Cluster Computing (2021) 24:1675–1694
https://doi.org/10.1007/s10586-020-03216-w

Smart contract based policies for the Internet of Things

Vikram Puri[1] · Ishaani Priyadarshini[2] · Raghvendra Kumar[3] · Chung Van Le[1]

Received: 18 June 2020 / Revised: 21 November 2020 / Accepted: 24 November 2020 / Published online: 1 January 2021

Abstract

Internet of Things (IoT) is one of the most powerful platforms that incorporates several other technological components within itself. The IoT ecosystem comprises devices, communications, protocols, analytics, cloud, automation, etc. Its magnitude keeps on increasing with the addition of tools and services. While IoT has many advantages like connectivity, efficiency, and convenience, it is a known fact that privacy and security issues are prevalent in the IoT network. To minimize the security and privacy issues, we propose a blockchain-based solution. In this paper, we design policies based on smart contacts, which is a self-enforcing agreement embedded in computer code managed by a blockchain. We propose three different policies: Hardware and Device Security Policies, Access and Authentication policies, and Application security for the IoT network. Since blockchain-based solutions ensure trust and stability, this may be one of the most robust techniques to alleviate the IoT network's security and privacy issues. Also, we calculate the throughput and latency of the IoT enabled blockchain network and compare the power consumption of the IoT device at the time of data request with other proposed systems.

Keywords Internet of Things (IoT) · Security · Privacy · Blockchain technology · Smart contracts · IoT policies

1 Introduction

Internet of Things (IoT) may be defined as a network of connected objects that are capable of collecting and exchanging data. Since devices are embedded and connected to the Internet, IoT finds its use in many applications like smart cities, industrial automation, wearables, healthcare and wellness-based systems, smart home applications, etc. [1–3]. New IoT devices with undiscovered vulnerabilities and unauthorized devices may contribute to different security issues [4]. Some common security issues are related to device update management, system hardening, botnet attacks, device hijacking, integrity risks, and rogue devices [5–8]. Due to the increasing number of devices added to the IoT network daily, privacy is a significant concern. This is because devices that form a part of wearables, smart homes, or smart appliances are excellent surveillance equipment. This makes them potential targets for information collection as well as intrusive digital advertising. Moreover, lack of compliance and IoT security standards may be responsible for creating devices without sufficient security features. Therefore, two critical areas of concern in an IoT network are privacy and compliance (policies). An Internet of Things policy (IoT policy) is a document that acts as a comprehensive guide to assist an organization in promoting IoT development. It may also be used for dealing with the complex issues related to that development. Privacy is an important issue that must be accompanied by some policies for securing the IoT network. In this paper, we design policies to reduce security and privacy issues. To solve the privacy and security issue for IoT, several works have been proposed in the past. Paul et al. [9] introduced an assessment framework for IoT services, while Bouachir et al. [10] presented

✉ Raghvendra Kumar
raghvendra@giet.edu

Vikram Puri
purivikram@duytan.edu.vn

Ishaani Priyadarshini
ishaani@udel.edu

Chung Van Le
levanchung@duytan.edu.vn

[1] Center of Visualization and Simulation, Duy Tan University, Da Nang, Vietnam

[2] Department of Electrical and Computer Engineering, University of Delaware, Newark, USA

[3] Computer Science and Engineering Department, GIET University, Gunupur, Odisha 765022, India

an IoT framework for securing Smart Cyber Infrastructures like smart homes or smart buildings, and Chatfield and Reddick [11] introduced a cybersecurity framework for IoT enabled smart government. Liu et al. [12] suggested a security framework for home appliances in smart homes to prevent information leakage and home appliance hacking. Along with frameworks, different protocols were also recommended. Attkan and Ahlawat [13]proposed an authentication protocol clubbed with a session key generation scheme for wireless sensor networks in IoT platforms, and Aloqaily et al. [14] suggested an energy trade framework using smart contracts owing to concerns over privacy, security vulnerabilities, hacking attacks, and information loss [15]. Another way to minimize security issues is to introduce new authentication schemes [16–18] and cryptographic measures [19, 20]. While minimizing privacy and security issues is undoubtedly a goal in IoT networks, there is also a need to ensure trust, accountability, and stability. The public key infrastructure and distributed ledger in blockchain technology make it one of the principal candidates for securing the IoT [21]. Several research works have aimed at securing IoT using Blockchain Technology [22–25]. While Blockchain technology is preferred for data integrity [26, 65–67], some of its limitations are that it may not apply to a vast distributed computing system, and may be time-consuming. Hence, to alleviate the security and privacy concerns in an IoT network and ensure accuracy, transparency, speed, security, efficiency, and trust, we rely on Smart Contracts. Smart contracts may be defined as lines of code that are stored on a blockchain [27]. These are automatically executed in a situation where some predetermined terms and conditions are met [28]. The code controls the execution, and transactions are trackable and irreversible. We use smart contracts to devise security and privacy policies for the IoT network such that violation of policies may have an impact on the smart contract. The novelty of this paper is as follows:

1. While privacy and policies (compliance) are essential aspects of IoT networks, and past works highlight these, there is limited work that focuses on both the components together. In this paper, we fuse both the concepts and design policies for IoT security and privacy.
2. Blockchain-based research work done in the past tackles privacy issues, but the blockchain-based platform for designing policies is the IoT network is yet to be explored extensively, and in this paper, we propose the same.
3. Smart Contracts have been considered for IoT related research mainly in terms of access control and management but have not mentioned IoT privacy in detail, which is highlighted in this article.
4. Finally, using throughput and latency, we evaluate the blockchain network.

The rest of the paper is organized as follows. Section 2 details the background. In this section, we discuss related works, technologies used and IoT policies. Section 3 describes the methodology of the proposed system. Section 4 presents the Results and Discussion for the work carried out along with a comparative study. Finally, in Sect. 5, we present the conclusion.

2 Background

In this section, we present three subsections. In the first subsection, we present the background of the study and followed by a discussion on the related works of blockchain technology. In the third subsection, we present the proposed IoT policies.

2.1 Related studies

Barrera et al. [29] discussed the design of a standardized network security policy enforcement architecture for IoT devices. Network behavior for any consumer IoT devices must be predictable. Hence the proposed method is an automated approach to propose network security policies for devices. The presented architecture, which is scalable and effective, does not require vendor cooperation or changes to devices or cloud infrastructure. Although the scheme looks promising, there may be specific issues related to device connectivity, identification, and complexity of IoT devices. WAN connectivity may not allow consumers to control the communication channel. Halepoto et al. [30] presented research on retransmission policies for Efficient Communication in IoT Applications. Many IoT applications are time-critical and need to maintain quality of service, reliability, and availability. While TCP creates a single connection between two devices, it may decrease availability if there is a connection error. The article evaluated the Stream Control Transmission Protocol on a multipath connection environment where connection failures are common. Although the proposed technique increases throughput, SCTP has several limitations. A maximum of eight source IP addresses and eight destination IP addresses may be allowed in SCTP communication. It only supports static IP NAT, and the interface packets coming in must belong to the same zone. Thus, the proposed approach has scalability issues. Atlam et al. [31] proposed an eXtensible Access Control Mark-up Language (XACML) approach for designing Access Control Policies for the Internet of Things. The XACML is efficient and compatible with different platforms. It provides a distributed and flexible approach to work in the IoT environment. The limitation of this scheme is that XACML has a verbose control access scheme, which can deter policy writers from taking advantage of its features [32]. Mishra et al. [33] surveyed the analysis of IoT congestion control policies. The article highlighted transport layer protocols TCP and UDP, application protocols XMPP (Extensible Messaging and Presence Protocol), MQTT (MQ Telemetry Transport), and RESTful HTTP the analysis. The paper discussed various congestion control algorithms, their advantages and disadvantages. While the article presents an extensive survey, it restricts the research domain to congestion control problems in the IoT network. Besides, the report discusses policies but fails to recommend suitable policies or a policy framework.

Le et al. [34] presented a policy-based identification technique for IoT devices' vendor and type by performing DNS traffic analysis by using algorithms like Term Frequency - Inverse Document Frequency (TF-IDF) to the domain resolution process. Evaluation of the proposed approach on traffic data depicts that the technique can identify 84% of the instances. The accuracy was found to be 91% for the IoT devices' vendor. While the approach is interesting, the results are not enough to convince how a large volume of data will be handled efficiently. Pal et al. [35] suggested a policy-based access control mechanism for IoT in healthcare. An access control architecture has been proposed to authorize users to services while protecting valuable unauthorized access resources. The approach is supported by XACML, which has several limitations. Further, the proposed research is restricted to a particular domain that considers access control in the IoT network. Nobakht et al. [36] devised a policy framework IoT-NetSec for Network Security using OpenFlow. Since network traffic rate and volume are dynamic, there may be differences in IoT systems and computer networks' characteristics. The IoT-Net suggests a policy-based and fine-grained traffic monitoring framework for IoT devices. The article is confined to network security in IoT, and the prototype implementation has been evaluated using only three network service attacks. Al-Shaboti et al. [37] proposed an Automatic Device Selection and Access Policy Generation that relies on user preference. Identifying suitable devices that satisfy user preferences and defining the workflow's security policies are significant IoT issues. The paper uses heuristic search algorithms to find preferred devices for the workflow. Genetic Algorithm has been the best approach for the research; however, Genetic Algorithms may not guarantee an optimal solution. Ding et al. [38] proposed a novel Attribute-Based Access Control Scheme Using Blockchain for IoT. Traditional access control methods provide complicated access management and lack credibility due to centralization. Blockchain technology has been used to record the distribution of attributes to avoid single point failure and data tampering. The scheme can resist multiple attacks and can be implemented in IoT networks. While blockchain technology promote data integrity, transaction processing may be very slow. Lim et al. [39] suggested a Federated Reinforcement Learning for Training Control Policies. The method was suggested for Multiple IoT Devices environment and focussed on optimally controlling IoT devices supporting the expansion of the Internet. In this paper, multiple reinforcement learning agents have been used to learn the optimal control policy of IoT devices. These devices are of the same type yet incorporate different dynamics. Since applying independent reinforcement learning to each IoT device individually would be costly and time-consuming, a new federated reinforcement learning architecture has been proposed, such that each agent working on its independent IoT device shares their learning experience, leading to faster learning speed. The only limitation of the proposed work is that federated learning requires frequent communication between nodes during the learning process and relies on significant computing power and memory and high bandwidth connections to exchange parameters of the machine learning model. Chen et al. [40] introduced an incentive aware prevention system for fake news based on blockchain technology. The study underpins the concept of Proof-of-Authority. A weighted ranking algorithm serves as the incentive mechanism and guarantees scalability. However, fake news can also be determined and prevented using a privacy aspect, which the study does not discuss. Al Ridhawi et al. [41] suggested a solution based on blockchain for decentralized service composition solutions concerning complex multimedia service delivery to cloud subscribers. The study does not rely on intermediary service to authenticate and deliver services and uses reinforcement learning to create secure and reliable paths. Reinforcement learning in excess may lead to the diminishing of results. Further, it requires heavy computation. Khalid et al. [42] introduced a blockchain-based authentication system for the IoT environment, along with an access control mechanism. The study has been performed for fog computing and public blockchain. The evaluation has been performed using apt parameters; however, the amount of energy consumed is quite large. Kumar et al. [43] presented an extensive survey on blockchain-based databases for IoT platforms along with its challenges. The study discusses the technology behind bitcoin or the Bitcoin Backbone Protocol and the consistency models. The study reports that the database does not satisfy the models; hence a solution is proposed. The limitation of the study is the narrow scope involving the cryptocurrency aspect of blockchain technology. Pavithran et al. [44] recommended a blockchain framework for the IoT environment, and identified key components and challenges concerning the same. The study also determines security issues related to blockchain frameworks for IoT. After simulating two blockchain implementations, it was found that the device to device architecture performs better, although the proposed work fails to provide confidentiality. Miloslavskaya et al. [45] proposed a blockchain-based system called the IoT-BlockSIEM for Security Information and Event Management (SIEM). The proposed framework is capable of performing numerous security controls and event detection. It is also responsible for collecting raw data for processing. The limitation of this research is that it considered only one aspect, i.e., Incident Management. Table 1 depicts the summary of the related works along with the advantages and disadvantages.

