Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.027675
Article

Tech Science Press

An Efficient SDFRM Security System for Blockchain Based Internet of Things

Vivekraj Mannayee[1,*] and Thirumalai Ramanathan[2]

[1]Faculty of Mechanical Engineering, Anna University, Chennai, 600025, Tamil Nadu, India
[2]Department of Mechanical Engineering, Dr. N. G. P. Institute of Technology, 641048, Tamil Nadu, India
*Corresponding Author: Vivekraj Mannayee. Email: vivekraj3phd@gmail.com
Received: 23 January 2022; Accepted: 14 March 2022

Abstract: Blockchain has recently sparked interest in both the technological and business firms. The Internet of Things's (IoT) core principle emerged due to the connectivity of several new technologies, including wireless technology, the Internet, embedded automation systems, and micro-electromechanical devices. Manufacturing environments and operations have been successfully converted by implementing recent advanced technology like Cloud Computing (CC), Cyber-Physical System (CSP), Information and Communication Technologies (ICT) and Enterprise Model, and other technological innovations into the fourth industrial revolution referred to as Industry 4.0. Data management is defined as the process of accumulation in order to make better business decisions, and process, secure and store information about a company. In the incipient model, there are interconnected contrivances and Machine-to-Machine (M2M) interactions, and transaction data are stored on the Blockchain. Security is a challenging aspect that must be punctiliously considered during the design and development phases of a CSP. In this research article, we proposed a Secure and Distributed Framework for Resource Management (SDFRM) in Industry 4.0 environments within a distributed and collaborative Industry 4.0 system, the dynamic and trust-based Distributed Management Framework (DMF) of shared resource access. Such issues are focused by taking into account of the traditional characteristics of IoT/Industrial Internet of Things' (IIoT)-predicated environments, an SDFRM in Industry 4.0 environments within a distributed and collaborative Industry 4.0 system. Also, to ensure strong privacy over the procedures associated with Access Control (AC), a privacy-preserving method is proposed and integrated into the DMF. The proposed DMF, based on blockchain technology and peer-to-peer networks, allows dynamic access management and system governance without using third parties who could be attacked. We worked hard to design and implement the proposal to demonstrate its viability and evaluate its performance. Our proposal outperforms the Multichain Blockchain in terms of successful storage transactions with an achieved average throughput of 98.15%.

Keywords: Internet of things; blockchain; data privacy; security system; distributed management; IIoT

1 Introduction

The advancement of Information Technology (IT) in the fields of wireless communications has caused an impact in the early phases of an emerging IT known as IoT, which represents the integration of physical smart devices within the Internet set-up. It has significantly emerged, and gradually affected human beings' daily routine lives, promoting an early generation of *state-of-the-art* and valuable services provided by a variety of application/industrial domains [1]. The most crucial topic in the IT industry right now is blockchain technology. Business leaders attempt to be prompted by cryptocurrency and Smart Agreement System (SAS) benefits, and blockchain has recently made headlines in business and technological news. In December 2021, there were 5.66 billion active ecumenical internet users, accounting for 60.15% of the ecumenical population. Every digital action is recorded and tracked, which could lead to the disclosure of sensitive information, thanks to the rapid advancement of technology. In such a paradigm, smart connectivity, remote sensing, and computation are all necessary components. They are integrated into manufacturing environments and processes, as well as other technologies such as CC, CPS, ICT, and Enterprise Model. Industry 4.0, the fourth wave of the industrial revolution, has arrived [2].

A CPS is a complex entity made up of physical and software components that exhibit specific characteristics, multiple and distinct behavioural modalities, operating on different temporal and spatial scales, and interact with one another in context-dependent ways. As a result, a CPS is primarily composed of interacting physical, human, analogue, and digital system components that are designed to operate by utilizing logic and integrated physics. CPSs are expected to make significant developments in emergency replication, personalize health care, electric power generation and distribution, traffic flow management, and various areas that exist currently [3].

Data are stored in a traditional data management system database, and data users/organizations can open AC. Due to various security concerns, many IoT/IIoT systems can never be deployed ecumenically unless congruent security solutions are implemented. As a result [4], they ensure that it is critical to have secure and trusted communication between heterogeneous devices as well as within dynamic and decentralized environments. This is done for getting user acceptance and preventing exchanged data glommed/tampered with malicious cyber threats who disrupt engendered processes and render various inoperable devices. Building a secure cyber system, on the other hand, entails not only protecting data exchange but also designing a system in which all participating devices and stakeholders trust the source of data and the data itself.

A more decentralized approach is optically determined as the solution for the long-term magnification of both IoT/IIoT by amending privacy and security in such environments. This primarily refers to the fact that when participating entities interact and collaborate, they are not required to rely on and trust external services and third parties' right to generate or share data [5]. Several proposals have been completed to ascertain distributed systems in IoT environments and reap scalability and security benefits. However, the latter are incapable of easily addressing the integrity, immutability, traceability, and notarization required for the most available cases in such environments. We proposed an SDFRM. The proposed framework implements precision, flexibility, and secure resource sharing access by installing the Open AC model. Blockchain maintains a continuous record of the flow of resources distributed and shared among cooperating parties' access privileges.

2 Related Works

Academics, researchers, and entrepreneurs have been drawn to the IoT because of its ability to provide innovative solutions across a wide range of applications. The IoT is a physical network that connects disparate devices and objects to control and manage sensing, processing, and communication processes without human intervention. With the introduction of smart homes, smart cities, and other smart things,

the IoT has grown in importance, opportunity, and development, with more than 50 billion connected devices that are expected by 2020 [6]. Wireless Sensor Networks and M2M/CPS are two network technologies that have emerged as critical components of the broader term of IoT in the related research works. As a result, in the IoT, using the standard Internet Protocol, security concerns about Wireless Sensor Network (WSN), and M2M/CPS arise, necessitating a security threat for the entire network system. On the other hand, malicious attacks have the potential to disrupt IoT services while also threatening data security, user privacy, and network confidentiality [7].

With blockchain technology's promise of data security and privacy benefits, one would expect the general public to embrace it unless they are unaware of these benefits. On the other hand, Blockchain is still in its infancy [8]. Due to its encrypted method, severe financial organizations adopted Bitcoin, the first digital currency built on Blockchain, as online payment transfer can be done conveniently without a centralized system; concerns about this technology, on the other hand, remain a significant challenge. A few years after its introduction, several security flaws in the cryptocurrency bitcoin were discovered. The authors discuss Wallet Software Attacks, Time jacking Attacks, the '*>50%'* Attack, Double-spending, and Selfish Mining Attacks, among other bitcoin security concerns and issues. The author also addressed blockchain technology issues, such as Majority Attacks (51% attacks) and forks [9].

There are three types of decentralized applications (dApps) in the same Ethereum network [10]. The first type is a dApp, which is fundamentally a simple agreement that distributes value between parties predicated on the SAS stored on the Ethereum blockchain. A dApp, on the other hand, deals with money but requires data from sources other than the Blockchain. Determinately, dApps can be utilized for a wide range of other purposes, including data storage. On the Ethereum network, three types of dApps can be run. The first type is a dApp, which is essentially a simple SAS that distributes value between parties when certain conditions are met predicated on the SAS stored on the Ethereum blockchain [11]. A dApp, on the other hand, involves money but requires information from sources other than the Blockchain. Conclusively, dApps can be utilized for various other applications, including data storage.

