Journal of Engineering, Project, and Production Management
2026, 16(7), 2026-0014

Implementation of a Secured Smart Home System Using Consortium Blockchain on Raspberry Pi as an Edge Gateway

Aurangjeb Khan[1] and K. Jayasudha[2]
[1] Assistant Professor, CMR University, Bangalore, India. and Research Scholar at Thiruvalluvar Govt. Arts College, Rasipuram, Tamil Nadu, India, E-mail: aurangjeb.khan83@gmail.com (corresponding author).
[2] Assistant Professor, Dept. of Computer Science, Thiruvalluvar Govt. Arts College, Rasipuram, Tamil Nadu, India, E-mail: jayasudhakaliannan@gmail.com

Engineering Management
Received April 19, 2026; revised June 17, 2026; accepted August 17, 2026
Available online August 30, 2026

Abstract: Smart homes are among the fastest-growing applications of the Internet of Things (IoT), enabling the automation and remote control of household devices such as door locks, lighting systems, fans, and security cameras. Ensuring the security of these interconnected devices is critical, as numerous global incidents have shown that traditional security mechanisms often fail to protect IoT systems from unauthorized access and cyberattacks. This paper presents a novel 4-layer architecture and the implementation of a blockchain-enabled, secure smart home system using ESP32 and Raspberry Pi nodes. A Consortium Blockchain is deployed on a Raspberry Pi to provide secure, tamper-resistant communication and device authentication within the smart home network. The proposed work also includes the system flowchart describing the blockchain workflow, 4-layer algorithms, and a methodology for implementing a consortium blockchain for smart home operations using high-end edge devices. Performance evaluation is conducted based on memory utilization, blockchain processing time, number of sensor nodes, and overall transaction latency. Experimental results demonstrate a system throughput of 51 Transactions Per Second (TPS) on a single sensor and single blockchain node, and an average response time of 1.292s with three additional sensor nodes, highlighting both good performance and sound security.

Keywords: Blockchain; consortium blockchain; internet of things (IoT); raspberry pi; smart home.

DOI 10.32738/JEPPM-2026-0014

1. Introduction

Revenue of Smart home in India is expected to show a Compound Annual Growth Rate (CAGR) from 2024 to 2029 of 8.59%, which is a projected market value of $9.8 billion by 2029, and globally the CAGR is expected to be 10.17% with a market value of $250.6 billion (Statista, 2022). According to statistics, the rapid growth of smart homes is evident, but security concerns are also growing with the number of new devices and network integration in smart home systems (Arif et al., 2020). IoT systems are prone to security attacks for many reasons. It is easy to access IoT devices, as they are usually low-processing-power, isolated and connected to a wireless network. According to Paloalto (2024), the average cost of an IoT security breach in 2023 was $9.5 million. These losses include legal fees, damage control, and more. The most common attacks on IoT systems are categorized as follows: 43% loss of sensitive data, 31% reputational damage, 17% loss of intellectual property, and 14% operational downtime. The most secure and tested technology for securing IoT devices is the Blockchain (Khan 2022). This article proposes a responsive and secure IoT-based smart home system with a 4-layer architecture that uses a consortium blockchain on Raspberry Pi nodes, making it more secure and efficient than existing smart home security systems, which are mostly implemented using cloud and ESP32 nodes (Arif et al., 2020). A cloud-based system with an ESP32, a low-powered device, is a completely internet-dependent system for blockchain operations. It consumes more power and requires a dedicated internet connection at all times. Raspberry Pi can perform most of the computation at the edge, hence limited to the internet. Consortium blockchain is a lightweight, semi-private blockchain

suitable for smart home, smart office, and other private security applications. Smart homes are primarily about automation, energy efficiency, customer experience, comfort and convenience, but a user can enjoy all of these only if they are free from threats. This article presents the implementation of a smart home system that uses a Raspberry Pi as an edge device and integrates with a Consortium Blockchain to protect the environment from various security threats. It introduces a novel system architecture, a detailed blockchain process flow diagram, and the corresponding implementation algorithms. Finally, the article provides a comprehensive evaluation and analysis of the system using real-time data, focusing on key parameters such as memory usage and command execution time. The article presents a novel methodology for securing a smart home using a consortium blockchain with IoT-based edge gateways, as presented in Section 3.

Blockchain technology is widely used in the cryptocurrency industry and has grown in popularity due to its security features. It increases system reliability and privacy due to decentralized network features. It is also known as distributed ledger technology (Arif et al., 2020; Khan, 2022). Blockchain technology's reliability and security features are being used across various project implementations to secure networks. By integrating this technology into our system, we can enhance the security. It works with the 3 steps: i) Maintaining data Integrity and storing cryptographic hashes of data on the blockchain enables the detection of unauthorized modifications and ensures tamper-proof data. ii) Decentralized authentication, in which each device has a unique blockchain ID, ensures robust security. iii) A smart contract system that ensures that nodes in the system can be accessed by only authorized persons. Ucam is one of the first blockchain security-based cameras (IoTeX Community, 2020). Blockchain is generally classified into 4 categories: public, private, consortium and hybrid blockchain, which are described and presented in the paper (Arif et al., 2020). and the process of implementing blockchain-based smart cameras using Ethereum on Raspberry Pi is presented in the CRC book chapter (Khan, 2022).