Based on the literature survey of the past research works followed by a critical analysis of the mentioned works, it is evident that IoT policy, in general, is yet to be explored. In the past, many research works considered specific domains of IoT policy, like access control or congestion control. The security and privacy issues have been prevalent in IoT for a while. The methods suggested evading privacy and security problems bt mean of proposed frameworks and architectures, machine learning techniques, blockchain technology, and protocols. However, few research works tackle the issue of security and privacy in IoT by designing policies and framwork [43–45]. Moreover, designing policies for IoT privacy and preservation using Smart Contracts is yet another innovative approach. Finally, the scope of designing policies is limited in most of the research works since they deal with specific policies.

2.2 Technologies used

In this section, we explain the working of overall blockchain technology. We list out the components that we have relied on for designing IoT privacy policies.

2.2.1 Blockchain technology

Blockchain may be defined as growing lists of records or blocks. These records are linked through cryptography such that every record or block incorporates a cryptographic hash of the previous block, a timestamp, and transaction data as a distributed ledger can record transactions between two parties efficiently in a verifiable and permanent way. Once a block has been added to the end of the blockchain, it is challenging to go back and alter the block's contents. This is because every block contains its hash, along with the hash of the previous block. Hash codes are created mathematically, such that digital information is converted into a string of numbers and letters. Modifying the information in any way results in a change for the hash code. Therefore, for changing a single block, an adversary must change every single block after it on the blockchain. This would lead to serious computation power as it involves recalculating all the hashes. Therefore, adding a block to the blockchain makes it extremely difficult to edit and almost impossible to delete. For addressing the issue of trust, there are tests known as consensus models, which are performed by blockchain networks for computers that want to join and add blocks to the chain. Consensus models require users to prove themselves before engaging in a blockchain network. Bitcoin employs a proof of work system which encourages computers proving that they have done work. This is usually achieved by solving a complex computational math problem. A computer solving one of these problems becomes eligible for adding a block to the blockchain. Blockchain technology finds its applications in banking sectors, cryptocurrencies, healthcare, smart contracts, supply chain uses, etc. [68, 69]. Typical advantages of blockchain technology include improved accuracy, cost reduction, and transparency. Blockchain is hard to tamper with, and the transactions are secure, private, and efficient.

2.2.2 Merkle root

A hash tree, or the Merkle tree, is responsible for efficiently and securely encoding the blockchain data. It leads to quick verification of blockchain data. This is because large amounts of data can be moved quickly from one computer node to another on the peer-to-peer blockchain network. Every transaction that takes place on the blockchain network has a hash associated with it. However, these hashes are not stored in sequential order on the block. They are preferably stored in the form of a tree-like structure. Each hash is linked to its parent in the tree-like structure, which makes it identical to a parent-child tree-like relation. A particular block may store numerous transactions, and all these transaction hashes in the block are also hashed. The result of all these hashes is a Merkle root. A Merkle root may be defined as the resulting hash of all the hashes of all the transactions. These transactions are part of a block in a blockchain network, and the Merkle tree is a sophisticated approach to verify the data. The single root hash at the top of the Merkle tree is connected to two hashes at level one. Each of these hashes at level one is further connected to two more hashes at level three, and a chain builds on. The structure formed relies on the number of transaction hashes. Lowest Level nodes are hashed first, such that all four hashes are included in the hash of nodes. The hash of these nodes is linked at level one. After that, hashing continues at level one such that hashes keep on reaching higher levels, until the top root hash, which is the Merkle root. Because of the tree-like linkage of hashes that began at

Table 1 Summary of the Related Works

Authors and year	Research and methodology	Advantages	Disadvantages
Barrera et al. (2018) [29]	Network security policy architecture for IoT	Effective, scalable, does not require vendor cooperation	Device connectivity, identification, WAN connectivity issues
Halepoto et al. (2018) [30]	Retransmission policies for efficient communication (IoT)	Increased throughput	Limited destination IP address, only supports static IP NAT, scalability issues
Atlam et al. (2018) [31]	eXtensible access control markup language for access control policies	Efficient, compatible, flexible	Verbose control access schema
Mishra et al. (2018) [33]	Study on IoT congestion control policies	Extensive survey	No policy framework discussed
Le et al. (2019) [34]	Policy-based identification technique	Accuracy up to 91	The proposed technique may not handle a large volume of data
Pal et al. (2019) [35]	A policy-based access control mechanism	Protecting valuable resources from unauthorized access	The very narrow scope of research
Nobakht et al. (2019) [36]	Policy framework IoT-NetSec for network security	the fine-grained traffic monitoring framework	Confined to network security, prototype implementation has been evaluated using only three network service attacks
Al-Shaboti et al. (2019) [37]	Automatic device selection and access policy generation	Identifying suitable devices	Genetic algorithms may not guarantee an optimal solution
Ding et al. (2019) [38]	Attribute-based access control scheme	Can resist multiple attacks	Slow transaction processing
Lim et al. (2020) [39]	Federated reinforcement learning for training control policies	Faster learning speed	Requires significant computing power and memory, and high bandwidth connections
Chen et al. (2020) [40]	Incentive aware prevention system for fake news	Scalability	Does not discuss privacy
Al Ridhavi et al. (2020) [41]	Blockchain-based solution for decentralized service composition for complex multimedia service delivery to cloud subscribers	No intermediary service required for authentication	High computation cost
Khalid et al. (2020) [42]	Blockchain-based authentication system	Evaluation depicts efficiency	High energy consumption
Tseng et al. (2020) [43]	An extensive survey on blockchain-based databases	The proposed solution for the database to satisfy models	Very narrow scope of research
Pavithran et al. (2020) [44]	Blockchain framework for the IoT environment	Device to device architecture performs better	Architecture fails to provide confidentiality
Miloslavskaya and Tolstoy (2020) [45]	Proposed IoTBlockSIEM	The system performs security information and event management	The narrow scope of research only focuses on Incident Management

the lowest level, the Merkle root at the top contains all the information about every transaction hash in the block. The single-point hash value is primarily useful for validating that block as it speeds up the verification process. Because this single hash value incorporates information about the entire tree, there is a need only to verify the transaction hash and its sibling-node. Once these are confirmed, it is possible to proceed upward to reach the top. The Merkle tree and Merkle root mechanism can reduce the levels of hashing to be performed since they promote faster verification and transactions. Figure 1 depicts the branching in a Merkle tree.

2.2.3 Smart contract

Smart contracts are a form of agreement between two parties. It exists in the form of computer code and runs on the blockchain. They are immune to modifications and are stored on a public database. The idea behind the functioning of Smart contracts is that they are automatically executed as soon as the conditions related to the agreement are met. This eradicates the need for a third party. Because of blockchain technology, smart contracts can be decentralized so that they are fair and trustless. Blockchain is a shared database that is

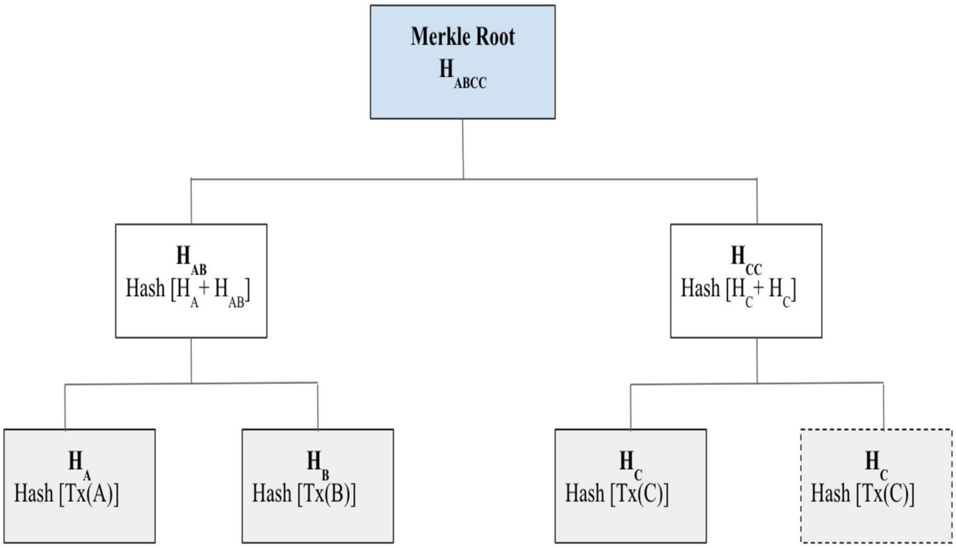


Fig. 1 Merkle Tree

run by many computers. These computers belong to different people; therefore, neither a single person nor a single company can control the blockchain. Consequently, it is nearly impossible to hack it. To hack a blockchain or a smart contract, the adversary must hack or gain control over more than half of the nodes. This makes it safe for smart contracts to operate since no one can change it. Smart Contracts are also capable of removing administrative overhead and are among the most attractive features associated with blockchain technology. Smart Contracts and Blockchain technology go hand in hand because blockchain can act as a database and confirm the transactions that took place, and smart contracts can execute predetermined conditions. Smart contracts act like computers running on simple if-then rules or conditional programming. Figure 2 depicts the basic working of a Smart Contract.

2.3 Internet of Things (IoT) policies

The future of IoT depends on digital policies. The different kinds of devices added to the IoT network may be in consumer products like wearables, smart TVs, smart light bulbs, etc. They can also be industrial or commercial devices like electric meters, manufacturing equipment, logistics systems, etc. Since everything will be connected, it is necessary to build the IoT on a foundation of strong digital policies. Consumer products may deal with Security, Software Updates, Feature Override, User Controls, Support Considerations, and Device Malfunctions. Similarly, Digital Policies Considerations for Industrial/Commercial Devices may be in the form of Uptime Rates, Maintenance, Scheduling Data Regulations and Restrictions, and Tamper-Proofing. To ensure that all these aspects are taken into account, it is necessary to treat IoT devices as digital properties by developing comprehensive IoT policies and ensuring that all IoT devices adhere to the IoT policies. A digital policy program simply adds another layer. If there are no digital policies, an organization is at higher risk. Figure 3 is an overall depiction of the same.

3 Proposed methodology

The proposed system architecture is classified into three layers: (1) IoT nodes, (2) Blockchain network 3) Application layer (Fig. 4)

IoT nodes Sensor nodes in the IoT network are the intelligent systems that are used to process and gather sensor information, and exchange data between the neighboring nodes or servers on the network. The controller's selection is an essential part of any IoT ecosystem node because of its data processing and data connectivity. A limited number of processors support Linux operating systems (LOS), and raspberry pi is one of these processors. Raspberry pi [47] is a low cost, credit card-sized computer that runs LOS and provides many GPIO pins that allow us to interact with the electronics interfacing and Internet of Things (IoT). Table 2 represents the raspberry pi models' technical specifications. Blockchain Network (especially Ethereum (explained in next part)) provides two different ways to connect with the decentralized application, and IoT devices are:

Geth Install Ethereum platform in every node and then mined to each other for developing blockchain network.

web3 This library helps to set up bridging between the devices and the blockchain network.