Permissioned blockchains in the public version have known and pre-approved participants/nodes and the ability to identify the one who can modify/control data in blocks and carry out transactions. Federated/consortial blockchains are another names for these. Public authorization blockchains are at the very least viewable by the general public. Private permissioned blockchains, on the other hand, only define those who are allowed to participate in, access, or view blockchain data. Participation is only by request/authorization. In general, the security policy that a blockchain implements defines authorizations' control [12].

A block is the "*basic component of a blockchain,"* according to the researcher, and is essentially the collection of data [13]. A block, in general, provides direct access to the previous blocks, some previous transactions, and a general agreement technique for securing its correctness. Due to their inherent differences, the operational details of the various types of blockchains are distinct. The fundamental model of Blockchain technology revolves around cryptographic hash functions in their most primitive form [14].

SAS is nearly impossible to discuss without mentioning applications on the Blockchain (particularly Ethereum) [15]. Perceptive SASs, as previously stated, are just computer codes that use a consensus algorithm to enable simple financial transactions. As the author points out, one of the goals of SAS, like Ethereum, is to have "*highly specialised reusable components"* that can be used to develop more greatly involute SAS, having allowed a smaller blockchain size, and many of the advanced manufacturing real-time applications discussed later are focused primarily on SAS [16].

3 Proposed System

3.1 The Model of Internet of Things-Blockchain

This section describes a blockchain-based decentralized AC method and a resource limit assignment policy. The method is built on the blockchain platform (Fig. 1), which enables the use of SASs in conjunction with a blockchain [17].

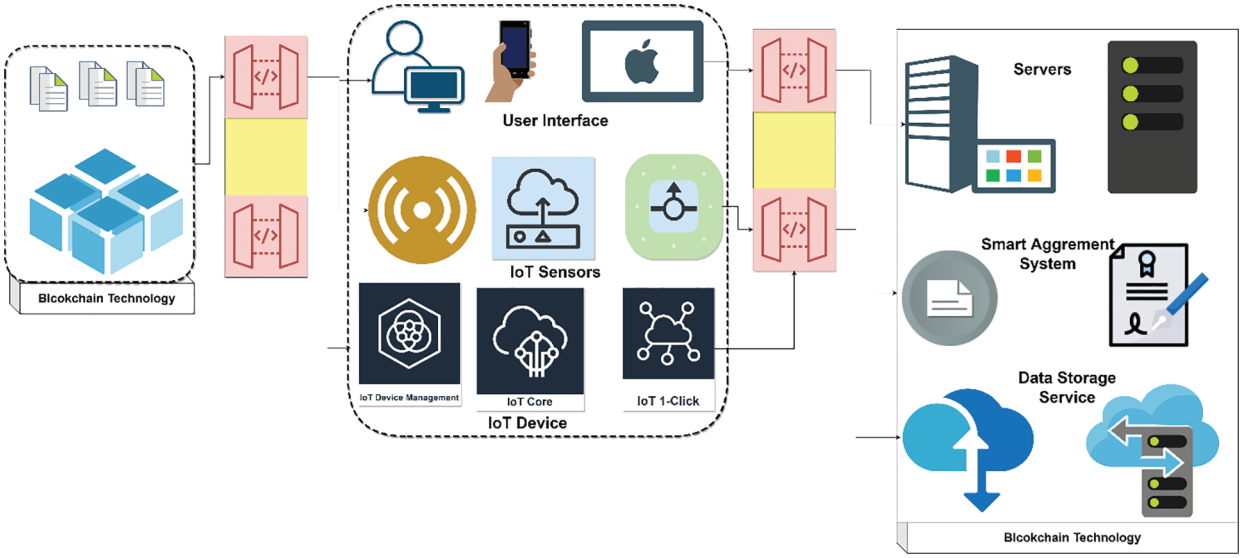

Figure 1: Proposed blockchain-IIoT model

With the introduction of DMF, blockchain technology regained the status of DMF. A good decentralized autonomous organization is an independent one that is not controlled by a centralized authority. Consider DMF's use of a taxi service combined with a self-driving vehicle. Assume that the owner can programme cars and set up taxi service guidelines, such as connecting to Internet-based taxi service providers to get users. The car is driven out on its own and began providing taxi service to passengers as directed by Taxi Service once the owner has completed the initial set-up. When the car runs out of gas while staying in a hotel, it drives itself to a recharging station, refuels with the money it has accumulated, and returns to the hotel. This section provides a high-level overview of how Blockchain works for those interested in learning more. The primary function of Blockchain is to add transaction records to a public ledger that keeps track of previous transactions. A group of records is referred to as a block. Because it is a chain of blocks, the public ledger of past transactions is known as the Blockchain [18]. When a transaction occurs, the Blockchain is responsible for informing the network. A Blockchain network user verifies the transaction's validity and secures legitimate data transactions from being misused/altered.

Fundamental Quantitative Principles and Definitions

The fundamental scientific methods covered in this study require a basic understanding of number theory principles and a summary of the mathematical models and functions that support blockchain security and privacy. Many mathematical computations, such as finite fields and, as a result, Elliptic Curves (EC) defined over finite fields, are used in EC supported by modular (MOD) arithmetic, a fundamental concept in blockchain technologies [19].

Example 1: Let us assume that {P, *Q, M}* are set of integers; *{a, b}* are corresponding to the MOD {M*}* if they have the identical value *{R}* as a reminder when they are sub-divided by {M*}*, Eq. (1)

$$\forall P,\ R,\ M \in Z,\ where,\ M > 0;\ P \equiv R\%M,\ if(m)|(P-R) \tag{1}$$

For case, in MOD 7, 26 corresponds to 5; (26 ≡ 5%7), denotation of "26–5" is 21, which is divided by 7.

Formulation 1: It is possible to declare any integer value P∈Z in the form; Eq. (2)

$$P = QxM,\ where,\ 0 \le R < M[\because R \in \{0,\ 1,\ 2,\ 3,\ \ldots,\ (M-1)] \tag{2}$$

For example, (31 = 47 + 3); and thus (13%7). Although there are infinite integer probability values for the remainder 'R' for any specified MOD, the number 31 is consistent with (03/10/17/-4/-11). Groups, fields, and EC are some of the other primary concepts introduced here.

Example 2: A 'G' is a set of values that are closed under a function '∘', such that, Eqs. (3)–(5)

$$P \circ Q = Z \in G,\ \forall P,\ Q \in G(P\circ) \circ Z = P(Q \circ Z);\ \forall P,\ QZ \in G \tag{3}$$

$$\exists e\colon (P \circ e) = (P \circ e) = P \tag{4}$$

$$\forall P,\ Q \in G \exists P-1\colon (P \circ P-1) = (P \circ P-1) = e;\ \forall P \in G \tag{5}$$

G is an algebraic expression; *if* it is also divisible in the context that (P, Q) = (Q, P); P, Q; It's worth noting that an array's order in a group is the least decimal digits '*k,*' which the assumed section incremented to get the individuality value; Eq. (6)

$$P \otimes P \otimes P \otimes \ldots . \otimes P \otimes P = Pk = e \tag{6}$$

A cyclic group has a division with the top order, known as a primitive value, which can generate every other value in the group.