Consortium blockchain is a semi-decentralized, permission blockchain that serves as a middle layer between public and private blockchains by permitting only authorized members to validate transactions. Permissioned access, semi-decentralized governance, collaboration, a hybrid approach, and shared control are the key characteristics of the consortium blockchain. It uses various consensus mechanisms, such as Proof of Authority (PoA) and Practical Byzantine Fault Tolerance (PBFT), which are better suited to trusted, smaller groups and offer energy efficiency. The proposed model uses the Hyperledger Fabric framework to set up the consortium blockchain on the Raspberry Pi. It is an enterprise-grade blockchain framework well-suited for building private, permissioned networks with high security. It is ideal for the consortium blockchain (Singh et al. 2019; Perera et al., 2020).

IoT is a network of sensors and actuators, integrated with other technologies, that turns an ordinary system into a smart system by enabling devices to communicate without user input. Smart meters and smart cars are common examples of IoT products. It is used in smart homes, through constrained networks and devices, for basic operations of home appliances (Khan, 2022). IEEE 802.11-based Wi-Fi is used on edge devices for image and voice processing. It is a technology that provides a platform for multiple technologies, such as AI/ML, Natural language processing, image processing, and data analytics, to work together to create smart systems across almost all fields. IoT is the base technology for this smart home implementation and security measure using a consortium blockchain (Arif et al., 2020) on a Raspberry Pi board.

Many studies propose various blockchain types to secure smart home and smart system networks. The research conducted by Arif et al. (2020) investigated the use of a consortium blockchain to secure a smart home using ESP32 edge devices. Khan (2022) has presented the implementation of smart-city camera security using the Ethereum blockchain on a Raspberry Pi board. Singh et al. (2019) present blockchain- and cloud-based security for the smart home; however, the implementation is not highlighted. Baucas et al. (2021) proposed an IoT-based smart device control system using a private blockchain; however, they present only the abstract architecture. Zhang et al. (2025) present a blockchain architecture for securing the IoT network and also present an implementation using the L610-CN and ML307A communication modules. Qashlan et al. (2021) have presented the security analysis of blockchain for smart homes. Most research uses cloud-based blockchain, with limited implementation on edge devices. A smart home is also implemented on the Raspberry Pi without blockchain, using facial recognition and fingerprint-based security, as reported by Khan et al. (2025). Blockchain-based security ensures high reliability even in the resource-constrained IoT system (Zhou et al., 2022). An article by Khan et al. (2025) presents the implementation of smart home security using blockchain on VLSI, which is a powerful solution but adds additional cost. Pawar et al. (2025) proposed a blockchain-based IoT framework for smart home security, but it lacks real-time implementation. This article identifies a gap in secured smart home implementations, as prior work lacks edge-computation-based implementations using a private blockchain framework. Therefore, we propose a system model and methodology for edge computing using a consortium blockchain-based implementation with minimal dependence on cloud services.

This article presents novel architectures, a data flow diagram and methodologies for implementing a secure smart home system using a consortium blockchain on edge devices such as Raspberry Pi as blockchain nodes (Servers) and ESP32 as client nodes for sensor and actuator integration. It also presents the algorithms for all processes, from authentication using the fabric CA to ledger updates upon command execution. Finally, we evaluate the system based on the time taken by the different blockchain operations and the number of devices in the system.

The implementation of a public blockchain in smart home networks is not suitable due to scalability, energy consumption, and the open network. Private, consortium and hybrid blockchains are considered better for implementing a

smart home system; however, implementing these technologies is expensive, and cloud storage usage is maximized. Hence, our aim is to propose a model for implementing a low-cost, consortium blockchain-based solution for the smart home network security using Raspberry Pi nodes as blockchain server nodes with ESP32 edge devices as client systems with direct peer-to-peer communication with limited use of centralized cloud services to reduce the latency and internet dependency of the system.

2. Experimental Methods

2.1. Hyperledger Fabric

Hyperledger Fabric is a blockchain framework for consortium blockchains, enabling a group of permissioned or pre-approved organizations to share a network. It allows private transactions between the members of the consortium. Four steps of its working: a) Organizations need to agree on the set of policies to do the transaction in the consortium, b) Authorized members create a channel on the private network with a specific configuration. c) A transaction is initiated and sent to the approving peers. Once approved, the requested transaction is committed to the ledger. d) Channels maintain the privacy of the data and the transactions, even on the same network; other organizations will not be aware of the data shared in the channel. (Ferrag et al., 2021; Wang et al., 2023; IBM, 2025).

2.2. Setting Up Consortium Blockchain on Edge Devices (Raspberry Pi)

Materials: The proposed model required a Raspberry Pi B4 board as a Blockchain Node (BCN), with 8GB RAM, 128 GB SSD external drive attached to the Pi board with a bootable 64-bit Raspbian Operating System, stable LAN, or Wi-Fi network for smooth transactions, as presented by Deebak (2023) and VNC Viewer for remote access only during the setup and updates. The SSD can be upgraded to 8TB as required. Other devices, like ESP32s, are used as sensor nodes (SRNs), and relays control the actuators (Lights/Fans) running on 220 Volt power. Hardware integration is presented in Fig. 1.