Installing the geth and step up Ethereum in every node and mine it with each other is quite complicated. However, web3 is much simpler than geth. It only needs to install the web3 library in the IoT nodes. The web3 library requires operating systems such as windows, mac, or Linux for the

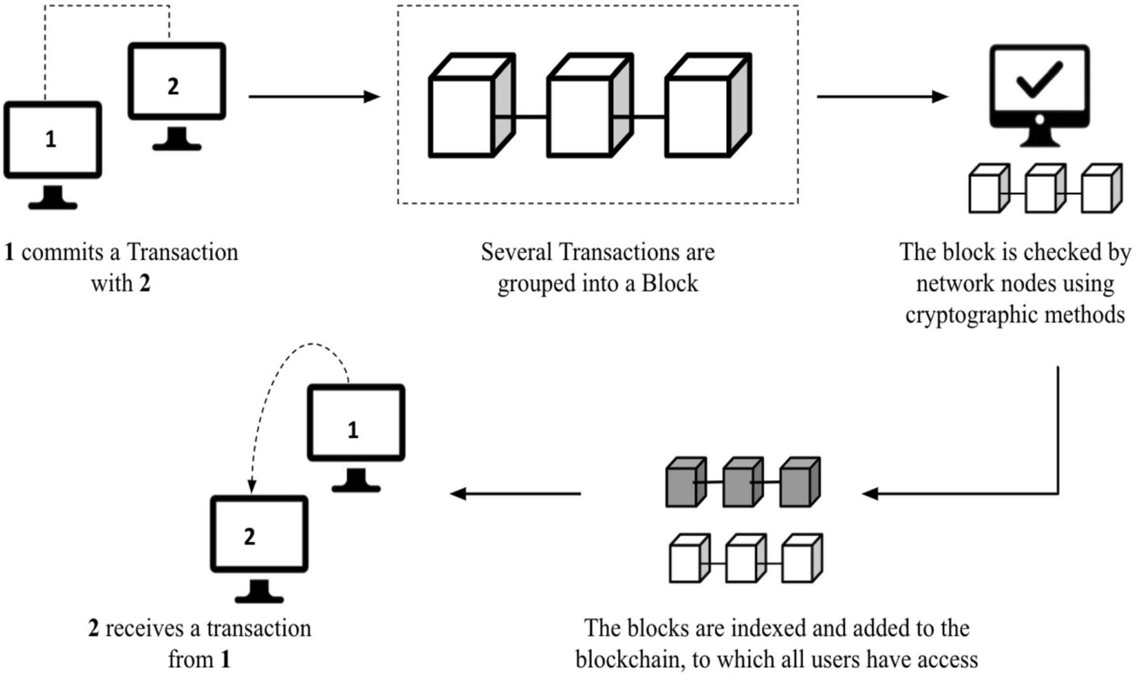

1
2
1 commits a Transaction with 2
Several Transactions are grouped into a Block
The block is checked by network nodes using cryptographic methods
1
2
2 receives a transaction from 1
The blocks are indexed and added to the blockchain, to which all users have access

Fig. 2 Working of smart contracts

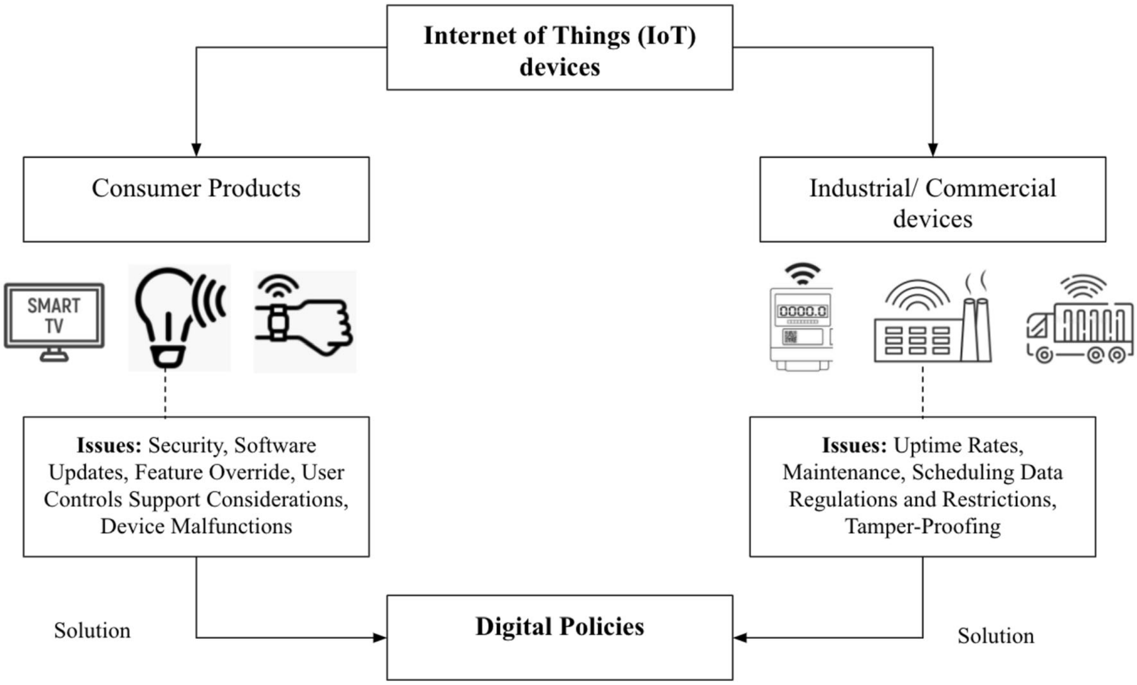

Internet of Things (IoT) devices
Consumer Products
Industrial/ Commercial devices
SMART TV
0000.0
Issues: Security, Software Updates, Feature Override, User Controls Support Considerations, Device Malfunctions
Issues: Uptime Rates, Maintenance, Scheduling Data Regulations and Restrictions, Tamper-Proofing
Solution
Digital Policies
Solution

Fig. 3 Internet of Things and digital policies

installation. Raspberry pi supports LOS that helps to install the web3 library and is also suitable for the proposed approach. DHT11 sensor is a low-cost temperature and humidity sensor integrated with the IoT node to communicate data between the blockchain networks. This sensor is fused with the dedicated negative temperature coefficient to measure temperature and send the temperature and humidity values to the raspberry pi via serial communication. A reliable temperature coefficient implies that the amount of resistance decreases with the increase in the value. The measuring ranges for temperature and humidity are 0 °C to 50 °C and 20% to 90%.

Blockchain network Usually, IoT devices are connected through the central server, especially in cloud architectures.

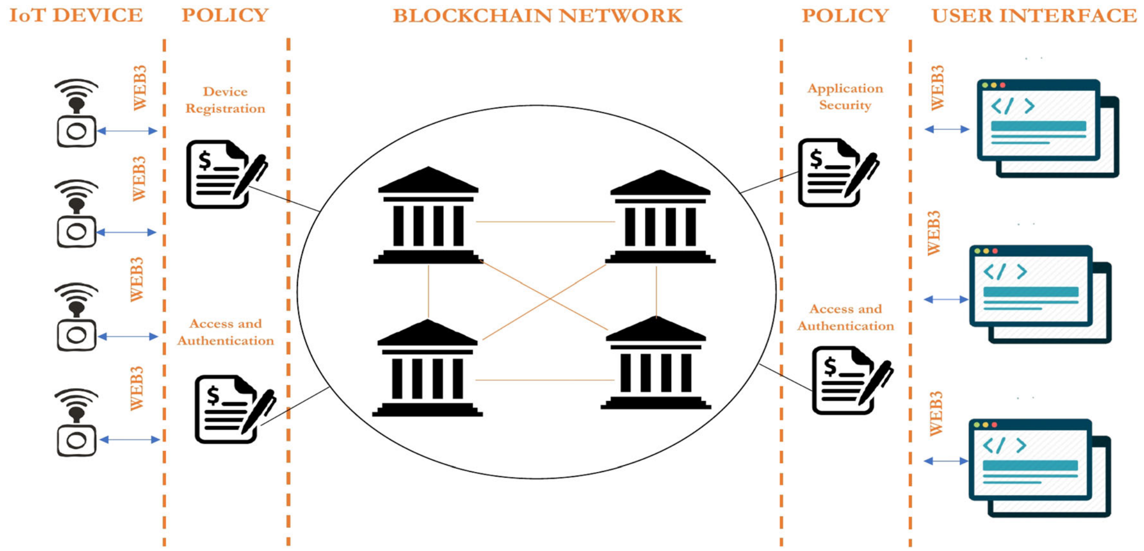


Fig. 4 Architecture of the proposed system

Table 2 Raspberry Pi technical specification

S.no.	Raspberry Pi model	Releasing year	ARM core and inbuilt RAM
1.	Raspberry Pi 4	2019	64 Bit; 2, 4 or 8 GB
2	Raspberry Pi 3A+	2019	64 Bit; 512 MB
3	Raspberry Pi 3B+	2018	64 Bit; 1 GB
4	Raspberry Pi Zero	2017	512 MB
5	Raspberry Pi 2 Model B	2016	32 Bit; 1 GB
6	Raspberry Pi 1 Model A+	2014	32 Bit; 512 MB
7	Raspberry Pi 1 Model B+	2014	32 Bit; 512 MB

Cloud server helps to process, send, and receive data via IoT devices, although cloud servers are vulnerable to numerous security attacks. Every portion of the IoT cloud architecture acts as a single point of failure [46]. To overcome these issues, Blockchain technology is one of the solutions for the IoT network. Blockchain technology is also beneficial since it leads to (1) reduction in cost due to non-interference of the third party, (2) single-point failure due to the decentralized ledger, (3) higher resistance to security attacks, (4) increased trust in IoT networks due to Cryptographic protocol. As of now, Blockchain technology has two major platforms, i.e., Bitcoin [48] and Ethereum [49]. In Bitcoin, there is limited efficiency due to cryptocurrencies. Still, Ethereum overcame the cryptocurrency limits and extended blockchain technology usage to real-world applications using smart contracts. In our proposed work, the Ethereum platform is deployed. Smart contracts work as a gateway between the blockchain network and IoT devices. Three different contracts, such as registered devices, data access, and checking the application vulnerability, are deployed in the blockchain network. Further explanations are discussed in later sections.

Application end layer A Decentralized application or Dapp or application layer is a service that allows straightforward interaction between the end-users and providers (e.g., peer to peer interaction between the owner and buyers) [50]. Dapps provide the interface for the end-user through the usage of JavaScript API. This API is used to create a pipeline between the blockchain network and web application. Dapp browsers can inject the Ethereum web3 API into JavaScript enabled web applications that help generate connectivity or work as a gateway between the Dapp application and blockchain network. Table 3 represents the list of Dapp browsers with platforms.

Cross-site scripting (XSS) allows hackers to inject malicious code into the web application for the motive of stealing user applications (especially login details). This is the most common web attack. Some of the application frameworks already provide some solutions, such as sanitization. Sanitization is the inspection of the untrusted value and converted into a particular value that can be safely inserted into the Document object model. In most cases, sanitization does not modify the value. Our proposed work also created a smart contract to avoid or block this attack and provide a secure IoT solution for the end-user.

In this section, we present the different policies for the IoT system, which are as follows:

Table 3 List of dapp browsers

Dapp Browser	Platform Supported
Mist	Windows, Mac, and Linux
Parity	Windows, Mac, and Linux
MetaMask	Browser supported: Window, Linux and Mac
Toshi	iOS and Android
Cipher	iOS and Android
Trust Browser	iOS and Android

3.1 Hardware and device security policy

In the proposed study, Merkle trees are considered for the device registration. The main advantage of the merkle tree's is to register the device without revealing its private parameters such as metadata (M_d), public key (P_b) or private key (P_r). Metadata contains device information such as device name, manufacturer, MAC address. In [51], researchers have generated ethereum wallet or elliptic curve key pairs at the initial step and considered Ganache to establish dummy public blockchain network that follows in the proposed study. Ganache provides ten ethereum accounts with 100 ether in each account for testing and development of the network. Metamask used an ethereum wallet. To create the metadata hash, three parameters are provided to the network: (1) Device Name, (2) Manufacturer name, (3) MAC address.