Example 3: A Field is a set F that sets minimum criteria:

(F, +) and (F0, •) are matrix multiplication group with a division of identity of 0 and 1

$$(P+Q) \circ Z = (P\circ) + (Q \circ Z) \forall P,\ Q,\ Z \in F \tag{7}$$

The order of a field determines whether it is a Finite/Galois Field. This section concludes with recommendations of the EC, Eq. (7)

Example 4: In EC, E is the result of the research to the given formula in cryptographic operations, Eq. (8)

$$Q^2 = P^3 + aP + Q/F_p,\ where,\ P > 3\&\&a,\ b \in F_p\colon 4P3 + 27Q2 \ne 0\%\ M \tag{8}$$

Consider the general linear model: Y2 = x3 + b; over a finite prime set F_p. Adding points on an EC consists of two components for equal and significantly different points. Eq. (9)

$$If\ point\ P = (P1,\ Q1)\ and\ Q = (P2,\ Q2),\ then\ (P1,\ Q1) + (P2,\ Q2) \tag{9}$$

It gives a new point, Eqs. (10)–(12)

$$R = (P3,\ Q3) \tag{10}$$

$$\begin{aligned} P3 &= S2 - P1 - P2 \\ Q3 &= S(P1 - P3) - Q1 \end{aligned} \tag{11}$$

$$S = (Q2 - Q1)(P2 - P1)\%\ P;$$
$$IF(P \neq Q) \tag{12}$$
$$S = (3x12+)2Q1\%\ P;\ IF(P = \Theta)$$

3.2 Preferences on Internet of Things and Industrial Internet of Things

In a well developing IT, IoT is expected to provide promising solutions that revolutionize our conduct and services across a wide range of industries, including healthcare, conveyance, energy, and manufacturing. The primary vision of this paradigm is to build a perspicacious world by connecting the physical, sensing, processing, and analytics worlds to the digital, cyber, and virtual worlds on a global scale, using an involute network that connects billions of devices, objects, housings, and people into a multi-technology/protocol/platform. Emerging technologies such as Big Data and analytics, CC, Machine Learning (ML), and the IoT are increasingly integrated into manufacturing sectors and processes, ushering in Industry 4.0. This revolution includes the implementation of highly flexible and efficient supply chains, on-demand manufacturing, logistics operations, manufacturing processes, newly emerging services, and acceptance of smart factories and cloud-based applications. The Smart Factory (SF) is a modernized version of the traditional factory that uses CPS to monitor and make decentralized decisions about the industry's physical processes via the IoT [20].

Thus, a core concept of Industry 4.0 and the SF is the utilization and integration of emerging IT within industrial and manufacturing processes, which could produce authorization to operate in an efficient, flexible, and cost-effective manner while maintaining consistently high quality and low cost. As a result, everything in and around the manufacturing supply chain is interconnected, including machines, data, processes, suppliers, customers, distributors, and even the product itself.

3.3 A Secure and Distributed Framework for Resource Management for IIoT

An SDRMF for Industry 4.0 Environments is introduced in this section (Fig. 2). This last one employs blockchain-based keen SAS to improve shared resource management and utilization while dynamically managing access authorizations over the resources in question. Additionally, to improve the security standards, this framework applies the principles of trust management into the Open-AC model. A trust system is used to evaluate the behaviour of access requester entities in this scenario to ensure the dynamic behaviour of security policies that are defined and validated as a function of the behaviour of the access requester entity.

Blockchain technology has been applied to a wide range of applications and AC models; as can be seen visually, only a few types of research have concentrated on implementing blockchain technology into open AC technologies in the event of smart factories. In most of the SF settings, our research goal is to not only focus on providing precision, flexible, and secure distributed resource access permission, but also to enhance distributed and dynamic management and administration of the overall system. It is here that everything relevant to this discussion components can make an effect of the control about integrating entities registration and agreement management for entities requesting mining authorizations. The integration of trust management with the Open AC model is another focus of this framework, determining whether subjects are trustworthy and well-deported enough to open resource accessed data.

Figure 3.5 depicts the overall method of the proposed system. Each domain (SF1, SF2, SF3) has entities that perform different functions, such as a human worker in the manufacturing subfactors SF1 and a supervisor agent in the quality control subfactors SF3. With each shared resource, the latter could do various things, such as using shipment vehicles, sharing trust records, and searching for available ones. Assume that a human worker in SF1 needs to use the shared resource Truck 3 to finish a shipment. Both

the agent and the authentication system that make authorization decisions verify contrivance identities and generate authorization tokens that must be authenticated.

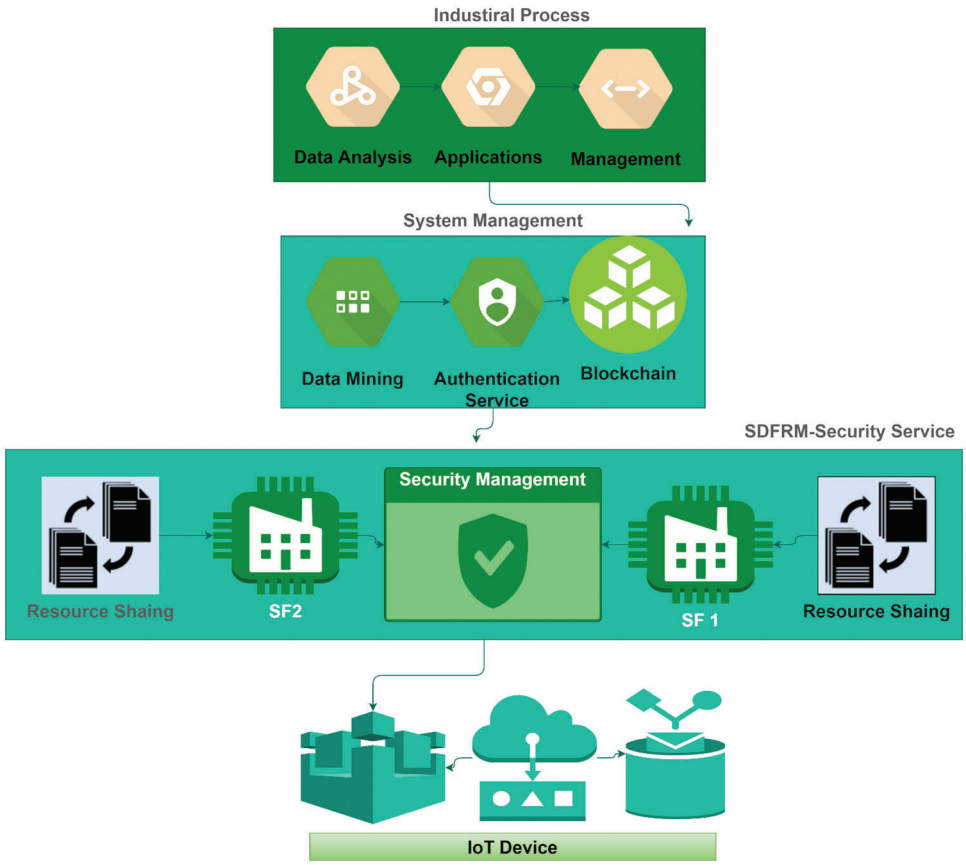

Figure 2: SDFRM model

IoT devices are adapted to collect and analyse data from smart products, other mobile devices, and smart services in the SF environment. SDFRM modules are installed in more reliable network components connected directly to each device unless implemented within the device/entity itself. An SDFRM trust module is also used to assess the request entity's integrity and evaluate its behaviour using various trust factors and criteria, rather than static analysis of the access request. The results of this evaluation are incorporated into the context structure and then sent to the SDFRM policy-manager-module, which is responsible for formulating the corresponding transaction and broadcasting it to the SDFRM distributed network, where the corresponding perspicacious SAS is implemented via the SDFRM client.