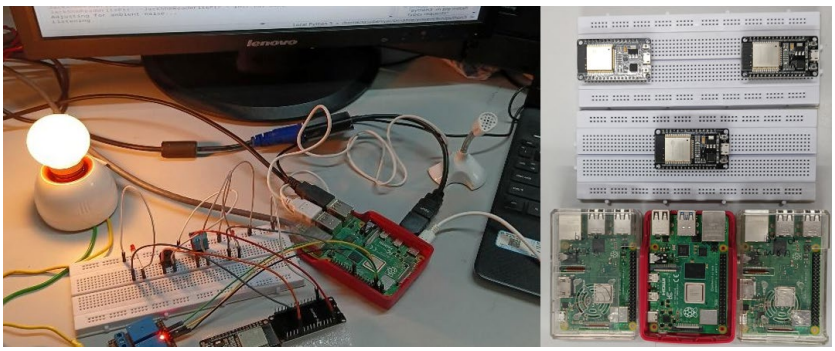

Fig. 1. Hardware setup with a Raspberry Pi board and ESP32 with sensors and actuators of a smart home

Commands to set up the Hyperledger Fabric framework for setting up the consortium blockchain on the Pi board. We need to install the dependency on the Pi board by executing the commands below (Hyperledger, 2020; Dockerdocs, 2025):

- sudo apt install -y docker.io docker-compose git curl build-essential
- sudo systemctl enable docker
- sudo systemctl start docker

Commands to install Hyperledger Fabric:

- git clone https://github.com/hyperledger/fabric-samples.git
- cd fabric-samples
- curl -sSL https://bit.ly/2ysbOFE | bash -s

To get all the required Fabric Docker images, such as Peer, Orderer, CA, etc., the network was setup by configuring /etc/hosts. The 3 Pi boards used in this research, acting as consortium members with IP addresses on the network, are configured as follows: 192.168.1.3 node1, 192.168.1.4 node2, 192.168.1.5 node3. To allow inter-node communication, we need to keep the ports open with the commands:

- sudo ufw allow 7050:7059/tcp

Launching the consortium blockchain on each Organization (Org):

- cd fabric-samples/test-network
- ./network.sh up createChannel -ca

For adding the nodes:

- ./network.sh addNode1

Each node (Node1, Node2) runs its own peer.
Commands for the verification of network:

- docker ps
- peer channel list
- geth attach http://localhost:8545

Command to deploy the smart contracts:

- Install & instantiate chaincode

Block Configuration parameters:

Our proposed model is based on 3 BCNs, where each block is configured to execute 20 commands with a 1 second latency for block confirmation. The data is simple, mostly commands for controlling home appliances. The transaction structure shows how a transaction is formed, verified and organized within the block. It includes the block header, transaction body, Merkle tree and the custom fields (Awan et al., 2025).

2.3.2. Key terms of the consortium blockchain built on Hyperledger Fabric

Org refers to consortium members having their own identity as Membership Service Providers (MSPs). MSP authenticates and authorizes participants using digital certificates. Certificate Authority (CA) issues and manages the certificates for users and peers. The peer node maintains the ledger and smart contracts (Chaincode). It can also endorse other peers and store ledger updates. Order Node is the ordering service node that collects endorsed transactions. Channel is a sub-network within the consortium (Jena et al., 2024). Each channel has its own ledger and members with access permission. A Smart Contract, or Chaincode, is the business logic that defines how a transaction modifies the ledger. It can be coded in Node.js, Java or Go. Ledger is the database maintained by peers, consisting of blockchain transaction logs. Endorsement Policy specifies which Org should endorse a transaction before it can be committed. The Fabric SDK contains libraries that enable the application to connect to the Fabric network to perform transactions and queries against the ledger.

3. Proposed Model

We propose a consortium blockchain-based smart home security system on edge devices with minimal reliance on the cloud, thereby ensuring complete security. We present a novel architecture in Fig. 2, a data flow diagram of the working process with the blockchain in Fig. 3, and four-level CBSH algorithms, which implement the system in Section 3.3.

3.1. Consortium Blockchain-Based Smart Home (CBSH) Architecture

The proposed model is implemented on the Consortium Blockchain using a novel 4-layer architecture, in which a web app layer is built on the server using the Fabric SDK and NodeJS for the user interface; a routes layer handles login credentials and registration; and a device control layer handles device control requests from users. The 3rd layer is the blockchain, which handles Hyperledger Fabric connections and network configuration, specifying the client app's connectivity to the blockchain. The device layer contains the general-purpose input/output (GPIO) and message queuing telemetry transport (MQTT) client applications for controlling and monitoring devices. This architecture provides a complete working model of the smart home system rather than merely highlighting the blockchain diagram and smart contracts. The consortium blockchain is preferred over private and hybrid blockchains due to its superior computational efficiency on Raspberry Pi nodes and its ability to facilitate secure collaboration among multiple participating entities.