H(D_a) and H(M_a)are the Device name and Manufacturer name, and H (MAC_{add}) is the MAC address of the IoT device and is presented in the form of hash values. Hash values of H(D_a) and H(M_a) are combined to form a combined hash H(D_a) + M_a), and the Hash of the H(MAC_{add}) remain the same. Merkle tree usually mines even number of transaction but in our case there are odd number of transaction. Ethereum uses modified merkle patricia tries for transaction mine so it duplicated H (MAC_{add}) to complete the even transactions. The final stage also called Merkle root, combines the value of every hash value in the system. Hash value of the system or Merkle root is H(H(D_a) + H(M_a) + H (MAC_{add})) (Fig. 5). After completing the Merkle root, firmware hash is generated, and devices are successfully registered in the blockchain network. Based on P_b and P_r key pair, M_d hash and firmware(*firm*) hash, a unique device ID(D_{id}) is allotted to the IoT device. This D_{id} will not match with another IoT device. Elliptic Curve Digital Signature Algorithm (ECDSA) used to generate the P_b and P_r key pair through the Elliptic curve cryptography (ECC). The generated key pair helps to authenticate and validate the message. The following steps are discussed about IoT policy for registered device are showed in Algorithm 1:

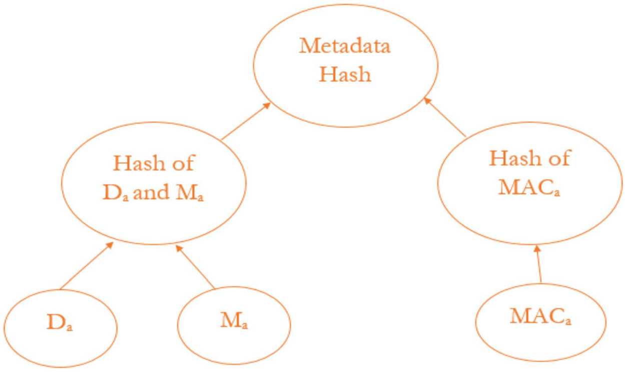


Fig. 5 Merkle root for device registration

```
Algorithm 1: IoT Policy for the Hardware
Result: Device Registration Status
Parameters: $D_{id}$, firm Hash, $MAC_{add}$, ;
while begin do
    Verification step by step;
    if $DP_{con}$ and $ABI == False$ then
        Error;
    else
        if $P_b$ and $P_r == False$ then
            Error;
        else
            Device Allow to Verify the
             parameters;
            if $MAC_{add} == True$ then
                Error;
            else
                Move to another parameter;
                if firm Hash == True then
                    Error;
                else
                    Move to another parameter;
                    if $D_{id} == True$ then
                        Error;
                    else
                        $D_{reg}$;
                    end
                end
            end
        end
    end
end
```

The IoT policies for the device registration are discussed below:

Step 1 In the initial step, Deployed contract address DP_{con} and ABI are being verified. If these parameters are not matched, the system will not allow or refuse the IoT device to enter the network.

Step 2 P_b and P_r key are required to verify that it was generated or not. Suppose the key pair is not generated. The smart contract will emit an error.

Step 3 Every IoT device MAC_{add} is different. If there is a matching at the time of device registration, the smart contract will emit an error. .

Step 4 After the successful completion of the Merkle root, *firm* hash is generated and should always be different. If there is possible matching due to some system fault, smart contract will emit an error.

Step 5 The last step is to verify the D_{id} with existing D_{id}'s. If there are some chances of matching, the smart contract emits the error.

3.2 Access and authentication policy

For data access and device authentication, IoT devices should be connected to the same blockchain network and verify initial parameters such as DP_{con}, ABI, and P_b key. This verification is required in the IoT device authentication and UI (User Interface authentication). The message is hashed via SHA-2 algorithm and generates a digital signature through ECDSA algorithm with P_r. The message is encrypted with a digital signature and D_{id} in the last stage. At the receiver end, the elliptic curve (EC) helps recover the P_b and matched D_{id} to allow the IoT device's data access. Algorithm 2 shows the complete process of access and authentication of the IoT devices in the blockchain network.

```
Algorithm 2: Access and Authenticate Policy
Result: Data Authentication or Not
Parameters: $D_{id}$, $P_b$, $P_r$ , $DP_{con}$, ABI ;
while begin do
    Data Access Authentication for IoT Device;
    if $DP_{con}$ and $ABI == False$ then
        Error;
    else
        Device Allow to Verify the parameters;
        if $P_b == False$ then
            Error;
        else
            Move to another parameter;
            if $D_{id} == False$ then
                Error;
            else
                Move to another parameter;
                check Front end application;
                if $DP_{con}$, ABI, $P_b == False$
                 then
                    Error;
                else
                    $Ack_{yes}$ == True;
                    Data Granted to access;
                end
            end
        end
    end
    Policy Change or Policy Updation;
    if $DP_{con}$, ABI, $P_b == False$ then
        Error;
    else
        Policies allow to change;
    end
end
```

The IoT policy for access and authentication are discussed below:

Step 1 The initial step for allowing access to data in the blockchain network is to check the DP_{con}, ABI, P_b. If these parameters are correct, blockchain allows verifying other parameters. Otherwise, the connection will be refused.

Step 2 P_b will match for one more time after the initial process is completed. If Key is not matched, then it denies the connection.

Step 3 The Smart contract checks D_{id}. If it is registered in the system, the network allows access; otherwise, it refuses.

Step 4 If there is successful acknowledgment from the blockchain network with "yes," it means devices and Front end application can access IoT data.

Step 5 To change or update the IoT policy, DP_{con}, ABI, P_b, D_{id} should be matched to the existing contract.

3.3 Application security policy

Front end applications are highly vulnerable that allow hackers to add some malicious code to steal user information. In the proposed work, we have deployed a smart contract to check or monitor any vulnerability in the front end application. Some applications already provide security layers such as angular-CLI. The proposed policy gives an extra layer of security to develop a more secure system. Algorithm 3 represents the security policies for the front end layer. Steps of the deployed security deployed approach as follows:

```
Algorithm 3: Policy for Application Security
Result: Prevent to Inject any malicious code
        and by-pass security
Parameters: $D_{id}$, $P_b$, $P_r$ , $DP_{con}$, ABI ;
while begin do
    Verification step by step;
    if $DP_{con}$ and $ABI == False$ then
        Error;
    else
        UI Allow to Verify the parameters;
        Checking the conditions for XSS via
         sanitizing;
        if $Bind_{CSS} == False$ then
            Vulnerability Error;
        else
            Move to another parameter;
            if $Url's\ EXE_{code} == False$ then
                Vulnerability Error;
            else
                Move to another parameter;
                if $Int_{csp} == False$ then
                    Vulnerability Error;
                else
                    Authenticate;
                end
            end
        end
    end
end
```

The security policy for the front end layer are discussed below:

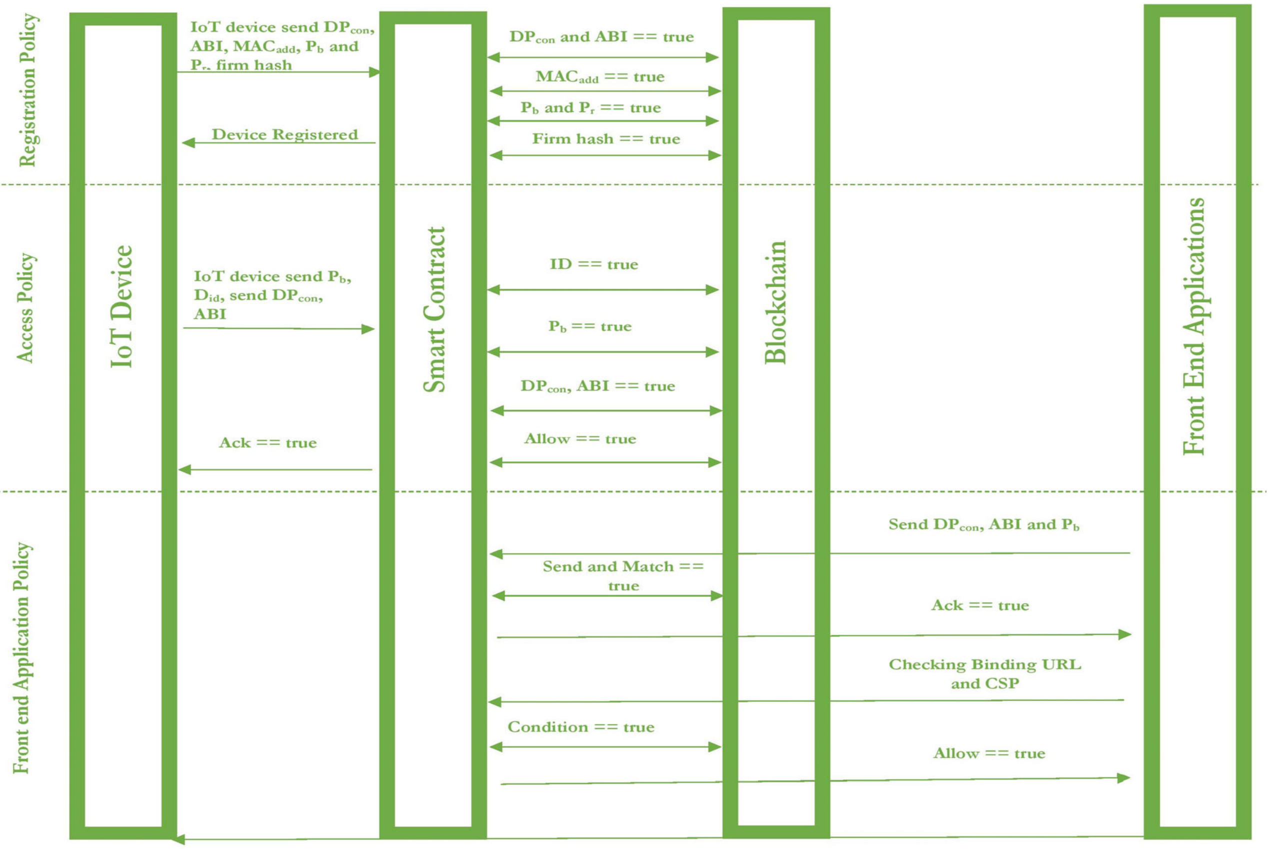


Fig. 6 Workflow of the proposed work

Step 1 In the Front end application, the first thing needed to verify the DP_{con}, ABI. If it is matched, UI(user interface) allows validating other parameters; else, an error is generated.

Step 2 Contract is checking the conditions for XSS (Cross-site scripting) via sanitizing.

Step 3 Contracts check the CSS (Cascading Style Sheets) property binding to the front end application, then moving to another step; otherwise, the vulnerability error is generated.

*Step 4*Check URLs added to the code are executed as code; otherwise, raise a vulnerability error.

Step 5 Check content security policy is integrated into the front end application; otherwise, raise the vulnerability error.

Figure 6 represents the workflow of IoT policies. It represents the fusion of the hardware policy, data access policies, and application security policy.

The proposed workflow of the system is categorized into four blocks and three layers. The block section consists of IoT devices, smart contracts, blockchain, and front end applications. The layers consist of registry policy, access policy, and front end application policy. UI sends a request to the blockchain network with DP_{con}, ABI, and P_b key. If these parameters match, UI can access the blockchain network to register the IoT device. IoT devices send request DP_{con}, ABI to enter into the blockchain network and start matching its parameters(D_{id}, *firm* Hash, MAC_{add},P_b and P_r key pair) for the unique D_{id}. In the data access and authentication policy, IoT devices send the request to a smart contract, which includes DP_{con}, ABI, D_{id} and P_b key. If the conditions are true, a blockchain network will allow the data access; otherwise, it will refuse the connection. The last layer is the front end application policy, one of the most critical layers in any application because front end applications are always highly vulnerable and mostly attached by hackers. The front end application sends a request to the smart contract about DP_{con}, ABI, and P_b key. Smart contracts allow the request if the parameters are matched. Smart contracts also send an acknowledgment (Ack) to the front end application. Smart contracts will check parameters of the front end application parameters such as URL's binding, content security policy. If these parameters match, smart contracts will allow the front end application to access the blockchain network and generate errors with the tag "High Vulnerability: Connection not Safe."

