ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/325-334
Molagara Sathwika *et. al.*, /International Journal of Engineering & Science Research

Full Length Research Article

End-To-End Security In Smart Homes Using A Consortium Blockchain Approach

Mr. Monis Tariq[1],Molagara Sathwika[2],Nimmala Harthik Goud[3],Ranabothu Vikas Reddy[4],Sarangi Bharath Sai[5]
[1]Assistant Professor ; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
[2,3,4]B.Tech Students; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
Mail Id; vr9705949@gmail.com
Accepted 22-03-2026

Abstract
The widespread integration of smart technologies across domains such as residential automation, urban infrastructure, healthcare, and transportation has significantly enhanced modern living standards. However, these interconnected environments introduce critical security concerns, including reliable device authentication, efficient key management for constrained devices, protection of sensitive data, and secure storage mechanisms. Inadequate safeguards in such systems can lead to unauthorized access, data leakage, and exploitation of user information.This paper proposes a blockchain-based security architecture specifically designed for smart home ecosystems. The framework is structured into two operational phases: the first phase focuses on device authentication and access control within the local smart environment, while the second phase ensures secure interaction with external service platforms through regulated access and immutable data handling. Blockchain technology is utilized to record device activities, transactions, and access events in a tamper-resistant manner using SHA-256-based cryptographic linking.To further strengthen security at the device level, the system incorporates computer vision techniques using OpenCV for real-time video monitoring and facial authentication. Additionally, encryption mechanisms are applied to protect multimedia streams and sensitive user data during transmission and storage. A functional prototype with a visual simulation of the smart home environment has been implemented to validate the approach.Experimental results indicate that the proposed framework effectively mitigates common security threats while achieving improved latency and throughput compared to existing solutions. The study demonstrates that integrating blockchain with lightweight cryptographic methods and intelligent monitoring can provide a robust end-to-end security solution for next-generation smart home systems.

Keywords: *Blockchain, Smart Home Security, Device Authentication, Access Control, SHA-256, OpenCV, Facial Recognition, Encryption, Internet of Things (IoT), Secure Data Storage.*

INTRODUCTION

Recent advancements in communication technologies such as fifth-generation (5G) networks, artificial intelligence (AI), and big data analytics have accelerated the growth of the Internet of Things (IoT). IoT enables the interconnection of physical devices equipped with sensors, actuators, and communication capabilities, allowing seamless data exchange and automation across diverse applications. As a result, IoT has expanded beyond industrial usage into domains such as smart cities, transportation, agriculture, healthcare, and energy management.

Among these applications, smart homes represent a critical component of modern smart city ecosystems. A smart home integrates multiple intelligent devices—including environmental sensors, surveillance systems, smart appliances, and healthcare monitors—to enhance convenience, efficiency, and quality of life. These devices collaborate through a network, often managed by a central gateway or fog node, which brings computational capabilities closer to end devices and reduces dependence on remote cloud infrastructure.

However, the continuous generation and transmission of large volumes of data from IoT devices introduce significant security and privacy concerns. Due to the distributed and heterogeneous nature of IoT systems, challenges such as secure device authentication, efficient key management, data confidentiality, and integrity remain unresolved. Unauthorized access to smart home networks can lead to surveillance, data leakage, or

malicious control of devices, posing serious risks to user safety and privacy.

Blockchain technology has emerged as a promising solution to address these challenges. Its decentralized, immutable, and tamper-resistant ledger enables secure transaction recording and transparent data management without relying on centralized authorities. Additionally, blockchain can facilitate secure key management and act as a distributed trust mechanism for resource-constrained IoT devices.

This work introduces a novel multi-chain blockchain framework that integrates private blockchain at the fog layer and consortium blockchain at the cloud layer. This hybrid approach enhances scalability, flexibility, and privacy while enabling efficient data processing and secure interoperability. Furthermore, the system incorporates intelligent monitoring through computer vision techniques, strengthening device-level authentication and overall system trust.