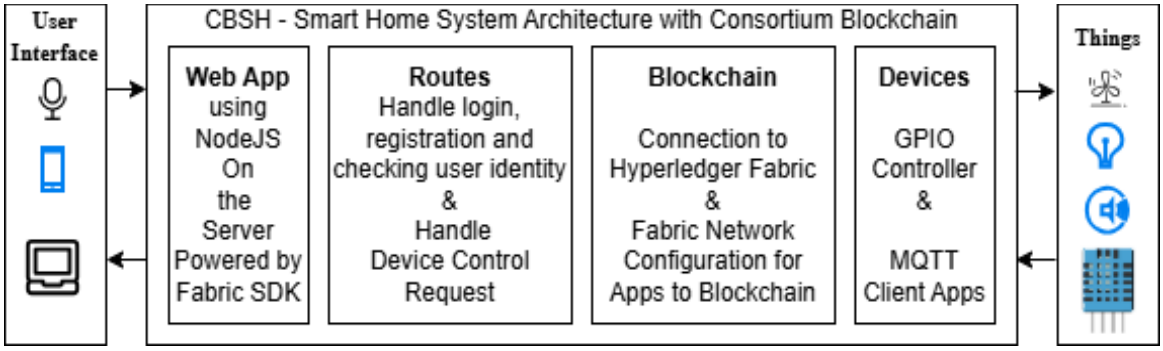

Fig. 2. Consortium blockchain-based 4-layer smart home system architecture

3.2. CBSH Data Flow Diagram of the Model

Fig.3 presents the Data Flow Diagram of the proposed architecture, outlining the complete process flow, including consortium blockchain operations, from user authentication through to smart home device control.

Fig. 3 presents the working mechanism of the model, from the user interface to command execution through the middle layer, which serves as a blockchain for validating the user's ID, validating blocks, executing commands and updating the ledger.

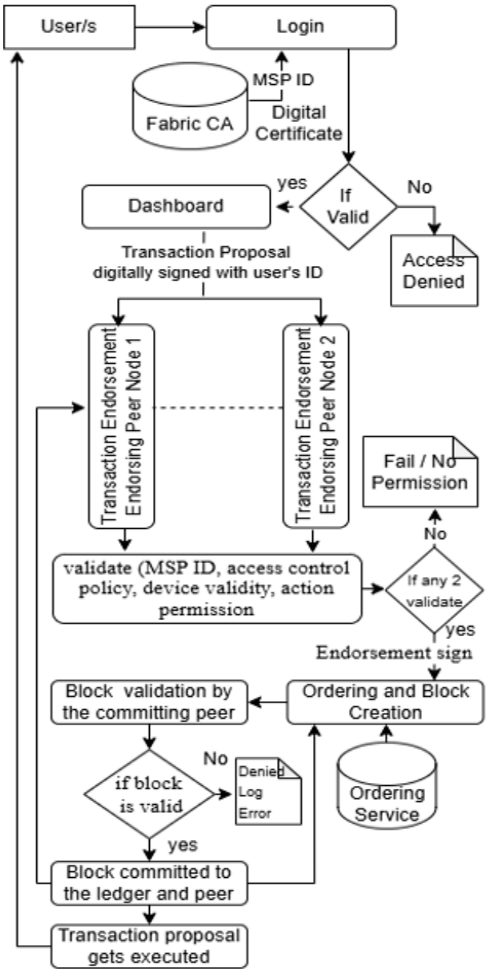

Fig. 3. Working process of proposed model from user authentication to IoT command execution

3.3. Proposed CBSH Algorithms for Model Implementation

The proposed model combined the strengths of the Consortium Blockchain and the edge device (Raspberry Pi) into a compact architecture and set of algorithms, with minimal dependence on cloud services, to ensure smooth functionality and robust security for the smart home. Algorithm 1 is implemented to verify users using the user ID, password, digital certificate (DC), and MSP ID. It also validates DC and MSP ID and produces the transaction proposals. Algorithm 2 validates the transaction proposal and produces the endorsement signature. Algorithm 3 sorts the transactions, groups the transactions into blocks and broadcasts the blocks to the peers. Algorithm 4 commits blocks to the ledger and executes the commands. All these algorithms are executed sequentially to complete the system's life cycle. The Proof of Authority (PoA) consensus algorithm is used in the implementation to maintain privacy and reduce system latency.

Algorithm 1: CBSH login authentication and command processing

//*Input:* User name and password
//*Output:* Dashboard with access control
1: Login Validation (UID, Password)
2: Verify the credentials with Fabric CA
3: if (valid) then:
4: get User (DC & MSP ID);
5: enable dashboard (device access control);
6: Wait for the user to choose a device control options
7: Prepare a transaction proposal using Fabric SDK
8: Digitally sign the transaction using the user's MSP ID and the DC.
9: Send the transaction proposal to endorsing Peer
10: else: display ("Access Denied") close the session

Algorithm 2: CBSH transactions endorsement

```
//Input: Transaction Proposals
//Output: Endorsement signature
1: Each endorsing peer receives the proposal.
2: check=validate (MSP ID, access control policy, device validity, action permission)
3: if (check) then
       Peer returns an endorsement signature.
4: else
       return fail to the client
```

Algorithm 3: CBSH ordering and block creation

```
//Input: Endorsement signature
//Output: Grouping & broadcasting blocks to peers
1: transactions=endorsements (node1)
2: invoking Ordering Services:
3:    sorting(transactions);
4:    grouping_Blocks (transactions);
5:    broadcast_To_Peer (blocks);
```

Algorithm 4: CBSH block validation, updating to ledger and the approved command execution

```
//Input: blocks, endorsed signature and read/write sets generated during simulation.
//Output: committing blocks to the ledger, execution of the user's commands
1: Peer nodes validate the received block:
       validate (endorsement signature, read/write sets)
2: If (validate) then Commit block to the ledger
       Update (world state "Command": "status")
3: If (invalid) then discard the transaction and log error.
4: Post-commit the transaction; app reads the new world state.
5: Executing device control action by GPIO or MQTT.
6: SaveToLedger (transaction, UID, Timestamp, Device ID, Action
```

3.4. Block Diagram of Software Stack to Secure Smart Home System With Additional Configuration

Additional configuration with system settings is required to harden the security by adopting the best practices as presented in Fig. 4 with a four-layer system security mechanism, along with the blockchain implementation using the novel architecture presented in Fig. 2, while Fig. 4 presents the security configuration on the model at each layer to secure the entire system stack from the applications layer to the device layer (Hanif et al., 2025; Shyika, 2024).