4 Results and discussion

The following subsections incorporate the results and evaluation methodologies for the proposed work.

4.1 Evaluation testbed

Four end nodes, such as three raspberry pi nodes and one computer node, are used to evaluate the proposed approach. The computer node worked as an RPC server and raspberry pi nodes as a client node. Moreover, three different kinds of raspberry pi are used, such as raspberry pi zero, raspberry pi model 3B, and raspberry pi model 3B+. Table 4 represents the internal configuration of the raspberry pi models.

Smart contracts are the way to deploy our proposed approach in the blockchain network, and ethereum is considered a blockchain platform. Solidity [52] is a high-level language that successfully codes the smart contract into the blockchain network. Angular-CLI employed an interaction bridge between the end nodes and smart contracts. Moreover, Ganache is also a part of the proposed work that provides blockchain development tools for testing and development purposes. The proposed approach using the Ganache is similar to the public blockchain network. Different kinds of IoT policies are deployed in our proposed policy framework, such as device registration policy, access and authentication of data policy, and application layer policy. The proposed system experimentation is performed as follow:

1. To evaluate the metadata of the registered device.
2. To evaluate the power consumption of the IoT device at the time of initial request generating.
3. To evaluate the bandwidth consumption during the uploading and downloading of IoT data.
4. To calculate the latency and throughput of the blockchain network.
5. To evaluate the front end vulnerability and generate an alert at the time of vulnerability detected.

4.1.1 Evaluation of registered device

Metadata of IoT devices are used to evaluate the registered IoT device on the blockchain network. Metadata consists of the device name, mac address, and manufacturer name. Public or Private keys are generated at the time of registering devices to encrypt and decrypt IoT device requests and messages. However, networks directly reject the device if the P_b does not match. In the proposed system, Merkle tree is employed to register the IoT device without revealing its metadata information. The main components of the Merkle tree are calculateMerkleroot(), validateMerkleroot(), fetchinglatestblock(), and hashPair(). Figure 8 represents the Merkle tree for the IoT devices. Merkle root is the combination of hashes of all block transactions in the blockchain network.

7075152d03a5cd92104887b476862778ec0c87be5c2fa1 c0a90f87c49fad6eff is a Merkle root that consist of another hashes (Fig. 7 Also, MerkleTree generates proof as well as verifies the proof. Figure 8 shows the testing and verifying Merkle leaves for the IoT network.

Smart contracts consist of the events used to create a device via metadata hash and device id. Moreover, it can also update the device property. Figure 9 shows the final JSON file after the registration of the IoT device on the blockchain network. Some components are included in the JSON file, such as an identifier, metadata hash, metadata, address, P_b, P_r, curve, and device id. Identifier and address are the account address of the IoT device on the ethereum network. Metadata Hash is a Merkle root created from the metadata, and metadata consists of information of device name, device manufacturer name, and mac address of the device. Public and Private keys are the key generated at the time of nodes registered in the ethereum. If any of the information(except the device name and manufacturer name) is similar to the old registered device, a new id will not be allocated. It only allots if all the above information is unique.

4.1.2 Access and authentication

To evaluate the IoT network's access and authentication, the first step required is to assess the cyber-attacks and

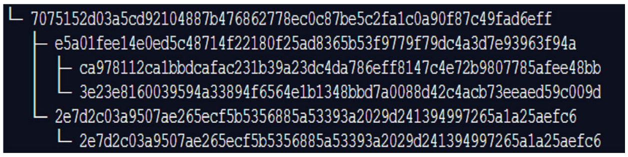
Table 4 Internal configuration of different raspberry pi models

Models	RAM	Booting time class 10 SD card Raspberian Buster (s)	Core (bit)	Clock frequency (GHz)
Raspberry Pi Zero	512 MB	50-80	32	1.0
Raspberry Pi Model 3B	1 GB	40-50	64	1.2
Raspberry Pi Model 3B+	1 GB	30-40	64	1.4

security requirements. Identification, single authentication or mutual authentication and spoof of attack, Sybil attack, and denial of service attacks are the security requirement and cyber-attacks. Authentication and mutual authentication [52, 52–54] play a significant role in the system security requirement. Smart contracts evaluate the device's pre-existence in the authentication system through metadata, address, and private key. If there is no pre-existence and device registered to the blockchain network, it will allow the system's IoT device transaction. For the mutual authentication system, every node or device participating in the blockchain network should have an authentication pass that elaborates trust between each node. However, fake authentication is quite tricky because the authentication pass is based on the private key, and every device or node has its private key. Moreover, these private keys are unique for every device or node. A spoofing attack is a situation when a computer application or hacker successfully identifies device information to gain unauthorized benefits. Hackers in the spoofing attacks need some information such as device id, metadata, and private key. However, hackers can get the device id and metadata from any source, but a private key is impossible to extract. Same in the case of the Sybil attack, hackers generate fake information in the system. The registered device will have a unique ID and P_b. The chances to replicate a P_b are less than equal to zero because it is constructed from the P_r key. DDoS attacks hacked connected devices such as web cameras, routers, access points, and autonomous systems due to the connectivity's centralization. Blockchain technology provides peer to peer connectivity without including any third party, i.e., cloud computing. All transactions ensured through the blockchain technology are in the form of cryptographic hashes that provide proof of work (PoW).

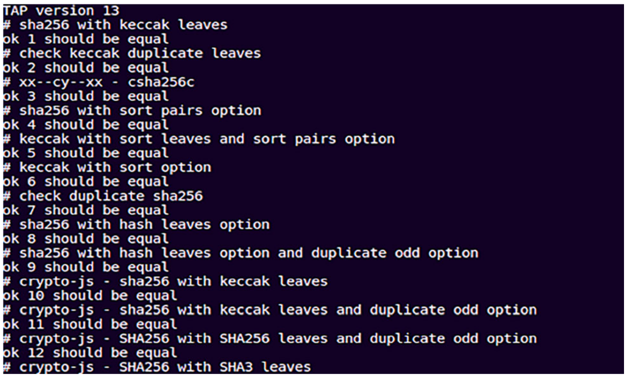
4.2 Evaluation

The primary focus of this study is to register an IoT device and provide a secure mechanism for transmitting the data between the blockchain networks. Throughput and latency are the two main parameters that help to evaluate the blockchain network. Transaction throughput is defined as the rate of the valid transactions committed by the blockchain network. The rate of a transaction is referred to as transaction per second(TPS) (see in Eq. 1)



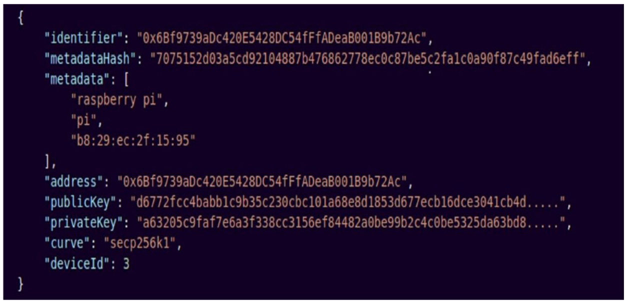
```
└ 7075152d03a5cd92104887b476862778ec0c87be5c2fa1c0a90f87c49fad6eff
   ├ e5a01fee14e0ed5c48714f22180f25ad8365b53f9779f79dc4a3d7e93963f94a
   │  ├ ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
   │  └ 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d
   └ 2e7d2c03a9507ae265ecf5b5356885a53393a2029d241394997265a1a25aefc6
      └ 2e7d2c03a9507ae265ecf5b5356885a53393a2029d241394997265a1a25aefc6
```

Fig. 7 MerkleTree for the IoT device



```
TAP version 13
# sha256 with keccak leaves
ok 1 should be equal
# check keccak duplicate leaves
ok 2 should be equal
# xx--cy--xx - csha256c
ok 3 should be equal
# sha256 with sort pairs option
ok 4 should be equal
# keccak with sort leaves and sort pairs option
ok 5 should be equal
# keccak with sort option
ok 6 should be equal
# check duplicate sha256
ok 7 should be equal
# sha256 with hash leaves option
ok 8 should be equal
# sha256 with hash leaves option and duplicate odd option
ok 9 should be equal
# crypto-js - sha256 with keccak leaves
ok 10 should be equal
# crypto-js - sha256 with keccak leaves and duplicate odd option
ok 11 should be equal
# crypto-js - SHA256 with SHA256 leaves and duplicate odd option
ok 12 should be equal
# crypto-js - SHA256 with SHA3 leaves
```

Fig. 8 Testing Merkle leaves for the IoT device



```
{
    "identifier": "0x6Bf9739aDc420E5428DC54fFfADeaB001B9b72Ac",
    "metadataHash": "7075152d03a5cd92104887b476862778ec0c87be5c2fa1c0a90f87c49fad6eff",
    "metadata": [
        "raspberry pi",
        "pi",
        "b8:29:ec:2f:15:95"
    ],
    "address": "0x6Bf9739aDc420E5428DC54fFfADeaB001B9b72Ac",
    "publicKey": "d6772fcc4babb1c9b35c230cbc101a68e8d1853d677ecb16dce3041cb4d.....",
    "privateKey": "a63205c9faf7e6a3f338cc3156ef84482a0be99b2c4c0be5325da63bd8.....",
    "curve": "secp256k1",
    "deviceId": 3
}
```

Fig. 9 JSON file after registering the IoT device

$$Throughput = \frac{Valid\_Transaction}{s} \tag{1}$$

Latency in the blockchain network term is the time between the first confirmed transaction and the registering transaction in the blockchain network. In every transaction, latency is the difference between the transaction completion time (v2) and transaction deployed time (v1) (see in Eq. 2)

$$Latency = v2 - v1 \tag{2}$$

The scalability of the blockchain network is directly dependent on the throughput and latency. Any kind of change in the parameters such as network size or hardware configuration will impact them through the blockchain network's latency. In the proposed work, we have connected three raspberry nodes clients such as raspberry pi model 3B, raspberry pi model 3B+, and raspberry pi zero and one laptop as a blockchain network RPC (Remote Procedure Call) server [55]. In [42, 56], the authors proposed an access and authentication system that runs on the QT Framework and tested every node including the ganache server node. Still, in our proposed work, one computer works as a RPC server, and other clients are connected through the Web3 library. Table 5 represents the system configuration comparison of proposed systems with other related systems.