Scope of the Project

This project focuses on the design and implementation of a secure and scalable framework for smart home environments using blockchain technology. The proposed system addresses key challenges in IoT security, including authentication, authorization, key management, data confidentiality, secure storage, and privacy preservation.

Unlike conventional approaches, the framework combines blockchain's immutable ledger with lightweight cryptographic mechanisms to ensure secure communication and tamper-proof record keeping. It also supports secure interaction between smart home devices and external cloud services, preventing unauthorized access and data misuse.

Additionally, the integration of computer vision techniques using OpenCV enables real-time monitoring and facial authentication, enhancing trust and user-device interaction. Overall, the scope encompasses a unified architecture that provides end-to-end security for smart home ecosystems.

Objective

The primary objective of this project is to develop a robust and efficient security solution for smart home systems by integrating blockchain technology with lightweight cryptographic and intelligent monitoring techniques.

Specifically, the project aims to:

- Ensure secure authentication of devices and users
- Provide efficient key management suitable for resource-constrained IoT devices
- Protect sensitive data from unauthorized access and tampering
- Enable secure and immutable storage of transactions and activity logs
- Support privacy-preserving communication with cloud services
- Enhance user trust through video-based monitoring and biometric authentication

The ultimate goal is to deliver a reliable, scalable, and high-performance platform that ensures comprehensive security in real-world smart home deployments.

Problem Statement

The rapid expansion of IoT-based smart home systems has introduced significant security challenges due to their distributed architecture and heterogeneous device ecosystem. Key concerns include device authentication, access control, data privacy, confidentiality, and integrity.

Traditional cloud-centric solutions are insufficient for addressing these issues, as they introduce central points of failure and increase the risk of data breaches. Moreover, the large volume of data generated by IoT devices creates additional challenges in terms of storage, processing, and secure transmission.

While blockchain technology offers decentralized trust and tamper-resistant storage, existing single-chain blockchain solutions often suffer from limitations related to scalability, flexibility, and transaction efficiency.

Therefore, there is a need for a comprehensive security framework that leverages multi-layer blockchain integration across fog and cloud environments. Such a solution should provide secure authentication, efficient transaction management, improved scalability, and enhanced privacy protection for smart home systems.

PROJECT DESCRIPTION

The proposed system is designed as a multi-layered architecture that integrates dual blockchain networks to ensure secure communication, authentication, and data management in smart home environments. The framework is structured into three primary tiers: device layer, fog layer, and cloud layer.

At the device layer, resource-constrained IoT sensors continuously collect environmental and user-related data. These devices initiate secure registration with nearby fog nodes using temporary session credentials. The fog layer acts as an intermediate processing unit and maintains a private blockchain network. This blockchain serves as a decentralized trust mechanism, enabling secure identity management using Elliptic Curve Cryptography (ECC). Communication security is established through Elliptic Curve Diffie–Hellman (ECDH) for key agreement and Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication.

To reduce computational and storage overhead, transactions generated by devices are aggregated at

the fog nodes. Instead of storing individual records, only aggregated signatures are committed to the blockchain, improving efficiency without compromising security.

A user-facing smart home dashboard interfaces with both blockchain layers, enabling real-time monitoring, visualization, and secure interaction with devices. This hierarchical design achieves a balance between security, scalability, and performance.

Methodology

The proposed methodology combines blockchain technology with lightweight cryptographic mechanisms to address security challenges in IoT-based smart home systems.

Two types of blockchain models are considered:

- **Permissionless blockchain**, which allows open participation but may introduce latency and overhead
- **Permissioned blockchain**, which restricts access to authorized entities, improving efficiency and control

To achieve a balanced approach, a **consortium blockchain** model is adopted at the cloud layer, where multiple trusted entities collaboratively manage the network. This ensures decentralization while maintaining performance.

REQUIREMENTS ENGINEERING

The proposed system is developed and evaluated using a combination of simulation tools, blockchain platforms, and distributed storage technologies to accurately represent a smart home environment. For simulation purposes, CupCarbon is utilized, which is a Smart City and Internet of Things Wireless Sensor Network simulator. This tool provides an interactive platform for modeling communication behavior, node mobility, and environmental conditions. It also integrates OpenStreetMap, allowing realistic placement of sensor nodes within a geographical context. Additionally, CupCarbon includes SenScript, a scripting language that enables developers to configure and program individual sensor nodes, making it suitable for prototyping IoT-based smart home networks.