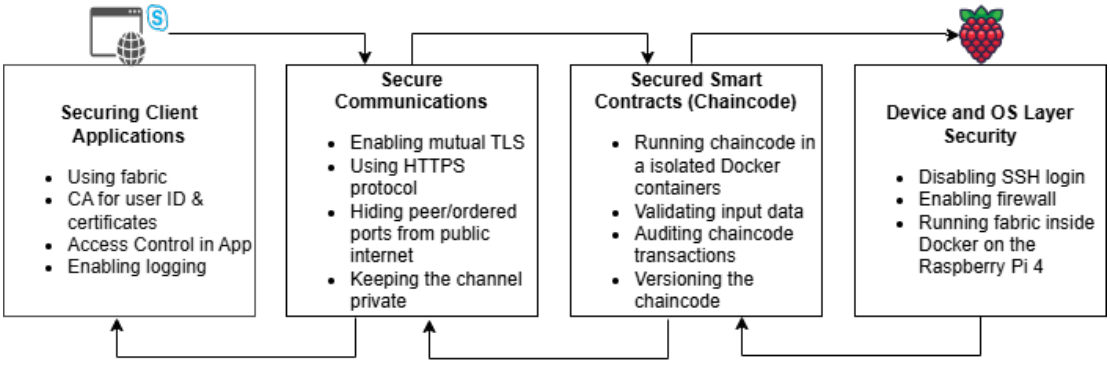

Fig. 4. Block diagram highlighting the 4-layer security mechanism for the entire system stack

4. Results and Discussion

Three Performance metrics are used here to evaluate the model: the first is the system's space complexity for all blockchain functions running on the Pi gateway. The second is based on the system's time complexity, and the third is based on scaling the number of sensors and blockchain nodes, as presented in the sections below. The performance evaluation is conducted on a well-defined benchmarking setup with Pi nodes, software dependencies and a Wireless Local Area Network (WLAN), as shown in Fig. 4, with the key performance metrics described here.

4.1. Memory Usage and Time Taken by Various Operations

Table 1 presents the average memory utilized by the blockchain operations in the model, while Table 2 presents the average time taken by the simple and complete operations in the model on the Raspberry Pi in a real-time environment, executing Python code on the Raspbian platform.

Table 1. Average memory usage by blockchain operations

System operations	Average memory usage
Fabric peer	900MB RAM
Orderer node	500MB RAM
CouchDB	800MB RAM
Chaincode container	250MB RAM
Fabric CA	260MB RAM
Docker daemon and OS overhead	400 MB RAM

Table 2. Average execution time of blockchain process

System operations	Average time
Simple read query **(**Chaincode query)	100ms
Submit transaction (includes Endorse + Order + Commit)	500ms.

4.2. Performance Evaluation Based on Memory Usage and Execution Time

Let us assume that Transactions Per Second (TPS) is the system throughput, total transaction latency is T_total, endorsement time is T_endorse, ordering time is T_order, and ledger commit time is T_commit. M_total is the total system memory =8192 MB for the 8 GB Pi B4, and the current consumed memory(M_used)=3110 MB. M_free= (M_total − M_used) and α is the memory impact factor.

$$M_{free} = M_{total} - M_{used} = 8192 - 3110 = 5082 \text{ MB} \quad (1)$$

Eq. (1) represents free memory; here, throughput is inversely proportional to processing time and directly proportional to free memory. Hence, the system throughput is:

$$TSP = \alpha * \frac{Mfree}{T\_endorse+T\_order+T\_commit} \quad (2)$$

Where: $T_{\_total} = T_{\_endorse} + T_{\_order} + T_{\_commit}$

T_{total}=120+200+180=500 ms=0.5 s

Hence the $TSP = \alpha * \frac{5082}{0.5} = \alpha * 10164$

Eq. (2) refers to the TSP based on the free memory and blockchain process time, where α is the empirical constant from benchmarking (usually between 0.003–0.015 on Pi B4 depending on the load). On the proposed model, the average value of α=0.005, then TPS=0.005 * 10164 => 50.82 =51 TPS.

The proposed model supports 51 TPS, meaning we can switch devices like lights and fans on/off 51 times per second with minimal load, using only one client device integrated with electrical or electronic devices and one blockchain server. System performance is presented in Table 3 for a greater number of client devices or SRNs under high load with 3 blockchain nodes (BCN). Fig. 5 presents the system performance based on Table 2 and the TSP outcome.

4.3. Model Performance, Based on Number of Devices and Response Time- Scalability Analysis

As presented in section 2.2, we have 3 sensor nodes (SRN) for controlling the devices and 3 blockchain nodes (BCN) for handling security and system setup. Performance equation Eq. (3) of the model with total transaction time is:

$$P = T\_total * (\log_2(BCN) + SRN / BCN) \quad (3)$$

$$P = 500ms * (\log_2(3) + 3/3) = 1292ms.$$

Where $\log_2(BCN)$ is the blockchain consensus time, and SRN/BCN is the sensors/actuators-to-blockchain-node communication overhead. Here, one SRN can integrate with up to 9 client devices (Espressif, 2020). The result is presented here based on three SRNs and three BCN nodes to handle the complete smart home network load and prevent single-point failure. The SRN can be increased as per the number of devices required in a smart home. Fig. 5 shows the analysis of the model up to 9 SRNs with 3 BCNs.

