf* 14:2592–2597
- Yang Z, Zhou Q, Lei L, Zheng K, Xiang W (2016) An IoT-cloud based wearable ECG monitoring system for smart healthcare. *J Med Syst* 40:1–11
- Alkinoon M, Choi SJ, Mohaisen D (2021) Measuring healthcare data breaches. In: *International Conference on Information Security Applications*. Springer, pp 265–277
- Yu E (2018) Singapore suffers 'most serious' data breach, affecting 1.5 m healthcare patients including prime minister
- Wei L, Yang Y, Wu J, Long C, Li B (2022) A comprehensive study. *IEEE Internet Things J*
- Anwar RW, Zainal A, Outay F, Yasar A, Iqbal S (2019) Btem: belief based trust evaluation mechanism for wireless sensor networks. *Futur Gener Comput Syst* 96:605–616
- Chen L, Lee W-K, Chang C-C, Choo K-KR, Zhang N (2019) Blockchain based searchable encryption for electronic health record sharing. *Futur Gener Comput Syst* 95:420–429
- Sharma A, et al (2020) Blockchain technology and its applications to combat covid-19 pandemic. *Res Biomed Eng* 1–8
- Meng W, Li W, Xiang Y, Choo K-KR (2017) A Bayesian inference-based detection mechanism to defend medical smartphone networks against insider attacks. *J Netw Comput Appl* 78:162–169
- Liu Y, Wang J, Yan Z, Wan Z, Jäntti R (2023) A survey on blockchain-based trust management for internet of things. *IEEE Internet Things J*
- Meng W, Tischhauser EW, Wang Q, Wang Y, Han J (2018) When intrusion detection meets blockchain technology: a review. *IEEE Access* 6:10179–10188
- Bhan R (2022) Hyperledger fabric for healthcare smartphone network. <https://github.com/Ratibhan/fabric.git>
- Zhou H, Wu Y, Feng L, Liu D (2016) A security mechanism for cluster-based WSN against selective forwarding. *Sensors* 16:1537
- Umar IA, Hanapi ZM, Sali A, Zulkarnain ZA (2017) Trufrix: a configurable trust-based cross-layer protocol for wireless sensor networks. *IEEE Access* 5:2550–2562
- Jiang J, Han G, Shu L, Chan S, Wang K (2015) A trust model based on cloud theory in underwater acoustic sensor networks. *IEEE Trans Ind Inf* 13:342–350
- AlFarraj O, AlZubi A, Tolba A (2018) Trust-based neighbor selection using activation function for secure routing in wireless sensor networks. *J Ambient Intell Hum Comput* 1–11
- Liu H, Zhang Y, Yang T (2018) Blockchain-enabled security in electric vehicles cloud and edge computing. *IEEE Netw* 32:78–83
- Nguyen GN, Le Viet NH, Devaraj AFS, Gobi R, Shankar K (2020) Blockchain enabled energy efficient red deer algorithm based clustering protocol for pervasive wireless sensor networks. *Sustain Comput Inf Syst* 28:100464
- Honar Pajooh H, Rashid M, Alam F, Demidenko S (2021) Multi-layer blockchain-based security architecture for internet of things. *Sensors* 21:772