To implement the private blockchain layer, Hyperledger Fabric is adopted due to its permissioned nature and modular architecture. Developed under the Linux Foundation, Hyperledger Fabric supports smart contracts, also known as chaincode, and provides mechanisms for identity management and access control. The platform organizes participants into multiple entities known as organizations, each associated with a Certificate Authority responsible for issuing digital identities. Communication between these organizations is facilitated through channels, ensuring secure and isolated data exchange. Transactions are validated, ordered, and committed to the ledger using consensus protocols such as RAFT or Practical Byzantine Fault Tolerance, thereby ensuring data integrity and consistency across the network.

In addition to Hyperledger Fabric, a private Ethereum network is considered to experiment with decentralized configurations. This network operates independently of the public Ethereum blockchain and consists of multiple nodes configured with a unique network identifier. A lightweight Proof-of-Authority consensus mechanism, such as Clique, is employed to reduce computational overhead while maintaining trust among participating nodes. This setup is particularly useful for testing blockchain synchronization and decentralized application development in a controlled environment.

To manage the large volume of data generated by IoT devices, the InterPlanetary File System is integrated as a decentralized storage solution. Unlike traditional centralized storage systems, IPFS distributes data across multiple nodes and retrieves it using content-based addressing through a distributed hash table. This approach enhances data availability, integrity, and resilience. To ensure persistent storage and accessibility, a pinning service such as Pinata is used, which maintains data availability within the IPFS network.

DESIGN ENGINEERING

This section presents an informal security evaluation of the proposed framework, focusing on its resilience against major cyber threats.

1. Unauthorized Sensor Registration

The framework enforces an offline initialization phase where the network administrator pre-registers all IoT devices. During this stage, each device is assigned a unique identifier, a one-time cryptographic key, and a timestamp, which are securely recorded on the private blockchain. Because only pre-registered devices are recognized, any attempt by an unknown or unauthorized sensor to join the network is automatically rejected. This mechanism ensures a secure onboarding process and prevents unauthorized device inclusion.

2. Man-in-the-Middle Attack

In a Man-in-the-Middle (MitM) attack, an adversary intercepts and potentially modifies communication between two parties. The proposed system mitigates this threat by using symmetric encryption keys derived through the Elliptic Curve Diffie–Hellman (ECDH) protocol. Since the encryption key is not exposed during transmission, attackers cannot decrypt intercepted messages. Furthermore, all communications are authenticated using the Elliptic Curve Digital Signature Algorithm (ECDSA). Compromising message integrity would require access to private keys, which is computationally infeasible due to the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Therefore, the system demonstrates strong resistance to MitM attacks.

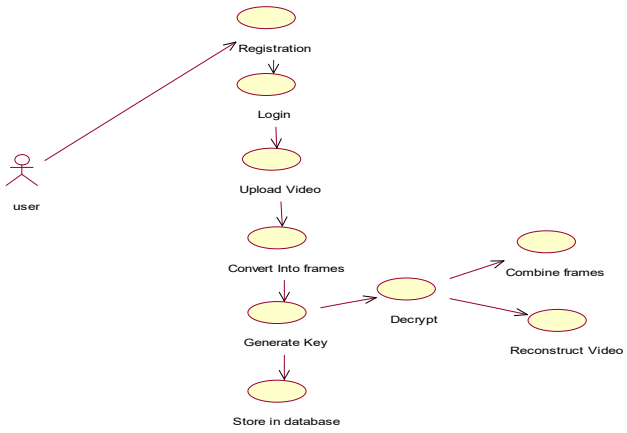
3. Distributed Denial-of-Service (DDoS) Attack

The blockchain network is permissioned and configured by the administrator, ensuring that only authenticated fog nodes and sensors participate. All legitimate entities are registered offline, and their identities are stored within the blockchain. Malicious activity from any registered node can be detected and revoked promptly. External adversaries are unable to overwhelm the system because unregistered requests are filtered out at the entry level. This controlled access significantly reduces the risk of DDoS attacks.