Fig. 6 presents the performance graph for the model as a function of the number of devices, highlighting scalability and total transaction latency. The proposed model comparisons with existing work in Table 4 highlight this article's achievement.

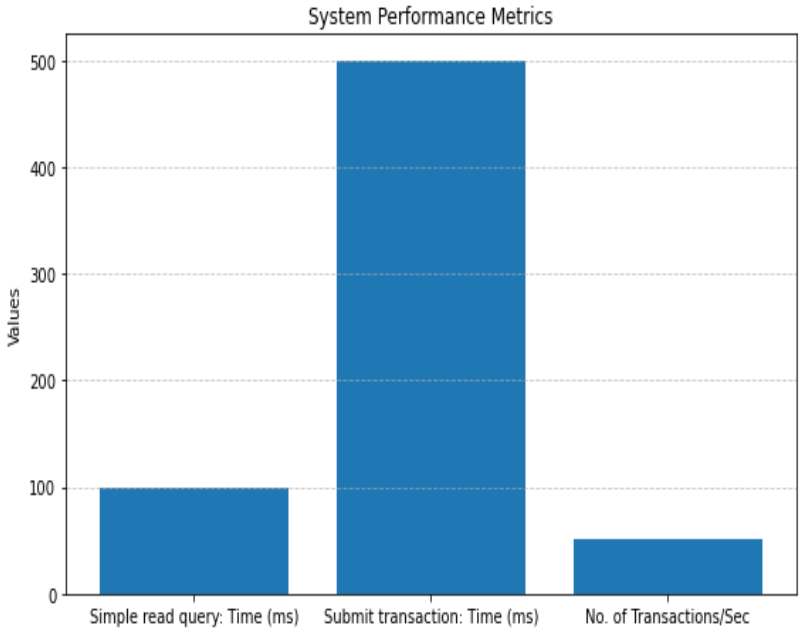

Fig. 5. System performance chart with simple read query, submit and number of transactions per second.

Table 3. Model performance versus number of sensor nodes (SRN)

No. of sensor nodes	Total response time
3 (27 devices in the network)	1.292 Seconds
6 (54 devices in the network)	1.792 Seconds
9 (81 devices in the network)	2.292 Seconds

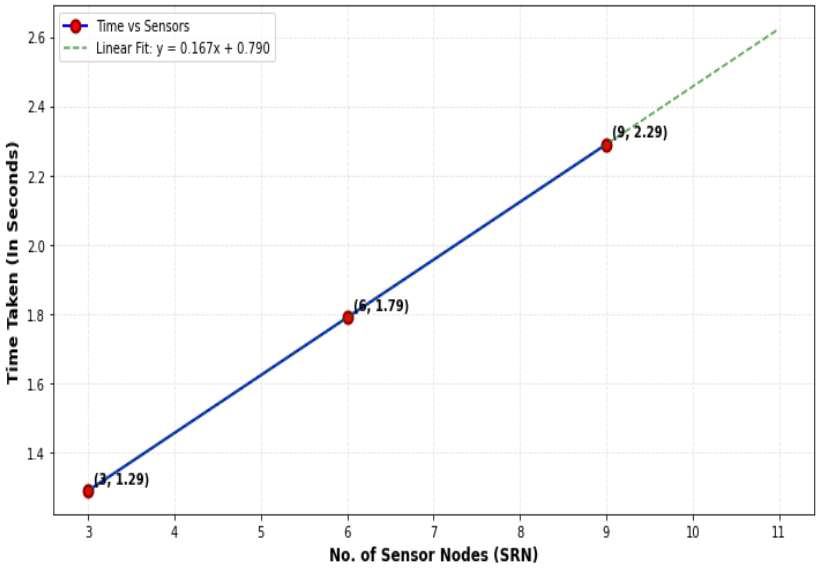

Fig. 6. Number of sensor nodes vs total response time

4.4. Privacy and Security Model Analysis

A smart home's automation system faces multiple security threats. Our effort is to minimize the security threat in the system. Table 5 presents the potential risks and measures taken to mitigate them by implementing consortium blockchain-based models on edge devices, using the proposed architecture and algorithms, with a minimum of 3 SRNs and 3 BCN nodes. Since all the transactions are digitally signed, time-stamped, and stored immutably, once recorded, entries cannot be modified due to the blockchain's hash chain: Hash n = H(Transaction $_n$ + Hash $_{n-1}$) (Adla et al. 2024). Any update changes the entire blockchain; hence, the system ensures accountability through a cryptographic, distributed ledger. Table 5 presents the security analysis of the model as per the system implementation.