3.4 Process Flow of Secure and Distributed Framework for Resource Management Model

Fig. 3 describes the various phases and operations of the proposed framework. This final one entails a series of functions that include:

a) Reaching desired agreement among collaborating parties who are registered and involved in the agreement method, and once identified, a SDFRM distributed network is created, and a grouping is defined.

b) If shared resources exist, they must be identified to be able to access and use in an authenticated method.
c) To do so, security rules that control access among collaborators must be defined first.
d) After their behaviour has been evaluated, entities requesting to join the system/participate in the agreement methods are identified and assigned roles.
e) When a requesting entity wants to use an existing resource, it sends an access request, which is then processed.

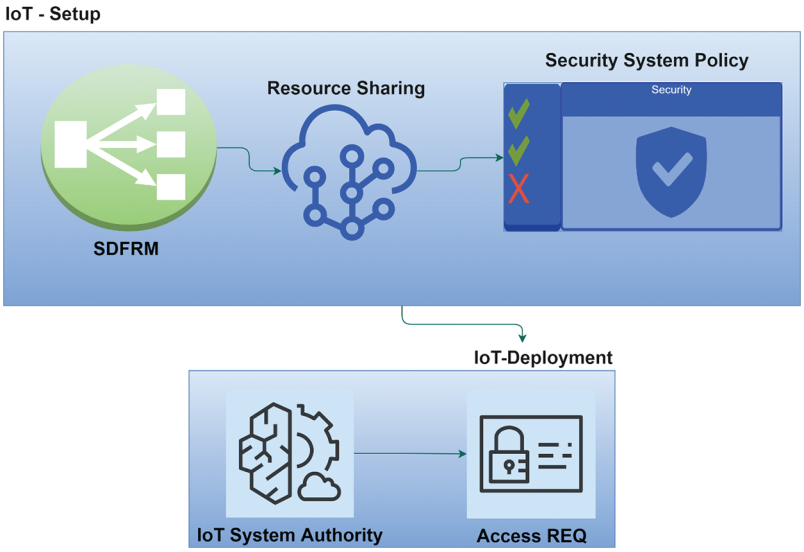

Figure 3: Flow of phases: SDFRM framework

3.5 A Secure and Distributed Framework for Resource Management: Creating a Network and Registering Functions

Integrating incipient entities with various roles and relegations can begin once the DMF is well-known and operational. The authentication system assumes that emerging entities have already been identified and have a unique public identifier. It's also possible that they were given an Ethereum address, which is required to use the Ethereum network. Before adding a new entity, the group of agreement entities tests to make sure that its public identifier matches its pretended role and is acceptable for the classification recommended by it.

To manage access to the shared resource, open AC policies should be embedded into the open AC smart SAS and decided to share across the SDFRM. The proposed framework performs the functions depicted in the Unified Modelling Language diagram in Fig. 4.

3.6 Deployment of Cloud Computing Based IIoT

For the prototype's implementation, four Virtual Machines (VM) were created. Ubuntu 18.04 with the following designations: Storage of 50 GB, 2 Central Processing Unit (CPU), and 4 GB of recollection. All of the prerequisites was installed precisely as they would in local implementation. Few quandaries arose during the prerequisites' installation because some software versions were incompatible with one another. As a result, a few new and updated environmental variables were integrated to ensure Golang's smooth operation. Organizations must check the user details when requesting information from the admin; system performance in terms of transaction throughput and replication time are all important reasons for deploying Hyperledger Fabric across multiple VMs, as previously stated. As a result, using multiple VMs

to run High Frequency (HF) yielded better results. It is also more distributed if multiple organizations are located on different EC-2 instances.

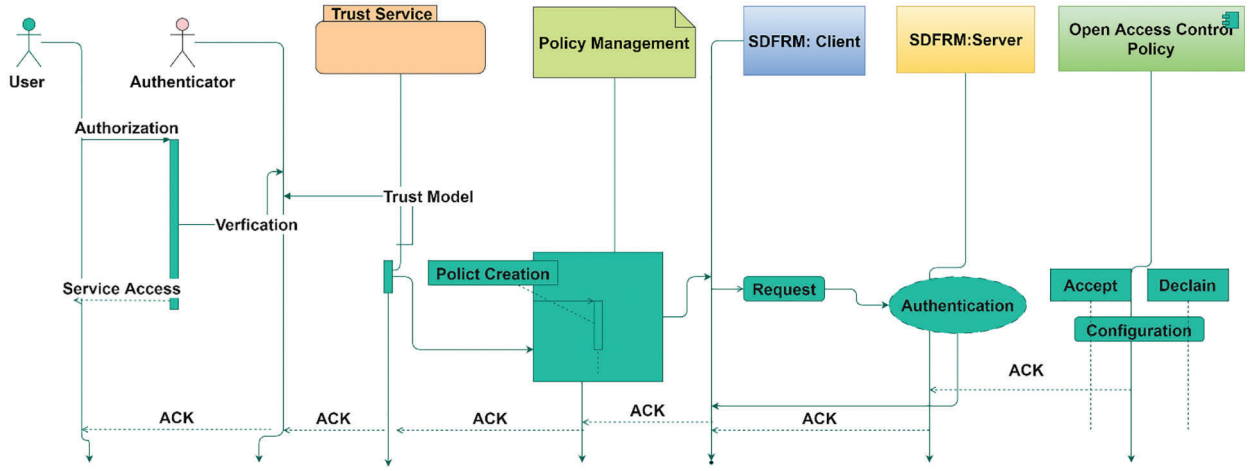

Figure 4: Access control policy-data driven transactional system

Crypto resources were primarily prepared for 3 organizations and also for organizations with a single order; each comprises a Central Ascendancy (CA), 2 ledgers, and 2 peers. There are 3 CAs, 6 ledgers, and 6 peers. There is a CA and three orders in the order organization in terms of generating certificates. The organization's Membership Service Provider (MSP) is intended for all participants. In the development of the genesis block, the organization's MSP is crucial. It's the 1st block that doesn't have any transaction information. It does, however, involve the MSP IDs and certificates of the three specific organizations. In the channel configuration transaction, the channel association and denomination are included. The genesis block and channel T_x*'s* evolution are depicted in Fig. 5.

The most recent Fabric releases take a sophisticated chaincode deployment and development method. In a VM, all certificates are created and then transferred via Secure File Transfer Protocol to their respective machines. A docker swarm network was built to allow them to communicate with one another. The addition of organizations to the channel is depicted in Fig. 6. After installing Fabric on all VMs, the focus was shifted to developing a chaincode that could be installed and utilized on peers. According to the most recent chaincode lifecycle, the peers are packaged, installed, and committed to the chaincode. It's worth noting that the term "*Chaincode Lifecycle"* refers to the complete process, as defined in Fabric 2.0. The following data can be stored using a chaincode: denomination, email, consent, and organization. An approving peer is a member of an organization's peer group. The chaincode is implemented by the endorsing peer. The Hyperledger Fabric Node Software Development Kit was habituated to record data on the Blockchain.