UML-Based System Representation

The system behavior and structure are modeled using standard UML diagrams, each serving a specific purpose:

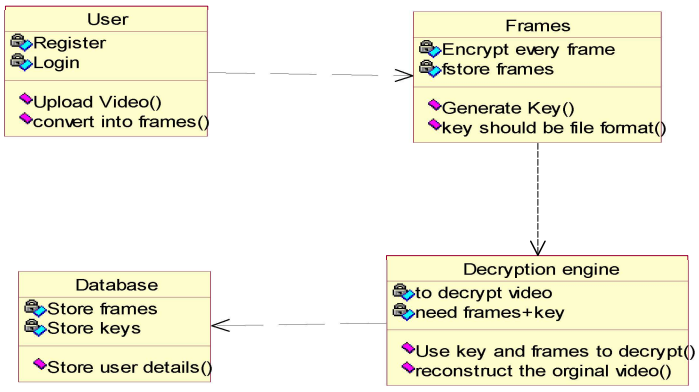
Use Case Diagram



Explanation:

The use case diagram illustrates interactions among users, administrators, and system components in the video encryption framework. Users upload videos, which are divided into frames, encrypted using AES, and distributed across storage nodes. The blockchain maintains encryption keys and access policies. During retrieval, user authentication and authorization are verified before decrypting and reconstructing the video. This diagram highlights system functionality and actor roles.

Class Diagram



Explanation:

The class diagram presents the static structure of the system by defining key classes such as *User*, *VideoProcessor*, *FrameManager*,

EncryptionEngine, *BlockchainHandler*, and *StorageNode*. Each class contains attributes (e.g., encryption keys, frame identifiers, metadata) and methods (e.g., encryptFrame(), storeChunk(), verifyAccess()). Relationships among classes demonstrate how encryption, storage, and blockchain operations are coordinated.

Object Diagram

Explanation:

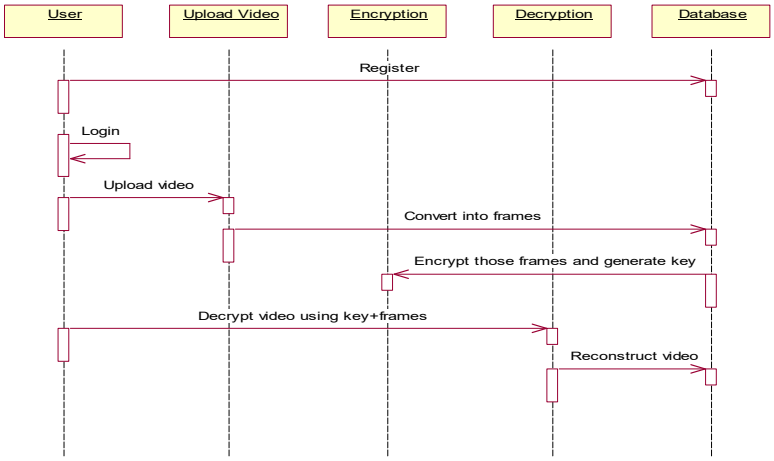
The object diagram represents a runtime instance of the system. A user object uploads a video, which is processed into frame objects. These frames are encrypted and linked to storage node instances. Simultaneously, blockchain records maintain metadata and cryptographic keys. During retrieval, decryption and reconstruction components interact with stored frames and blockchain records to regenerate the original video.

State Diagram

Explanation:

The state diagram models the lifecycle of a video file. The process transitions through states such as *Uploaded*, *Frame Extracted*, *Encrypted*, and *Stored*. Upon request, the system retrieves, decrypts, and reconstructs the video. Each transition is triggered by specific operations such as encryption or authentication.