Table 4. Key performance comparisons with existing model

Performance metric	Proposed model	Existing system (Arif et al., 2020)	Improvement
Method	Edge computing	Cloud computing	Reduced latency, reduced bandwidth usage and better security.
Response time	1.292 seconds	3 seconds	The proposed system is 57% faster compared to the existing system.
Throughput	51 task per second	0.6 tasks per second	Exceptional improvement was recorded in the proposed system
Latency variation	± 56ms	± 400ms	Proposed model is stable with the load
Data Privacy	High	Moderate	Sensitive data is not transmitted externally

Table 5. Potential risks and measures taken to mitigate risks

Potential risks	Measures taken to mitigate the risks in the model	Residual risk
Unauthorized access	Unauthorized commands are rejected by the blockchain's endorsement and the policy rules.	Very low
Single point of failure	Ledger and logic are distributed across multiple peers.	Low
Data tampering	Every transaction is recorded in the ledger; it cannot be deleted or modified.	Negligible
Malicious admins	Endorsement policies require multiple node approvals before changes are valid.	Negligible
Device spoofing	Devices with valid certificates issued by Fabric CA can only join or be controlled.	Negligible
Replay attacks	Fabric includes nonce and timestamp in each transaction; duplicates are rejected.	Negligible
Cloud dependency	Little cloud dependency, as all transaction logic and data remain on the edge.	Very low
Device manipulation	Blockchain verification ensures that device commands execute only if the ledger confirms the transaction's authenticity.	Negligible
Accountability	Immutable blockchain records every operation with a timestamp and a signature.	Negligible
Denial of Service (DoS)	Lightweight block parameters and rate limiting prevent DoS attacks.	Low

The model is tested with up to 3 BCN nodes and 9 SRNs to make it cost-effective for a smart home, but the number of BCN nodes can be increased in the network as required to improve system reliability.

5. Conclusion

Securing the IoT system is a high priority in implementing a smart system to ensure reliability and dependability. This article investigates recent work on securing smart homes and other smart systems using blockchain technologies. Taking a leap from the previous work, we implement and propose a responsive and secure smart home system using a consortium blockchain with an excellent throughput of 51 transactions per second, and a 1.292 seconds response time on the edge devices. This article proposes the implementation of the complete consortium blockchain using the Hyperledger Fabric framework on the Raspberry Pi node, along with the user interface application for controlling and monitoring the devices, with little dependency on cloud services. The ESP32 is used as a sensor node (SRN) to integrate sensors and actuators. It interacts with the Pi Blockchain Nodes (BCN) as a server using a Wi-Fi network. This article presents a novel architecture for the system, a blockchain process flow diagram, and Consortium Blockchain-based Smart Home (CBSH) algorithms for implementation. Furthermore, we evaluate the system based on memory utilization and average time taken by the consortium blockchain processes. We also evaluate the system based on the number of sensor nodes and the total transaction latency. Our system demonstrates better performance compared to the existing model, as presented in Table 4. The system demonstrates remarkable throughput, processing 51 instructions per second on a single SRN with one BCN, and achieves a final system performance of 1.292 seconds with 3 SRNs and 3 BCNs. Evaluation is computed with the pattern up to nine SRNs and with three BCNs, as presented in Table 3. Further research will be conducted to investigate the performance of the system with different types of loads on the system, such as a voice command-based user interface application in the smart home network.

Acknowledgments

The authors would like to thank their institutions for providing the infrastructure and technical resources necessary for conducting this research.

Author Contributions

Aurangjeb Khan contributed to the literature review, methodology development, system implementation, manuscript preparation, and editing. K. Jayasudha supervised the research, reviewed the manuscript, provided intellectual input, and approved the final manuscript for submission.

Funding

No funding was received for this work.

Institutional Review Board Statement

This study did not involve human participants or the collection of personal or sensitive data.

Declaration of Artificial Intelligence (AI) Tools

Authors declare that AI tools were used solely to assist with language refinement and grammar checking. All scientific content, research design, analysis, interpretation of results, and conclusions were conceived, verified, and approved by the authors, who take full responsibility for the content of this manuscript.

References

Arif, S., Khan, M. A., Rehman, S. U., Kabir, M. A., and Imran, M. (2020). Investigating Smart Home Security: Is Blockchain The Answer? *IEEE Access,* doi:10.1109/access.2020.3004662.

Awan, S. A., Khattak, M. A. K., and Sathio, A. A. (2025). Dynamic strategy for adaptive block size optimization in blockchain technology. *Discov Sustain* 6, 849. https://doi.org/10.1007/s43621-025-01749-x

Baucas, M. J., Gadsden, S. A., and Spachos, P. (2021). IoT-Based Smart Home Device Monitor Using Private Blockchain Technology and Localization. *IEEE Networking Letters*, 3(2), 52-55, doi: 10.1109/LNET.2021.3070270.

Deebak, B. D. (2023). A Lightweight Blockchain-Based Remote Mutual Authentication for AI-Empowered IoT Sustainable Computing Systems. *IEEE Internet of Things Journal*, 10(8), 6652-6660, doi: 10.1109/JIOT.2022.3152546.

Dockerdocs. (2025). Install Docker Engine on Raspberry Pi OS. Retrieved from https://docs.docker.com/engine/install/raspberry-pi-os/ on 20th Oct 2025.

Espressif. (2020). Software Framework-Bluetooth. Retrieved from https://docs.espressif.com/projects/esp-faq/en/latest/software-framework/bt/ble.html on 3rd Nov 2025.

Ferrag, M. A., and Shu, L. (2021). The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial. *IEEE Internet of Things Journal*, 8(24), 17236-17260, doi: 10.1109/JIOT.2021.3078072.

Hanif, M., Munir, E. U., Rehan, M. M., Ahmad, S. G., Khan, I., and Setchi, R. (2025). Tiered blockchain framework: A secure, trustworthy, and cost-efficient solution for the digital rights protection. *Blockchain: Research and Applications,* 6(4), https://doi.org/10.1016/j.bcra.2025.100308.