The front-end, which is used to invoke chain code functions via Application Programming Interface (API) endpoints, was built with React. Users can directly store data/information from reacting applications on the cloud database and indite consent details on the Blockchain once they have been authenticated. They can also visually detect whether any messages or notifications from organizations requesting additional information have arrived. Organizations that join the network can use Chaincode's features to see the details of network users' consent, whereas even after the user has given their approvalyou can request full access from the admin. The cloud database is an Amazon S3 storage block in which users can upload files. Because it is a private block, it is inaccessible to the general public (Fig. 7).

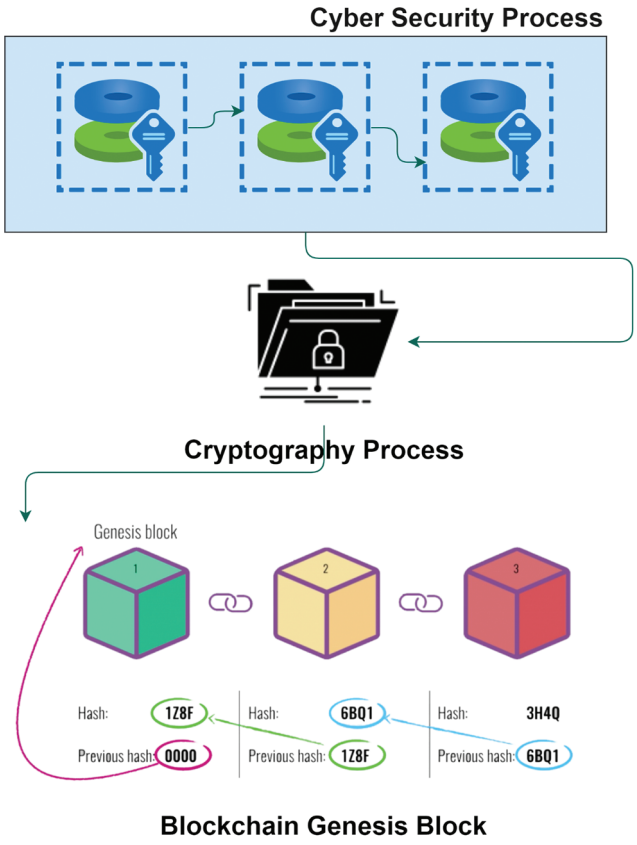

Figure 5: Deployment of the blockchain network

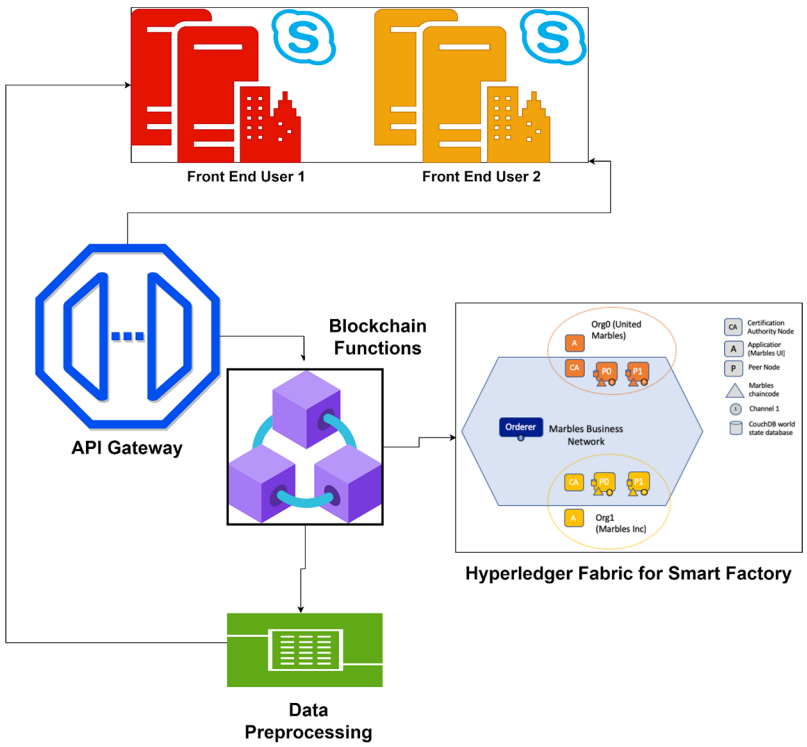

Figure 6: Set-up of VM with API

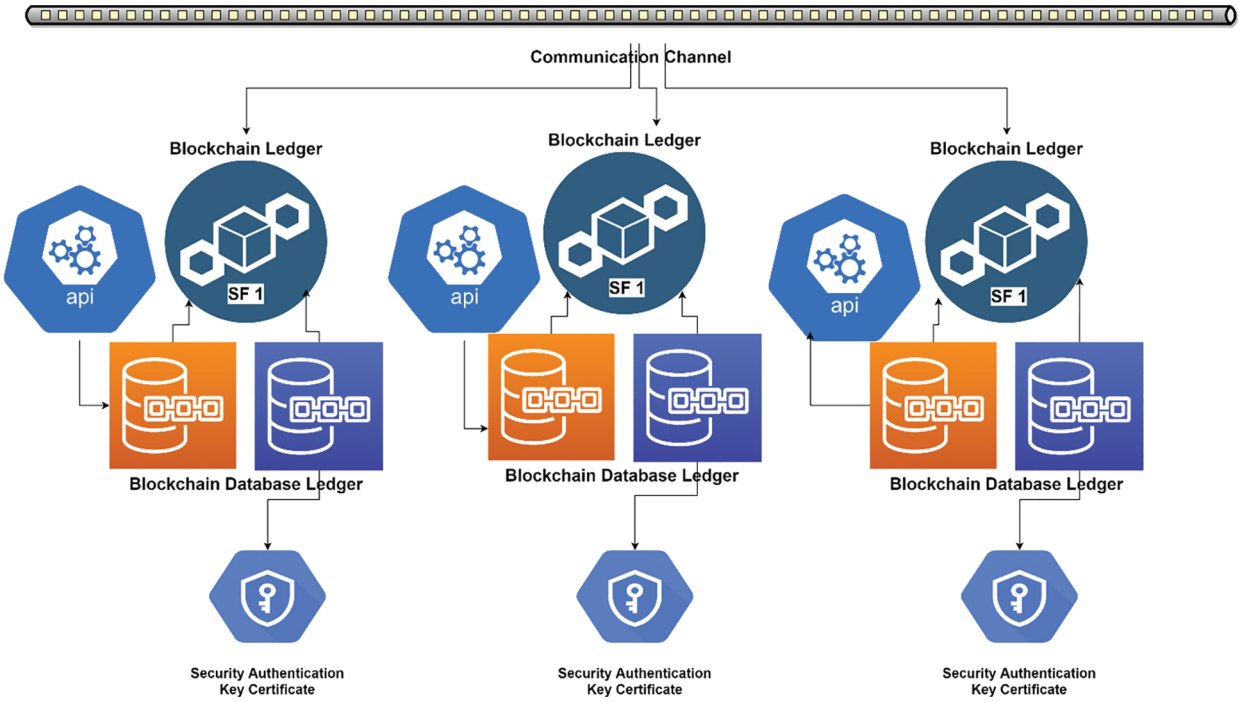

Figure 7: Overall blockchain network communication

Furthermore, it has been subjected to mandatory safeguards to defend and secure it. The following are the most significant implementation issues:

- Gaining a thorough understanding of blockchain ideas to create a cohesive framework for private data management.
- The use of a variety of software to assist with fabric installation.
- There is insufficient and overly complicated information on using the Fabric SDK.
- Managing the front-end development of our proposed system, such as Cross-Inception Resource Sharing (CORS).