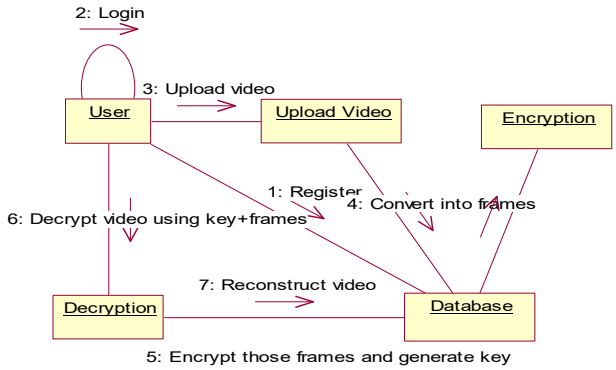
Sequence Diagram



Explanation:

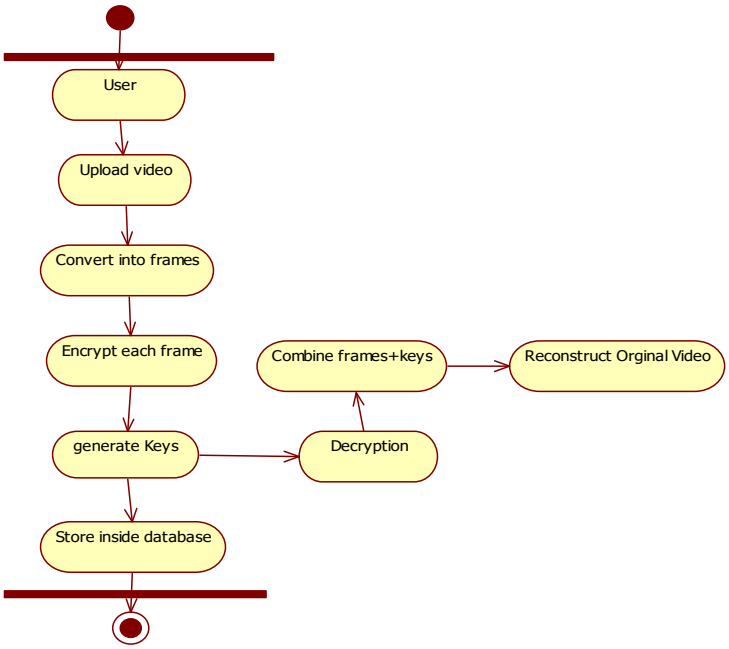
The sequence diagram depicts the chronological flow of operations. A video is uploaded, segmented into frames, encrypted, and stored. Blockchain mechanisms handle key verification and integrity checks. During playback, frames are retrieved, decrypted, and reassembled. This diagram emphasizes the temporal coordination between components.

Collaboration Diagram



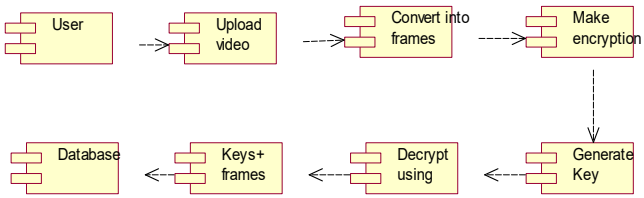
Explanation:
The collaboration diagram focuses on communication between system components. It highlights how objects interact structurally, exchanging messages to perform tasks such as data distribution, encryption coordination, and storage management.

Activity Diagram



Explanation:
The activity diagram outlines the workflow of video processing. It includes steps such as video upload, frame extraction, encryption, blockchain logging, storage, authentication, retrieval, and reconstruction. Decision nodes represent access control validation.

Component Diagram



Explanation:
The component diagram divides the system into modules such as user interface, video processing, encryption engine, blockchain manager, and storage system. These components interact to provide secure and efficient processing of video data.