21. Kumar R et al (2019) A multimodal malware detection technique for android IoT devices using various features. *IEEE Access* 7:64411–64430
22. Tariq N et al (2020) A blockchain-based multi-mobile code-driven trust mechanism for detecting internal attacks in internet of things. *Sensors* 21:23
23. She W et al (2019) Blockchain trust model for malicious node detection in wireless sensor networks. *IEEE Access* 7:38947–38956
24. Hammi MT, Bellot P, Serhrouchni A (2018) Bctrust: a decentralized authentication blockchain-based mechanism. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp 1–6
25. Khatoon A (2020) A blockchain-based smart contract system for healthcare management. *Electronics*. <https://doi.org/10.3390/electronics9010094>
26. Srivastava G, Crichigno J, Dhar S (2019) A light and secure healthcare blockchain for IoT medical devices. In: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). IEEE, pp 1–5
27. Yáñez W, Mahmud R, Bahsoon R, Zhang Y, Buyya R (2020) Data allocation mechanism for internet-of-things systems with blockchain. *IEEE Internet Things J* 7:3509–3522
28. Namasudra S, Sharma P, Crespo RG, Shanmuganathan V (2022) Blockchain-based medical certificate generation and verification for IoT-based healthcare systems. *IEEE Consum Electron Mag*
29. Abdellatif AA et al (2020) sshealth: toward secure, blockchain-enabled healthcare systems. *IEEE Netw* 34:312–319
30. Jeet R, Kang SS, Safiu Hoque SM, Dugbakie BN (2022) Secure model for IoT healthcare system under encrypted blockchain framework. *Secur Commun Netw* 2022
31. Dai H-N, Imran M, Haider N (2020) Blockchain-enabled internet of medical things to combat covid-19. *IEEE Internet Things Mag* 3:52–57
32. Nimbalkar P, Kshirsagar D (2021) Analysis of rule-based classifiers for IDS in IoT. In: *Data Science and Security*. Springer, pp 461–467
33. Satheesbabu S, Gokulakrishnan P, Dhanalakshmi N (2021) The surveillance of intrusion detection systems and approaches. *Inf Technol Ind* 9:1135–1150
34. Alevizos L, Ta VT, Hashem Eiza M (2021) Augmenting zero trust architecture to endpoints using blockchain: a state-of-the-art review. *Secur Privacy e191*
35. Gu J, Lu S (2021) An effective intrusion detection approach using SVM with Naïve Bayes feature embedding. *Comput Secur* 103:102158
36. Li W, Meng W, Parra-Arnau J, Choo K-KR (2021) Enhancing challenge-based collaborative intrusion detection against insider attacks using spatial correlation. In: 2021 IEEE Conference on Dependable and Secure Computing (DSC). IEEE, pp 1–8
37. Marcus Y, Heilman E, Goldberg S (2018) Low-resource eclipse attacks on ethereum’s peer-to-peer network. *IACR Cryptol ePrint Arch* 2018:236
38. Rajput AR, Li Q, Ahvanooy MT (2021) A blockchain-based secret-data sharing framework for personal health records in emergency condition. *Healthcare*. <https://doi.org/10.3390/healthcare9020206>
39. Xu X et al (2021) Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain. *ACM Trans Internet Technol (TOIT)* 21:1–17
40. Han G, Jiang J, Shu L, Guizani M (2015) An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic sensor network. *IEEE Trans Mob Comput* 14:2447–2459
41. Zhang B et al (2023) A reputation mechanism based deep reinforcement learning and blockchain to suppress selfish node attack motivation in vehicular ad-hoc network. *Futur Gener Comput Syst* 139:17–28
42. Ji D, et al (2022) Smartphone-based electrochemical system for biosensors and biodetection. In: *Biomedical Engineering Technologies*. Springer, pp 493–514
43. Bhan R (2022) 256-bit elliptic curve cryptography (ECC) parameters secp256k1 associated with a Koblitz curve. <https://github.com/Ratibhan/ECC.git>
44. Vahdati Z, Yasin S, Ghasempour A, Salehi M (2019) Comparison of ECC and RSA algorithms in IoT devices. *J Theor Appl Inf Technol* 97
45. Chen G, Branch J, Pflug M, Zhu L, Szymanski B (2015) Sense: a wireless sensor network simulator. In: *Advances in Pervasive Computing and Networking*. Springer, pp 249–267

46. Sharma P, Namasudra S, Crespo RG, Parra-Fuente J, Trivedi MC (2023) Ehdhe: enhancing security of healthcare documents in IoT-enabled digital healthcare ecosystems using blockchain. *Inf Sci* 629:703–718
47. Blasco J (2022) Ids testing tool. <https://code.google.com/archive/p/wireless-intrusion-detection-system-testing-tool/>
48. Wagner P et al (2020) Ptb-xl, a large publicly available electrocardiography dataset. *Sci Dta* 7:154

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Rati Bhan¹ · Rajendra Pamula¹ · Parvez Faruki² · Jyoti Gajrani³

Rajendra Pamula
rajendra@iitism.ac.in

Parvez Faruki
parvezfaruki.kg@gmail.com

Jyoti Gajrani
jyotigajrani@ecajmer.ac.in

¹ Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Dhanbad 826004, India

² Department of Technical Education, Govt. of Gujarat, Gandhinagar, India

³ Department of Computer Science and Engineering, Engineering College Ajmer, Ajmer, India