4 Result and Discussion

A computer with an Intel Core *i*5-3210 M processor and 6 GB of Random Access Memory (RAM) can mine Ethereum, a Dell Precision M6800 VM with a 4th Generation Intel Core *i*7 processor and 8 GB of RAM. The researchers built a 3-node private Ethereum blockchain network with Ethereum mining functionality, a computer with an Intel Core *i*5-3210 M processor running at 2.40 GHz on which an Ethereum node was set up. Initially, a public blockchain was used to design and develop the system (Ethereum). The system was then tested using a private blockchain on a single VM (Hyperledger). This section describes the outcomes of Ethereum's implementation.

We chose Ethereum as the platform for putting our proposal into action because it is the most popular blockchain platform for developing perspicacious SASs right now. We utilized the Geth client, which was installed on each entity and configured to act as an Ethereum node. To implement the ring signature method, we utilized the Secp256K1 method for engendering EC parameters and the signature where ECDSA is performed. This method was called so because of its unique implementation of EC point addition and multiplication by a scalar; the system allows high-speed access results.

Algorithm for Block Concatenation

Step 1. START Function
Step 2. Input as Blockchain_Digit
Step 3. Output as Hash_Function_String
Step 4. Set Resulant Value (RV) = '0|00|000|010'
Step 5. Set Node Value (NV) = '0'
Step 6. While Process
Step 7. For (Block_INDEX = [0, 100] Do
Step 8. BC->Current Value = Block(.)
Step 9. BC->Check Value = length − 1
Step 10. RV = string(txt).ENCODE[]
Step 11. Hash_Value = HF(B_Check_Value, B_Current_Value, RV, NV)
Step 12. If HF(Current Value == SHA-512[0:2] <= NV Then
Step 13. Break
Step 14. Else
Step 15. NV = NV + 1
Step 16. End If
Step 17. End for
Step 18. End While
Step 19. End

4.1 Encryption Method for Smart Agreement Systems

An SAS for the public blockchain (Ethereum) system was created to allow data transfer between the admin and a third party. Figure 5.1 depicts the SAS algorithm. The main reason for including a price in the astute SAS is that, the required Ether , whichis expensive to make it work can be used as a customer incentive that benefits both the organization and the customers. On a test network, the SAS was deployed. When a third party requests data, it sets it off. On an average, transaction cost of deploying this SAS is 0.000413356 Ether and invoking the first function costs 0.0000397623 Ether. After a successful data transfer, the magnitude of Ether used to invoke the second function is 0.000048865 Ether.

Algorithm of SASs

Step 1. Input
Step 1. Trusted User = TU
Step 2. Domain = D
Step 3. Payment = PMT
Step 4. Waiting Time = WT
Step 5. Initial
Step 6. WT for PMT

(Continued)

Algorithm (continued)

Step 7. *WT for Transaction*
Step 8. *WT for Complete_Process*
Step 9. *Begin*
Step 10. *Set Current_State_Value = 0*
Step 11. *Authorize Administrator_User*
Step 12. *FUN (1)*
Step 13. *REQ = WT_PMT*
Step 14. *Do Transaction*
Step 15. *REQ = WT_Tranx*
Step 16. *FUN (2)*
Step 17. *Approve TU*
Step 18. *REQ = PMT*
Step 19. *Complete_Tranx*
Step 20. *End*

4.2 Implementation Using Hyperledger

The configuration of the local VM is shown in Tab. 1. Each organization's peers is installed on separate ports within the same VM in this configuration.

Table 1: Configuration of local VM set-up

Parameter	Set value
Test case	Open-Source linux
Number of processor	8
Memory size	8 GB
Storage size	100 GB

J-meter was used to assess the network's performance. We used 200 threads with a one-second ramp-up time for the test. The rate at which inexperienced concurrent users attempt to access the system during a load test is ramp-up haste. The experiment was carried out to determine the system's throughput using a J-meter load test. The reads' throughput is calculated.

This experiment uses 120 attacks with a one-second ramp-up period, as stated verbally. The test was repeated four times to ensure that the final results were not significantly different. The combined results of all four experiments are shown in Fig. 8. The number of transactions per second is represented by the Y-axis, while the X-axis represents the number of active threads. As shown in Fig. 9, every 200 users resulted in an average of 40 failed transactions. As a result, the success rate of local implementation is estimated to be around 86.79%. We did not achieve flawless prosperous transactions in any of the experiments performed on a local VM.

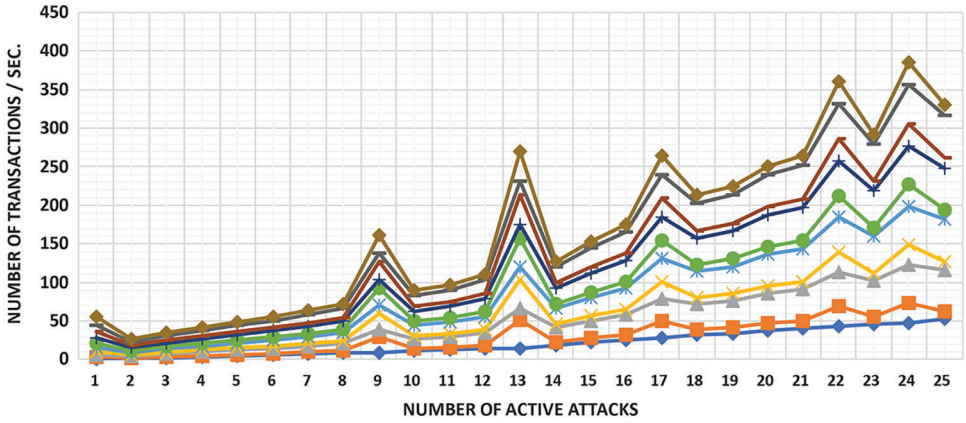

Figure 8: Outcomes of active attacks *vs.* transactions

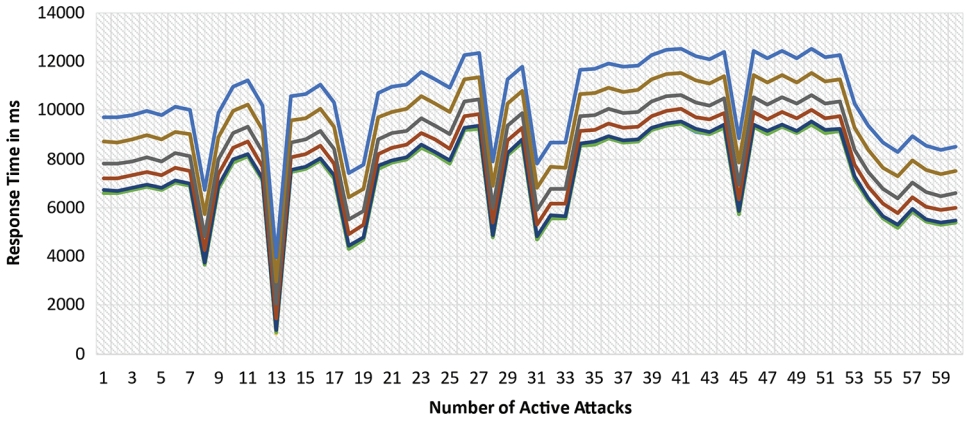

Figure 9: Outcomes of active attacks *vs.* response time

The average throughput is 10.9, with 85.18% success rate. Because the system was built on a single VM, this is the case. The success rate of cloud implementation is low in general. Overall, the read throughput is noticeably lower. The system's replication time is another essential metric to consider. It is also evaluated using a J-meter load test with 200 users. In Fig. 9, the results of all experiments' coalesced replication times are shown.