Entity-Relationship (ER) Diagram

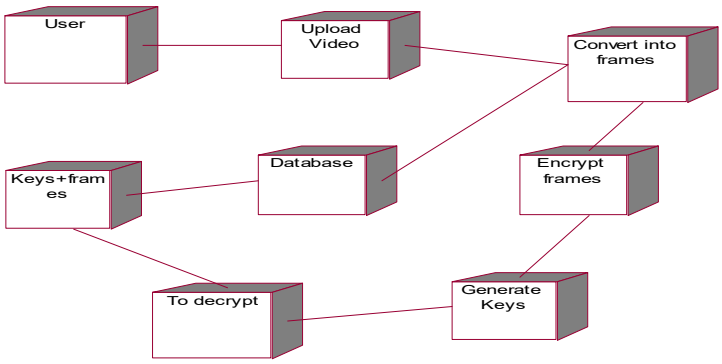
Explanation:
The ER diagram defines the logical data model of the system. Entities such as users, devices, blockchain records, and storage nodes are connected through relationships. The blockchain network consists of multiple fog nodes, each associated with a certificate authority and database instance. Transactions are ordered using RAFT consensus, and all components are deployed as containerized services.

Data Flow Diagram (DFD)

Explanation:
The DFD represents how data moves through the

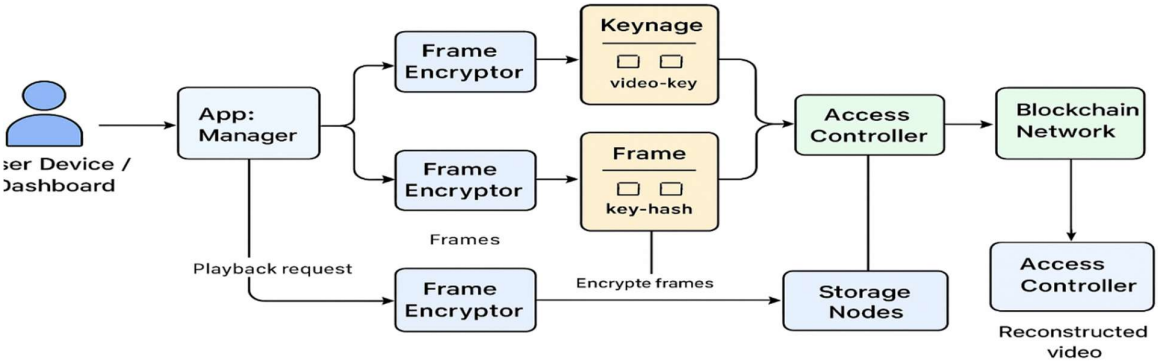
system. It identifies input sources, processing stages, storage locations, and outputs. Unlike sequence diagrams, it focuses on data movement rather than execution order.

Deployment Diagram



Explanation:
The deployment diagram shows the physical distribution of system components. User devices connect to application servers that perform processing and encryption. Blockchain nodes manage keys and policies, while distributed storage systems hold encrypted data. During playback, all components collaborate to deliver reconstructed video.

System Architecture



The proposed smart home security framework integrates blockchain, lightweight cryptography, and intelligent monitoring to provide end-to-end security. The architecture operates in two main phases: **user authentication** and **secure service access**.

During registration, users submit credentials to a private blockchain network. A cloud node generates a public-private key pair for each user. To enhance privacy, login credentials are derived using cryptographic hash functions: the username is generated from the public key, while the password is derived from the private key combined with the user-defined password. These credentials are stored on the blockchain, ensuring immutability and resistance to tampering.

During authentication, users recompute their hashed credentials using their private key, ensuring that only legitimate users can gain access. This mechanism effectively mitigates impersonation risks.

Additionally, encryption mechanisms protect video streams and sensitive data, making the system suitable for resource-constrained IoT environments. By combining blockchain technology with efficient cryptographic techniques and intelligent monitoring, the framework achieves secure device authentication, reliable key management, and tamper-proof data storage.

DEVELOPMENT TOOLS

This chapter describes the programming languages, frameworks, and tools used in the development of the proposed system. The implementation is primarily based on Java technologies, including Java Standard Edition, Java Enterprise Edition, and Java Micro Edition. Among these, Java Enterprise Edition (J2EE) has been selected as the main

platform because of its ability to support scalable, distributed, and web-based applications efficiently.