Table 5 Comparison of proposed systems configuration with other systems

Systems	Main node	Connected nodes	Pathway
Proposed work	ASUS Laptop	3 Raspberry Pi of different models	WEB3.JS RPC Server
Lightweight authentication system [42]	Dell Vostro Laptop	Raspberry Pi and HP ProBook	QT Framework C++
Bubble of Trust [56]	HP Laptop	Raspberry Pi and HP Laptop	QT Framework C++

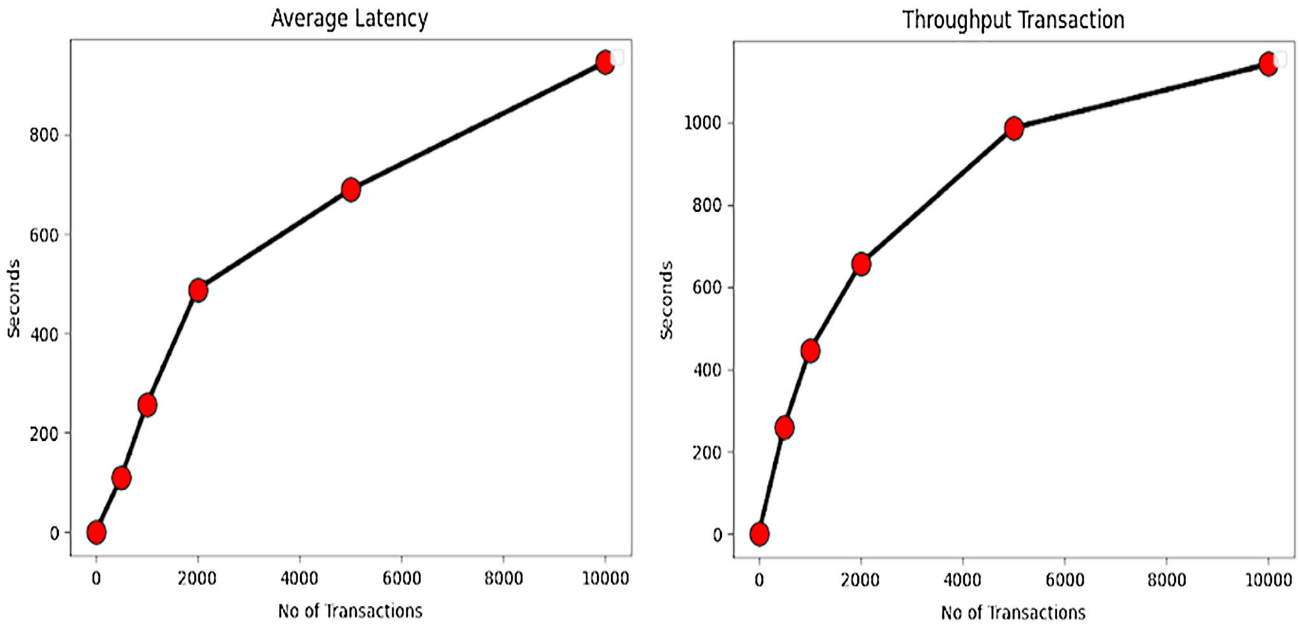
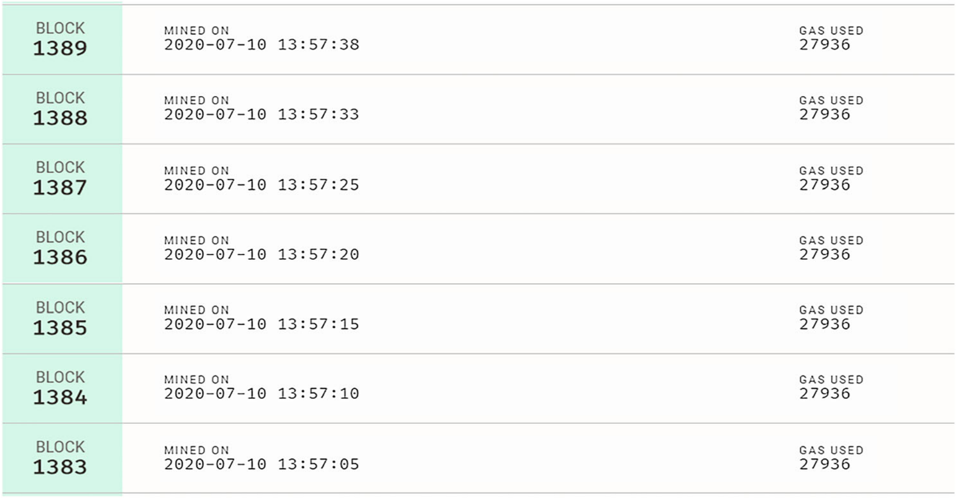


Fig. 10 Calculated throughput and latency of the proposed blockchain network system

Ganache has been used as a main node in our system to mine the transaction received from the client nodes. The client nodes are independent from the geth synchronization and only need enough computation power to process send and receive access requests. Till now, proposed solutions results [55, 56] regarding the access and authentication system were limited to time consumption and CPU usage. We have calculated the throughput and latency of the IoT-Blockchain network and performed five experiments to calculate the network's throughput and latency. The number of the transactions per second and network latency is calculated at 500, 1000, 2000, 5000, and 10,000. Throughput transactions of the IoT data on the blockchain network at 500, 1000, 2000, 5000 and 10,000 transactions are 1.92, 2.24, 3.04, 5.07 and 8.77 respectively (Fig. 10). The network's latency at 500, 1000, 2000, 5000 and 10,000 transactions are 4.54, 3.90, 4.08, 7.25, and 10.582, respectively (Fig. 10). Figure 11 represents the confirmation of blocks mined with time at the IoT-Blockchain network.

4.2.1 Energy efficiency of IoT devices

Power consumption, aka energy efficiency of the raspberry pi (IoT device), is considered at the time of initial request generated from the IoT node. The proposed system [42, 56] devices consumes more power as compared to our proposed method. In [56], source code and smart contracts are available on the GitHub repository, but wasn't [42] available. So we designed a smart contract according to their proposed algorithms and tested on our proposed setup. In addition, their smart contracts are valid to send the string message but our proposed study include the real-time sensor DHT11. We modified their smart contract to send the sensor data. Table 6 shows a comparison of the proposed systems' power consumption with other systems [42, 56].To measure the power consumption of IoT device, we used external power meter. The calculation of the power consumption for IoT device is the difference between the power consumed by the device in data transmitted state and power consumed by the device in the ideal state. As per Fig. 12, the raspberry pi 3B+ IoT node consumed less power than other nodes (raspberry pi 3B and

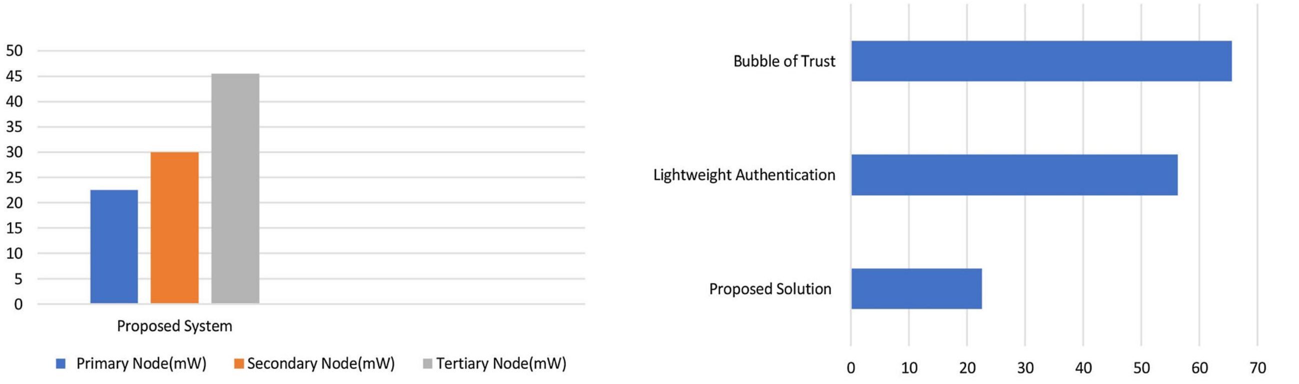


BLOCK 1389	MINED ON 2020-07-10 13:57:38	GAS USED 27936
BLOCK 1388	MINED ON 2020-07-10 13:57:33	GAS USED 27936
BLOCK 1387	MINED ON 2020-07-10 13:57:25	GAS USED 27936
BLOCK 1386	MINED ON 2020-07-10 13:57:20	GAS USED 27936
BLOCK 1385	MINED ON 2020-07-10 13:57:15	GAS USED 27936
BLOCK 1384	MINED ON 2020-07-10 13:57:10	GAS USED 27936
BLOCK 1383	MINED ON 2020-07-10 13:57:05	GAS USED 27936

Fig. 11 Successful mined of blocks with timestamp

Table 6 Comparison of power consumption of IoT device in proposed system with other systems

Systems	Nodes	Power consumption
Proposed system	Raspberry Pi as IoT nodes and Laptop as RPC	Raspberry Pi 3B + (Primary node): 22.54 mW Raspberry Pi 3B (secondary node): 30 mW Raspberry Pi Zero (tertiary node): 45.54 mW
Lightweight authentication system [42]	Laptop and Raspberry Pi as IoT nodes	Raspberry Pi as IoT node: 56.24 mW
Bubble of trust [56]	Laptop and Raspberry Pi as IoT nodes	Raspberry Pi as IoT node: 65.54 mW



a Power consumption of proposed system IoT nodes

b Comparison of IoT node power consumption with the existing solutions

Fig. 12 Comparison of power consumption: **a** proposed system IoT nodes and **b** with existing solutions

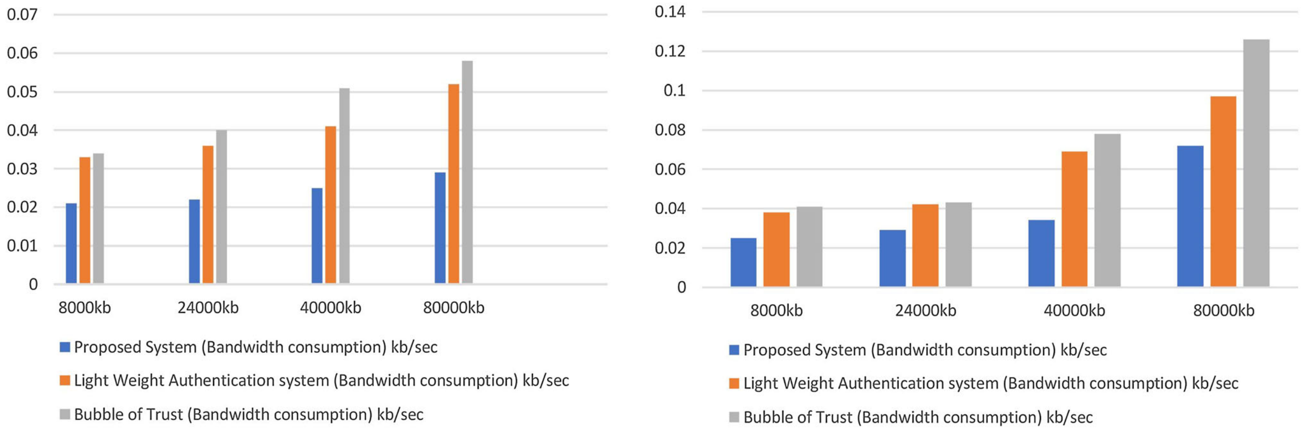
raspberry pi zero). Our proposed IoT devices' power consumption consumes less power or more energy efficiency than the existing solution.