Hyperledger Fabric Docs. (2020). Install Fabric. Retrieved from https://hyperledger-fabric.readthedocs.io/en/latest/install.html on 20th Oct 2025.

IBM. (2025). What is Hyperledger Fabric. Retrieved from https://www.ibm.com/think/topics/hyperledger on 20th Oct 2025.

IoTeX Community. (2020). Ucam. Retrieved from https://community.iotex.io/t/ucam-the-first-blockchain-powered-security-camera/664 on 28th Dec 2024.

Jena, S. K., Kumar, B., Mohanty, B., Singhal, A., and Barik, R. C. (2024). An advanced blockchain-based hyperledger fabric solution for tracing fraudulent claims in the healthcare industry. ISSN 2772-6622, https://doi.org/10.1016.

Khan, A. and Jayanthi, M. (2022). Blockchain for securing the IoT-based smart wireless surveillance cameras, Blockchain for Industry 4.0, CRC Press. DOI: 10.1201/9781003282914-10.

Khan, A., Jayasudha, K., and Jagadeesan, M. (2024). Facial Recognition and Fingerprint-Based Authentication System for a Smart Home Security Using IoT. Springer, Singapore. https://doi.org/10.1007/978-981-97-6103-6_20.

Khan, A., Jayasudha, K., and Jagadeesan, M. (2025). Privacy-Preserving Access Control for IoT Smart Homes Using Hyperledger Fabric Consortium Blockchain and Edge Computing on Raspberry Pi. *Journal of VLSI Circuits and Systems*, *7*(2), 37–42. https://doi.org/10.31838/JCVS/07.02.05.

Padma, A., and Ramaiah, M. (2024). Blockchain based solution for secure information sharing in pharma supply chain management, Heliyon, 10(22), e40273, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2024.e40273.

Paloalto. (2024). The 2024 Benchmark Report on IoT Security. Retrieved from https://www.paloaltonetworks.com/resources/research/the-2024-benchmark-report-on-iot-security on 7th Jan 2025.

Pawar, P. P., Kumar, D., Addula, S. R., Cheema, Q. N., Haq, A. U., and Meesala, M. K. (2025). A Blockchain-Based IoT Framework for Smart Homes: Enhancing Energy Prediction and Security with LSTM and Equilibrium Optimization. 10.1109/icoicc64033.2025.11052010.

Perera, S., Nanayakkara, S., Rodrigo, M. N. N., Senaratne, S., and Weinand, R. (2020). Blockchain technology: Is it hype or real in the construction industry? Journal of Industrial Information Integration, 17, ISSN 2452-414X, https://doi.org/10.1016/j.jii.2020.100125.

Qashlan, A., Nanda, P., He, X., and Mohanty, M. (2021). Privacy-Preserving Mechanism in Smart Home Using Blockchain. *IEEE Access*, 9, 103651-103669, doi: 10.1109/ACCESS.2021.3098795.

Singh, S., Ra, I. H., Meng, W., Kaur, M., and Cho, G. H. (2019). SH-BlockCC: A secure and efficient Internet of Things smart home architecture based on cloud computing and blockchain technology. *International Journal of Distributed Sensor Networks*, 15(4), 118.

Shyika, A. (2024). Building a 2025 Security Stack: A Veteran CISO's Guide to Cost-Effective Priorities. Retrieved from https://underdefense.com/blog/security-stack-guide/ on 5th Dec 2025.

Statista. (2022). Smart Home – India. Retrieved from https://www.statista.com/outlook/cmo/smart-home/india on 28th Dec 2024.

Wang, M., Zhu, T., Zuo, X., Ye, D., Yu, S., and Zhou, W. (2023). Blockchain-Empowered Multiagent Systems: Advancing IoT Security and Transaction Efficiency. *IEEE Internet of Things Journal*, 11(7), 11217-11231, doi: 10.1109/JIOT.2023.3329961.

Zhang, L., Hang, L., Zu, K., Wang, Y., and Yang, K. (2025). Constructing Next-Generation IoT Security: Embedded Smart Contracts and Multilayer Security Protection. *IEEE Internet of Things Journal*, 12(12), 19716-19731, doi: 10.1109/JIOT.2025.3541256.

Zhou, J., Feng, G., and Wang, Y. (2022). Optimal Deployment Mechanism of Blockchain in Resource-Constrained IoT Systems. *IEEE Internet of Things Journal*, 9(11), 8168-8177, doi: 10.1109/JIOT.2021.3106355.

Aurangjeb Khan is currently serving as an Assistant Professor at CMR University and is pursuing his doctoral research at Thiruvalluvar Government Arts College, which is affiliated with Periyar University. His research interests encompass Internet of Things (IoT), Blockchain Technology, Digital Twins, Human–Computer Interaction, and Algorithm Design. He has authored eight Scopus-indexed research publications and eleven textbooks covering diverse subjects, including IoT, Cyber Security, .NET Framework, and Mobile Application Development

K. Jayasudha is currently an Assistant Professor at the PG and Research Department of Computer Science at Thiruvalluvar Government Arts College, Rasipuram. She has over 21 years of teaching experience in computer science. Her academic contributions include numerous research publications in reputed national and international journals. As a dedicated research mentor, she has successfully guided several M. Phil. and Ph.D. scholars and is currently supervising three doctoral researchers in computer science.