The average replication time for all results is approximately 5603.5 ms for completed transactions (6.5 s). The failed transactions led to the conclusion that the network on the local VM is unstable.

We investigated the performance time required for the response time of the attacked user while varying the number of users involved in the trust process in the following to demonstrate the feasibility of our proposal for achieving distributed and privacy AC for Industry 4.0 applications. As a result, this time is compared to the performance time required to consummate the entire AC transaction. Tab. 2 exhibits the results of the case study we presented. This section summarises the processing time for user processes as well as the size of attacks.

Figs. 10 and 11 depict the subsisting relationship between the number of ring participants and the time required to perform user processes, as well as the response time. These figures show a linear relationship between the number of public keys corresponding to users and other parameters such as response time and file size. In terms of performance time, the more user who logs in, the less efficient the process

becomes. By comparing this time to the performance time required for executing the perspicacious Access SAS for an entire AC transaction, which is less than 30 s for a 10-users size, we can demonstrate the capability of our framework in achieving distributed privacy-cognizant AC. It's consequential to note that the time it takes to deploy and run SAS strongly is influenced by several factors, including the system's computing power and networking model. As a result, the time cost in the public Ethereum system in the authentic system may differ significantly from this case study.

Table 2: Performance by a user

User level	Time generation (ms)	Time verification (ms)	File size (MB)
1	0.071	0.006	4
2	0.098	0.009	10
3	0.109	0.032	20
4	0.138	0.091	40
5	0.189	0.037	60

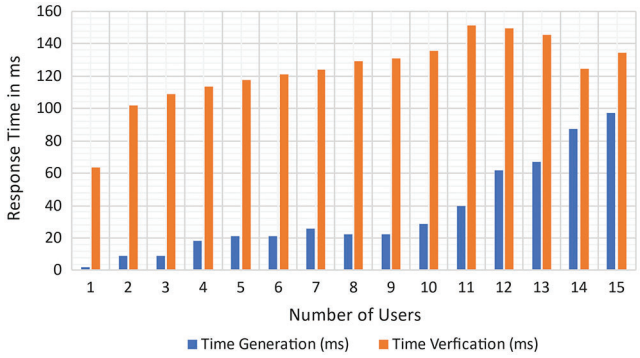

Figure 10: Users *vs.* response time

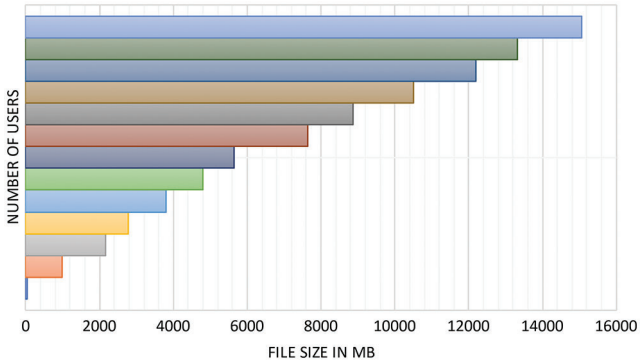

Figure 11: Users *vs.* file size

4.3 Challenges on Attacks

Then we look at how resistant our proposal is to malicious attacks launched by various IoT network devices. First, we look at how a well-comported node's average trust value changes as the number of

misbehaving nodes' launching bad-mouthing attacks changes. The maximum number of nodes in the network is set to 100. In this section, we put our proposal to the test against malicious behaviour by limiting the number of false nodes to 30.18% of the total node count in the first test case and 50.18% in the second test case.

The system is reliable to cyberattacks for up to 45.17% of the overall network nodes. Fig. 12 shows the average trust's research findings towards the true node as the total false node's percentage changes. As the number of malicious nodes increases, so does the time complexity and trust partiality (Tab. 3).

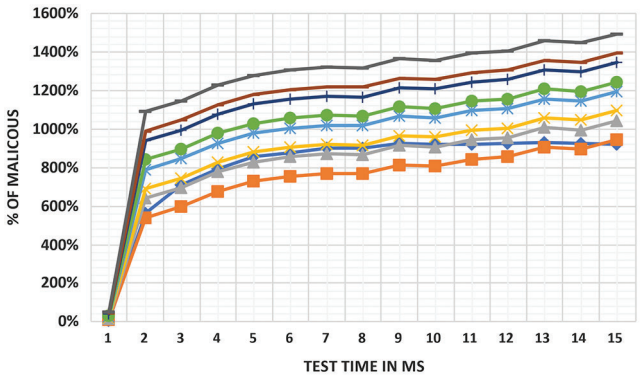

Figure 12: Analysis of trusted node system

Table 3: Simulation parameters

Simulation parameter	Value
Tool	NS 3-3.13
Execution time	2 h.
Area	100 m × 100 m
Nodes distribution	Random access
Number of. nodes	100
Number of false nodes	0% and 50%
Time of trust level	300 s
Starting trust time	0.5
Trust pause	0.1
Multi-chain mining range	0.3
Multi-chain block size	8 Mb
Multi-chain transaction	4 Mb

In Fig. 13, we can see how a malicious node's trust value changes as the percentage of total false nodes launching ballot stuffing and on-off attack changes. The malevolent node alternates between true and false packet distribution behaviour in an on-off attack. The false node, in particular, distributes packets in the defined interval to gain high trust rates; once its trust rates reach 0.95, it begins to misbehave; and once the trust rate falls below 1 and 10, it resumes its true behaviour to increase trust rate.

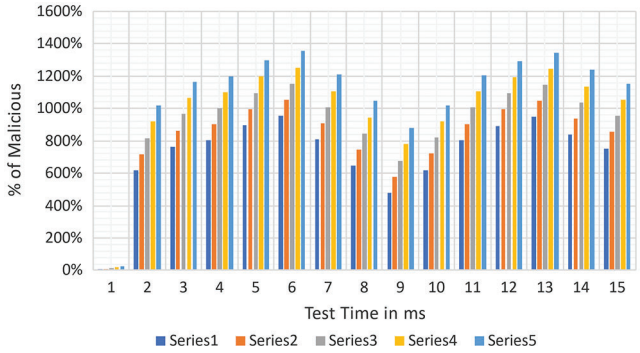

Figure 13: Analysis of malicious node *vs.* trust level

When it comes to the impact of the malicious node percentage, trust values vary significantly more when it is higher because more malicious nodes can collaborate to promote the false node and raise its trust level. The experimental results were predicated because each node computes a new result predicated on trust evaluations from the Δ_t=2300 s in the past. This section proved how blockchain technology can enhance the precision and resilience of a network against one-time attacks. Indeed, because of the Blockchain's traceability feature, the ledger's trust information is time-stamped and aeonian is stored for future use. We can detect such alternative malicious behaviour, for example, by maintaining track of and reviewing log results, and as a result, it prevents the malicious node from achieving high trust rates in the future. As shown in Fig. 13, once the trust rate of the bad node falls below 0 and 5, it resumes normal behaviour, and it is penalized by Blockchain technology that provides this feature, and its rate is close to 0 and 5.