4.2.2 Bandwidth consumption

We evaluate and measure the proposed system's bandwidth usage via upload and download data to and from the blockchain server. We considered four different file sizes, such as 200 kb, 600 kb, 1000 kb, and 2000 kb, with an

Table 7 Comparison of bandwidth consumption

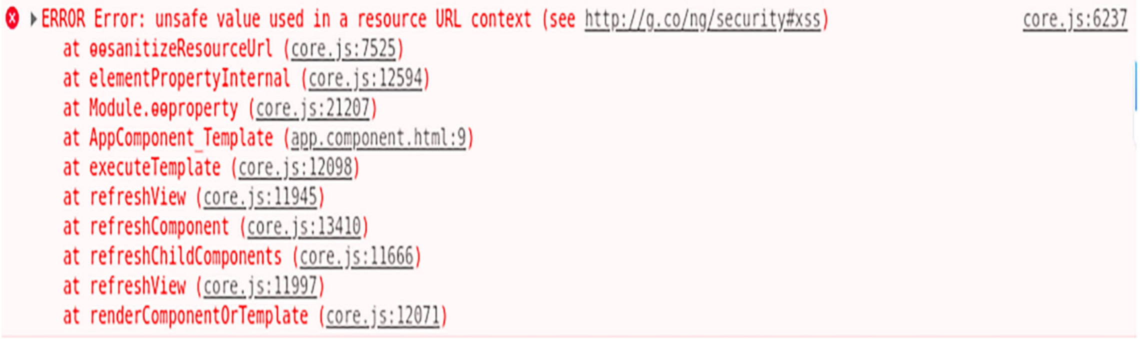
Data size (kb)	Transactions	Proposed system (bandwidth consumption) (kb/s)	Lightweight authentication system (bandwidth consumption) (kb/s)	Bubble of trust (bandwidth consumption) (kb/s)
200	40	Upload: 0.021 Download: 0.025	Upload: 0.033 Download: 0.038	Upload: 0.034 Download: 0.041
600	40	Upload: 0.022 Download: 0.029	Upload: 0.036 Download: 0.042	Upload: 0.040 Download: 0.043
1000	40	Upload: 0.025 Download: 0.034	Upload: 0.041 Download: 0.069	Upload: 0.051 Download: 0.078
2000	40	Upload: 0.029 Download: 0.072	Upload: 0.052 Download: 0.097	Upload: 0.058 Download: 0.126



a Bandwidth consumption at the time of data upload

b Bandwidth consumption at the time of data download

Fig. 13 Comparison of bandwidth consumption with existing solutions: **a** data upload **b** data download



```
⊗ ▸ERROR Error: unsafe value used in a resource URL context (see http://g.co/ng/security#xss)        core.js:6237
      at ɵɵsanitizeResourceUrl (core.js:7525)
      at elementPropertyInternal (core.js:12594)
      at Module.ɵɵproperty (core.js:21207)
      at AppComponent_Template (app.component.html:9)
      at executeTemplate (core.js:12098)
      at refreshView (core.js:11945)
      at refreshComponent (core.js:13410)
      at refreshChildComponents (core.js:11666)
      at refreshView (core.js:11997)
      at renderComponentOrTemplate (core.js:12071)
```

Fig. 14 Alert generated at the time of front end policy violated

average of 40 transactions and a comparison with the existing works.

The bandwidth capacity is measured through the speedtest- cli [63], and packet size measure through the iPerf [64]. As per Table 7, the upload data consumes less bandwidth as compared to the download data. The proposed system consumes less bandwidth (80 Mb i.e. 2000 data size × 40 transactions) is 0.029 kb/s and 0.072 kb/s at the time of upload and download bandwidth consumption respectively as compared to other systems lightweight authentication (upload: 0.052 kb/s and download: 0.097 kb/s) and bubble of trust (upload: 0.058 kb/s and download: 0.126 kb/s) (see in Fig. 13).

Figure 14 represents that the front end application policy is violated due to unsafe URL context, and error is generated at the Google Chrome browser console.

4.3 Comparative analysis of the study

Table 8 presents the comparative analysis of our proposed work with works done in the past.

Table 8 Comparative analysis of the existing research works

Author and Year	Research work	Methodology/parameters	Results
Dorri et al., 2019 [57]	Security and privacy in IoT	Lightweight and scalable blockchain (LSB)	LSB decreases packet overhead, increases scalability
Shabandri and Maheshwari, 2019 [58]	Security and Privacy in IoT	Distributed Ledgers with IOTA and The Tangle	Scalability, fault tolerance, and quantum resistance
Watanabe and Fan, 2019 [59]	Security in IoT networks	Chip level blockchain security solution	Scalable, protected data transaction
Du et al., 2020 [60]	Security in IoT	Three-dimensional blockchain architecture (Spacechain)	Spacechain shows better scalability and better network throughput
Yazdinejad et al., 2020 [61]	Security in IoT networks	Software-defined networking controller architecture	Increased throughput and better overhead
Singh et al., 2020 [62]	Effective big data analysis	Blockchain-enabled Intelligent IoT Architecture with Artificial Intelligence (BlockIoTIntelligence)	BlockIoTIntelligence. shows better accuracy and latency
Our proposed work	IoT security and privacy	Smart contract based policies	The proposed approach shows satisfactory performance in terms of latency and throughput, shows better performance with respect to bandwidth and power consumption

5 Conclusion

As we know, technology these days is significantly dominated by IoT networks and devices. Owing to the security and policy issues in the IoT network environment, we suggested digital policies using Smart Contracts. Three different policies, i.e., Hardware and Device Security Policies, Access and Authentication policies, and Application security for the IoT network, have been proposed to increase IoT security. The overall architecture of the proposed system and the corresponding algorithms responsible for the working of smart contract-based policies are discussed in detail. Latency, Throughput, Bandwidth Consumption, and Energy efficiency have been taken into account for evaluating the system. Our results also show the comparison of devices power usage, which is satisfactory for the proposed approach. An alert system at the time of front end policy violation has also been appended to the designed system. Since security issues in IoT are many, we would like to develop some other digital policies related to cloud and device management in the future. Device management poses severe threats due to various issues associated with altering the device's function, controlling the device, and gathering data from the device. Specific policies targeting mobile systems and embedded devices may also be designed in the future. As the attack surface grows for IoT devices, it may also be interesting to tackle the security and privacy issues specific to industries like healthcare, finance, government, etc. Another interesting concept that might be worth exploring in the future could involve taking security and privacy to the next level involving accountability, transparency, and trustworthiness. Likewise, IoT platforms may be embedded with Blockchain technology and Artificial Intelligence techniques to build trust in IoT and AI-based platforms. Investigating the performance of different machine learning algorithms in such environments would be beneficial in building trustworthy systems. While we discussed IoT based policies in this study, security policies are yet to be explored. Due to the diversity in tools and methodologies involved in security operations, policies may be specific across devices and environments. Therefore, it might be interesting to study the relationship between IoT policies and security policies in the future.

References

1. Puri, V., Jha, S., Kumar, R., Priyadarshini, I., Abdel-Basset, M., Elhoseny, M., Long, H.V.: A hybrid artificial intelligence and internet of things model for the generation of renewable resource of energy. IEEE Access **7**, 111181–111191 (2019)

2. Puri, V., Priyadarshini, I., Kumar, R., Kim, L. C.: Blockchain meets IIoT: an architecture for privacy preservation and security in IIoT. In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–7. IEEE (2020, March)
3. Puri, V., Jagdev, S. S., Tromp, J. G., Van Le, C.: Smart bicycle: IoT-based transportation service. In: Intelligent Computing in Engineering, pp. 1037–1043. Springer, Singapore (2020)
4. Priyadarshini, I., Cotton, C.: Intelligence in cyberspace: the road to cyber singularity. J. Exp. Theor. Artif. Intell. (2020). https://doi.org/10.1080/0952813X.2020.1784296
5. Berdik, D., Otoum, S., Schmidt, N., Porter, D., Jararweh, Y.: A survey on blockchain for information systems management and security. Inf. Process. Manag. **58**(1), 102397 (2012)
6. Priyadarshini, I.: Cyber security risks in robotics. In: Cyber Security and Threats: Concepts. Methodologies, Tools, and Applications, pp. 1235–1250. IGI Global, Pennsylvania (2018)
7. Priyadarshini, I.: Introduction on Cybersecurity. Cyber Security in Parallel and Distributed Computing: Concepts, Techniques, Applications and Case Studies, pp. 1–37. Wiley, Hoboken (2018)
8. Tuan, T.A., Long, H.V., Kumar, R., Priyadarshini, I., Son, N.T.K.: Performance evaluation of Botnet DDoS attack detection using machine learning. Evol. Intell. **13**, 283–294 (2019)
9. Paul, N., Tesfay, W. B., Kipker, D. K., Stelter, M., Pape, S.: Assessing privacy policies of Internet of Things services. In: IFIP International Conference on ICT Systems Security and Privacy Protection, pp. 156–169. Springer, Cham (2018)
10. Bouachir, O., Aloqaily, M., Tesng, L., Boukerche, A.: Blockchain and Fog Computing for Cyber-Physical Systems: Case of Smart Industry. (2020) arXiv preprint arXiv:2005.12834
11. Chatfield, A.T., Reddick, C.G.: A framework for Internet of Things-enabled smart government: a case of IoT cybersecurity policies and use cases in US federal government. Gov. Inf. Q. **36**(2), 346–357 (2019)
12. Liu, H., Chen, G., Huang, Y.: Smart hardware hybrid secure searchable encryption in cloud with IoT privacy management for smart home system. Clust. Comput. **22**(1), 1125–1135 (2019)
13. Attkan, A., Ahlawat, P.: Lightweight two-factor authentication protocol and session key generation scheme for WSN in IoT deployment. In: Advances in Cybernetics. Cognition, and Machine Learning for Communication Technologies, pp. 189–198. Springer, Singapore (2020)
14. Aloqaily, M., Boukerche, A., Bouachir, O., Khalid, F., Jangsher, S.: An energy trade framework using smart contracts: overview and challenges. IEEE Netw. **34**(4), 119–125 (2020)
15. Vo, T., Sharma, R., Kumar, R., Son, L.H., Pham, B.T., Tien Bui, D., Le, T.: Crime rate detection using social media of different crime locations and Twitter part-of-speech tagger with Brown clustering. J. Intell. Fuzzy Syst. **38**(4), 4287–4299 (2020)
16. Priyadarshini, I.: Features and architecture of the modern cyber range: a qualitative analysis and survey. Doctoral dissertation, University of Delaware (2018)
17. Priyadarshini, I., Cotton, C.: Internet memes: a novel approach to distinguish humans and bots for authentication. In: Proceedings of the Future Technologies Conference, pp. 204–222. Springer, Cham (2019)
18. Priyadarshini, I., Wang, H., Cotton, C.: Some cyberpsychology techniques to distinguish humans and bots for authentication. In: Proceedings of the Future Technologies Conference, pp. 306–323. Springer, Cham (2019)
19. Hamza, R., Yan, Z., Muhammad, K., Bellavista, P., Titouna, F.: A privacy-preserving cryptosystem for IoT E-healthcare. Inf. Sci. (2019). https://doi.org/10.1016/j.ins.2019.01.070
20. Pasupuleti, S. K., Varma, D.: Lightweight ciphertext-policy attribute-based encryption scheme for data privacy and security in cloud-assisted IoT. In: Real-Time Data Analytics for Large Scale Sensor Data, pp. 97–114. Academic Press, New York (2020)
21. Priyadarshini, I.: Introduction to blockchain technology. In: Cyber Security in Parallel and Distributed Computing: Concepts, Techniques, Applications and Case Studies, pp. 91–107. Wiley, New York (2019)
22. Mohanta, B.K., Jena, D., Ramasubbareddy, S., Daneshmand, M., Gandomi, A.H.: Addressing security and privacy issues of IoT using blockchain technology. IEEE Internet of Things J. (2020). https://doi.org/10.1109/JIOT.2020.3008906
23. Qian, Y., Jiang, Y., Chen, J., Zhang, Y., Song, J., Zhou, M., Pustšek, M.: Towards decentralized IoT security enhancement: a blockchain approach. Comput. Electr. Eng. **72**, 266–273 (2018)
24. Singh, N., Kumar, T., Vardhan, M.: Blockchain-based e-cheque clearing framework with trust based consensus mechanism. Clust. Comput. (2020). https://doi.org/10.1007/s10586-020-03163-6
25. Banerjee, M., Lee, J., Choo, K.K.R.: A blockchain future for internet of things security: a position paper. Digit. Commun. Netw. **4**(3), 149–160 (2018)
26. Rosa, M., Barraca, J.P., Rocha, N.P.: Blockchain structures to guarantee logging integrity of a digital platform to support community-dwelling older adults. Clust. Comput. **23**, 1887–1898 (2020)
27. González, J.C., García-Díaz, V., Núñez-Valdez, E.R., et al.: Replacing email protocols with blockchain-based smart contracts. Clust. Comput. **23**, 1795–1801 (2020)
28. Hewa, T., Ylianttila, M., Liyanage, M.: Survey on blockchain based smart contracts: applications, opportunities and challenges. J. Netw. Comput. Appl. (2020). https://doi.org/10.1016/j.jnca.2020.102857
29. Barrera, D., Molloy, I., Huang, H.: Standardizing IoT network security policy enforcement. In: Workshop on Decentralized IoT Security and Standards (DISS), vol. 2018, p. 6 (2018)
30. Halepoto, I.A., Khan, U.A., Arain, A.A.: Retransmission policies for efficient communication in IoT applications. In: 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 197–202. IEEE (2018)
31. Atlam, H.F., Alassafi, M.O., Alenezi, A., Walters, R.J., Wills, G.B.: XACML for building access control policies in the Internet of Things. In: IoTBDS, pp. 253–260 (2018)
32. Ellis, W., Hersh, D.: XACML 3.0 Analysis, Purdue University (2015)
33. Mishra, N., Verma, L.P., Srivastava, P.K., Gupta, A.: An analysis of IoT congestion control policies. Procedia Comput. Sci. **132**, 444–450 (2018)
34. Le, F., Ortiz, J., Verma, D., Kandlur, D.: Policy-based identification of IoT devices' vendor and type by DNS traffic analysis. In: Policy-Based Autonomic Data Governance, pp. 180–201. Springer, Cham (2019)
35. Pal, S., Hitchens, M., Varadharajan, V., Rabehaja, T.: Policy-based access control for constrained healthcare resources in the context of the Internet of Things. J. Netw. Comput. Appl. **139**, 57–74 (2019)
36. Nobakht, M., Russell, C., Hu, W., Seneviratne, A.: IOT-NET-SEC: policy-based IoT network security using OpenFlow. In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 955–960. IEEE (2019)
37. Al-Shaboti, M., Chen, A., Welch, I.: Automatic Device Selection and Access Policy Generation based on user preference for IoT activity workflow. In 2019 18th IEEE International Conference On Trust, Security And Privacy in Computing and Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 769–774. IEEE (2019)