In this section, we change the size of the file containing trust information and trust rates for each evaluated entity in the network to refer to how long it takes to replicate. The files range in size from 5 KB to 10 MB. The average replication time of the blockchain network grows in lockstep with the size of the trust information file, as shown in Fig. 14. It takes 2097 ms for a 10 MB trust file, which reasonably considers the security features of blockchain technology. In addition, we assessed the computational resources required by validator entities to process incoming requests and transactions in this step to determine the efficacy of our proposal, as well as its applicability and suitability for IoT systems. The time the trust information file was sent for storage within the blockchain network and the time a transaction was completed, successful transaction verification would be received.

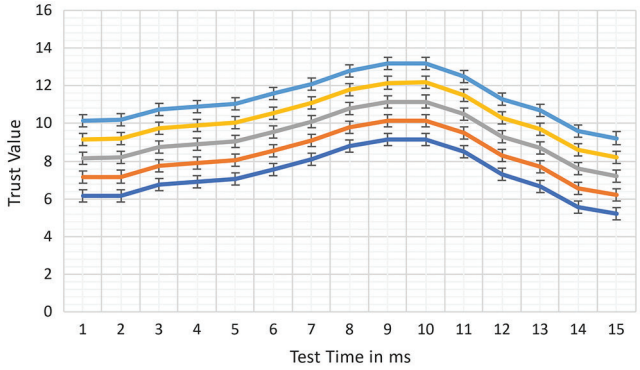

Figure 14: Analysis of malicious node *vs.* trust node evaluation with blockchain

The percentage of successful storage transactions processed in the Multichain network as a function of the size of the trust information file is exposed in Fig. 15. We calculated the percentage by comparing the number of successfully processed transactions to the number of established transactions. The results show that for transaction validation, block generation, and integration within the blockchain network, a 1 GB trust file is mandatory, and that the process takes 750 ms and consumes 40% of CPU and 4 of RAM, and that the process consumes 35.56% of CPU and 4 GB of RAM, demonstrating and proving its, d deployability, reliability, and feasibility in IoT environments. Our proposal outperforms the Multichain Blockchain in terms of successful storage transactions; the Multichain Blockchain's average throughput is 98.25%.

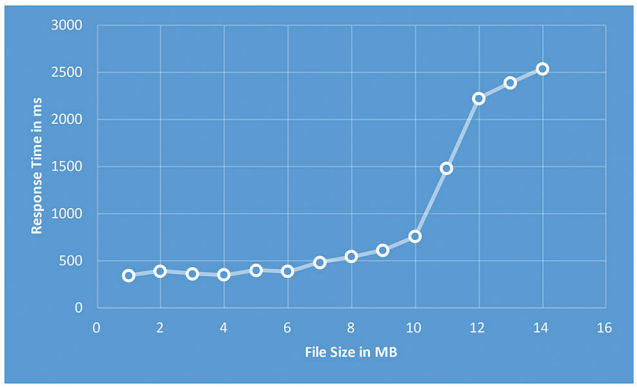

Figure 15: Average response time *vs.* file size with multichain blockchain's

5 Conclusion and Future Work

Blockchain is a trending topic in the IT world right now, and it's not only technically interesting, but it's additionally appealing in a commercial point of view. The Cyber IoT concept is based on connecting sensors, smart devices, Personal Computer (PC) and people, smart buildings and cities, electrical and water grids, and smart factories, to name a few. Security remains one of the most pressing concerns in CPSs. Cyber-attacks and threats increase as development, smart cities, and a globally connected system are enabled by the advanced Internet. Because network architectures and engineering systems are becoming more involute, and decentralized, t-less network architectures that perform a critical role in the advancement of CPSs are required. Blockchain technology is a promising field for enabling such advancements and realizing future-proof security solutions. This research work investigated different CPS security threats and proposed blockchain-predicated security procedures for implementation. We understood that this paper focuses on designing and implementing an agreement management system for private data. Furthermore, by appraising the significance of data collection, technology upgrades the current agreement process with Blockchain. Our proposal outperforms the Multichain Blockchain in terms of successful storage transactions in which the achieved average throughput is 98.15%.

We offer three benefits that are in line with the concept of the open model: disruptive changes, no margin costs, and no lock-in effect. However, the technological viewpoint is akin to hybrid identity in terms of interoperability. We can imagine utilizing any type of digital identity application to access these services, with no limits on the platform. Furthermore, from a business viewpoint, open Blockchain is more likely to be opportune in this case.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. M. Mu ller, D. Kiel and K. I. Voigt, "What drives the implementation of industry 4. 0? the role of opportunities and challenges in the context of sustainability," *Sustainability*, vol. 10, no. 1, pp. 247, 2018.

[2] Y. G. Srivastava, R. M. Parizi, A. Dehghantanha, K. K. R. Choo and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2146–2156, 2020.

[3] D. Dwivedi, G. Srivastava, S. Dhar and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, pp. 1–17, 2019.

[4] R. Guo, H. Shi, Q. Zhao and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11 676–11 686, 2018.

[5] S. T. Zargar, J. Joshi and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOs) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[6] V. A. K. Das, N. Kumar and M. Alazab, "Smart secure sensing for IoT-based agriculture: Blockchain perspective," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17591–17607, 2020.

[7] J. Grecuccio, E. Giusto, F. Fiori and M. Rebaudengo, "Combining blockchain and IoT: Food-chain traceability and beyond," *Energies*, vol. 13, no. 15, pp. 3820, 2020.

[8] S. Z. Husain, M. Abououf, M. Alblooshi and K. Salah, "Monetization of IoT data using smart contracts," *IET Networks*, vol. 8, no. 1, pp. 32–37, 2018.

[9] S. Zhou, H. Huang, W. Chen, P. Zhou, Z. Zheng *et al.,* "Pirate: A blockchain-based secure framework of distributed machine learning in 5G networks," *IEEE Network*, vol. 34, no. 6, pp. 84–91, 2020.

[10] Y. Chen, L. Su and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[11] G. T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 101–128, 2018.

[12] V. Sharma, I. You, F. Palmieri, D. N. K. Jayakody and J. Li, "Secure and energy-efficient handover in fog networks using blockchain-based DMM," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 22–31, 2018.

[13] H. Wang, L. Wang, Z. Zhou, X. Tao, G. Pau *et al.,* "Blockchain-based resource allocation model in fog computing," *Applied Sciences*, vol. 9, no. 24, pp. 5538, 2019.

[14] X. Li, J. Ma, W. Wang, Y. Xiong and J. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environment," *Mathematical and Computer Modelling*, vol. 58, no. 12, pp. 85–95, 2013.

[15] M. Haghi, K. Thurow and R. Stoll, "Wearable devices in medical internet of things: Scientific research and commercially available devices," *Healthcare Informatics Research*, vol. 23, no. 1, pp. 4–15, 2017.

[16] W. Liang, W. Huang, J. Long, K. C. Li and D. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[17] N. Elisa, L. Yang, F. Chao and Y. Cao, "A framework of blockchain-based secure and privacy-preserving E-government system," *Wireless Networks*, vol. 2018, pp. 1–11, 2018.

[18] P. Zhang, H. Wang, C. Hu and C. Lin, "On denial of service attacks in software defined networks," *IEEE Network*, vol. 30, no. 6, pp. 28–33, 2016.

[19] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan *et al.,* "Charm: A framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[20] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang *et al.,* "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.