38. Ding, S., Cao, J., Li, C., Fan, K., Li, H.: A novel attribute-based access control scheme using blockchain for IoT. IEEE Access **7**, 38431–38441 (2019)
39. Lim, H.K., Kim, J.B., Heo, J.S., Han, Y.H.: Federated reinforcement learning for training control policies on multiple IoT devices. Sensors **20**(5), 1359 (2020)
40. Chen, Q., Srivastava, G., Parizi, R.M., Aloqaily, M., Al Ridhawi, I.: An incentive-aware blockchain-based solution for internet of fake media things. Inf. Process. Manag. **57**(6), 102370 (2020)
41. Al Ridhawi, I., Aloqaily, M., Boukerche, A., Jaraweh, Y.: A blockchain-based decentralized composition solution for IoT services. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2020)
42. Khalid, U., Asim, M., Baker, T., Hung, P.C., Tariq, M.A., Rafferty, L.: A decentralized lightweight blockchain-based authentication mechanism for IoT systems. Clust. Comput. **23**, 2067–2087 (2020)
43. Kumar, R., Venkanna, U., Tiwari, V.: Opti-PUM: an optimal policy update mechanism for link failure prevention in mobile SDWM-IoT networks. IEEE Syst. J. (2020). https://doi.org/10.1109/JSYST.2020.3009325
44. Pavithran, D., Shaalan, K., Al-Karaki, J.N., Gawanmeh, A.: Towards building a blockchain framework for IoT. Clust. Comput. **23**, 2089–2103 (2020)
45. Miloslavskaya, N., Tolstoy, A.: IoTBlockSIEM for information security incident management in the internet of things ecosystem. Clust. Comput. **23**, 1911–1925 (2020)
46. Puri, V., Kumar, R., Van Le, C., Sharma, R., Priyadarshini, I.: A vital role of blockchain technology toward Internet of vehicles. In: Handbook of Research on Blockchain Technology, pp. 407–416. Academic Press, New York (2020)
47. Raspberry Pi Documentation: https://www.raspberrypi.org/documentation/. Accessed 12 June 2020
48. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: beyond bitcoin. Appl. Innov. **2**(6–10), 71 (2016)
49. Swan, M.: Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc., Beijing (2015)
50. Dapps: Retrieved from https://ethdocs.org/en/latest/contracts-and-transactions/developer-tools.html (n.d.). Accessed 12 June 2020
51. Silvio: (2018, October 2).https://urn.nsk.hr/urn:nbn:hr:190:464395
52. Ethereum foundation: Solidity documentation (2017)
53. Sultan, A., Mushtaq, M. A., Abubakar, M.: IOT security issues via blockchain: a review paper. In: Proceedings of the 2019 International Conference on Blockchain Technology, pp. 60–65 (2019)
54. Al-Turjman, F. (ed.): Security in IoT-Enabled Spaces. CRC Press, Boca Raton (2019)
55. Ganache Server: https://github.com/trufflesuite/ganache-cli. Accessed 6 June 2020
56. Hammi, M.T., Hammi, B., Bellot, P., Serhrouchni, A.: Bubbles of trust: a decentralized blockchain-based authentication system for IoT. Comput. Security **78**, 126–142 (2018)
57. Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P.: LSB: a lightweight scalable blockchain for IoT security and anonymity. J. Parallel Distrib. Comput. **134**, 180–197 (2019)
58. Shabandri, B., Maheshwari, P.: Enhancing IoT security and privacy using distributed ledgers with IOTA and the tangle. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 1069–1075. IEEE (2019)
59. Watanabe, H., Fan, H.: A novel chip-level blockchain security solution for the Internet of Things networks. Technologies **7**(1), 28 (2019)
60. Du, M., Wang, K., Liu, Y., Qian, K., Sun, Y., Xu, W., Guo, S.: Spacechain: a three-dimensional blockchain architecture for IoT security. IEEE Wirel. Commun. **27**(3), 38–45 (2020)
61. Yazdinejad, A., Parizi, R.M., Dehghantanha, A., Zhang, Q., Choo, K.K.R.: An energy-efficient SDN controller architecture for IoT networks with blockchain-based security. IEEE Trans. Serv. Comput. (2020). https://doi.org/10.1109/TSC.2020.2966970
62. Singh, S.K., Rathore, S., Park, J.H.: Blockiot intelligence: a blockchain-enabled intelligent IoT architecture with artificial intelligence. Fut. Gen. Comput. Syst. **110**, 721–743 (2020)
63. Speedtest-cli: https://pypi.org/project/speedtest-cli/. Accessed 18 Sept 2020
64. IPerf: https://www.linode.com/docs/networking/diagnostics/install-iperf-to-diagnose-network-speed-in-linux/. Accessed 18 Sept 2020
65. Li, H., Pei, L., Liao, D., et al.: BDDT: use blockchain to facilitate IoT data transactions. Cluster. Comput. (2020). https://doi.org/10.1007/s10586-020-03119-w
66. Niranjanamurthy, M., Nithya, B.N., Jagannatha, S.: Analysis of Blockchain technology: pros, cons and SWOT. Clust. Comput. **22**(6), 14743–14757 (2019)
67. Alfandi, O., Khanji, S., Ahmad, L., Khattak, A.: A survey on boosting IoT security and privacy through blockchain. Clust. Comput. (2020). https://doi.org/10.1007/s10586-020-03137-8
68. Neiheiser, R., Inácio, G., Rech, L., Fraga, J.: HRM smart contracts on the blockchain: emulated vs native. Clust. Comput. **23**, 2105–2122 (2020)
69. Alazab, M., Alhyari, S., Awajan, A., Abdallah, A.B.: Blockchain technology in supply chain management: an empirical study of the factors affecting user adoption/acceptance. Clust. Comput. (2020). https://doi.org/10.1007/s10586-020-03200-4

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

Vikram Puri Vikram Puri is a Researcher at the Center of Simulation and Visualization, Duy Tan University, Da Nang, Vietnam. He has Master of Science in Computer Science at Duy Tan University, Vietnam and Bachelor of Technology in Electronics and Communication at Punjab Technical University, Punjab, India. He has total 6 years of Industrial experience and delivered many workshops and seminars regarding the new technologies in collaboration with industrial organization. He is also working as consultant for the corporate. He has written research papers and articles in the international journals and conferences. He is also acts as reviewers for the SCI journals and conferences. Currently, his area of research in Internet of Things (IoT), Embedded System, Trustworthy AI and Blockchain Technology.

Ishaani Priyadarshini Ishaani Priyadarshini is a Ph.D. Candidate at the University of Delaware, USA. She obtained her Master's Degree in Cybersecurity from the University of Delaware. Prior to that she completed her Bachelor's degree in Computer Science Engineering and a Masters degree in Information Security from Kalinga Institute of Industrial Technology, India. She has authored several book chapters for reputed publishers and is also an author to several publications for SCIE indexed journals. As a certified reviewer, she conducts peer review of research papers for prestigious IEEE , Elsevier and Springer journals and is a part of the Editorial Board for International Journal of Information Security and Privacy (IJISP). Her areas of research include Cybersecurity, Artificial Intelligence, and HCI.

Raghvendra Kumar Dr. Raghvendra Kumar is working as Associate Professor in Computer Science and Engineering Department at GIET University, India. He received B. Tech, M.Tech and Ph.D. in Computer Science and Engineering, India, and Postdoc Fellow from Institute of Information Technology, Virtual Reality and Multimedia, Vietnam. He serves as Series Editor Internet of Everything (IOE): Security and Privacy Paradigm, Green Engineering and Technology: Concepts and Applications, publishes by CRC press, Taylor & Francis Group, USA, and Bio-Medical Engineering: Techniques and Applications, Publishes by Apple Academic Press, CRC Press, Taylor & Francis Group, USA. He also serves as acquisition editor for Computer Science by Apple Academic Press, CRC Press, Taylor & Francis Group, USA. He has published number of research papers in international journal (SCI/SCIE/ESCI/Scopus) and conferences including IEEE and Springer as well as serve as organizing chair (RICE-2019, 2020), volume Editor (RICE-2018), Keynote speaker, session chair, Co-chair, publicity chair, publication chair, advisory board, Technical program Committee members in many international and national conferences and serve as guest editors in many special issues from reputed journals (Indexed By: Scopus, ESCI, SCI). He also published 13 chapters in edited book published by IGI Global, Springer and Elsevier. His researches areas are Computer Networks, Data Mining, cloud computing and Secure Multiparty Computations, Theory of Computer Science and Design of Algorithms. He authored and Edited 23 computer science books in field of Internet of Things, Data Mining, Biomedical Engineering, Big Data, Robotics, and IGI Global Publication, USA, IOS Press Netherland, Springer, Elsevier, CRC Press, USA.

Chung Van Le Chung Van Le is Vice-Director Center of Visualization and Simulation. He has a MSc in Computer Science of DuyTan University, 2011, Vietnam and a BSc in Computer Science at Da Nang University, 2004, Viet Nam. He is currently pursuing a PhD at Duy Tan University, Vietnam. He teaches at Duy Tan University, Da Nang, Vietnam. He has a total academic teaching experience of 7 years. He researches on field medical image processing, e-Health, virtual simulation in medicine. He was the director of eUniversity eLearning Center, 2012–2014, and Vice Director of the Center of Software Engineering Duy Tan University, Da Nang, Viet Nam, 2005–2012. He is a Member of the R&D team for system customized advertising content in real-time context-recognition technology using automated monitoring and users, for the Ministry of Education and Training, Viet Nam, 2016–2018. He is the Leader of Preservation for the Hoi An Architectural Cultural Heritage through 3D digitization for the Quang Nam City People's Committee, 2016–2017. He is Duy Tan University Leader Software Develop 3D virtual body system for teaching anatomy and virtual endoscopic techniques for medical students, 2014-2016.