Java is a high-level programming language developed by James Gosling at Sun Microsystems in 1995. It derives its syntax from C and C++ but simplifies programming by eliminating complex features such as manual memory management. Java programs are compiled into bytecode, which runs on the Java Virtual Machine, making the language platform-independent. This feature supports the widely known principle of "write once, run anywhere," allowing applications to run across different systems without modification. Java is widely used in various domains, including web applications, enterprise systems, and mobile computing, due to its robustness, portability, and security.

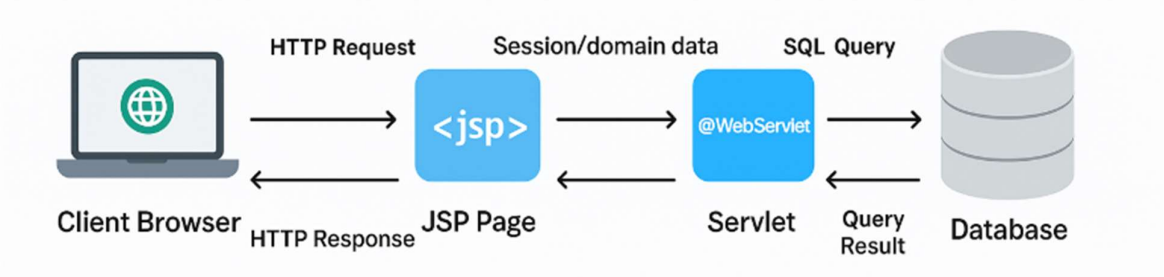
The popularity of Java among developers is largely due to its versatility and efficiency. It enables developers to create applications that can run on multiple platforms, develop dynamic web applications, and build server-side systems for handling data processing and user interactions. Java also supports integration of multiple services into unified systems and is suitable for developing applications for embedded devices and mobile platforms. These features have made Java one of the most widely used programming languages in the world.

Java follows object-oriented programming principles, which form the foundation of its design. Inheritance allows new classes to reuse properties and methods from existing classes, reducing redundancy and improving code maintainability. Encapsulation ensures that data and methods are combined into a single unit while restricting unauthorized access. Polymorphism enables methods to perform different operations based on the input provided, increasing flexibility. Dynamic binding allows method calls to be resolved at runtime, making programs more adaptable and extensible.

For user interface development, Java provides Swing, a GUI toolkit built on top of the Abstract Window Toolkit. Swing offers a wide range of components such as buttons, frames, and lists that help in creating interactive interfaces. Most Swing components are lightweight, meaning they are rendered using Java code rather than relying on the operating system. This ensures consistent behavior across different platforms. However, top-level containers such as frames and dialogs are heavyweight components, as they interact directly with the operating system.

Advanced Java technologies extend the capabilities of core Java to support enterprise-level and web-based applications. Servlets are server-side programs that handle client requests and generate responses, while JavaServer Pages enable the creation of dynamic web content by embedding Java code within HTML. Java Database Connectivity provides a standard interface for connecting to relational databases and executing SQL queries. JavaBeans are reusable components that encapsulate multiple objects into a single unit, promoting modular development. The Model-View-Controller architecture separates application logic, user interface, and control flow, improving maintainability. Filters and listeners are used to handle request processing and application events, while session management techniques help maintain user data across multiple interactions.

Apache Tomcat is used as the web server and servlet container for deploying the application. It is an open-source platform that supports Java technologies such as Servlets and JSP. Tomcat is lightweight, easy to configure, and widely used for developing and deploying web applications. It integrates well with popular development environments and simplifies the deployment process.



Software Testing

Software testing is a critical phase in the software development lifecycle aimed at identifying errors and ensuring system reliability. Testing involves systematically exercising software components to verify that the system meets specified requirements and performs as expected under various conditions. Its primary goal is to detect faults, weaknesses, or inconsistencies before deployment, thereby reducing the risk of failures in operational environments. Various testing strategies exist, each addressing specific functional or structural requirements.

Developing Testing Methodologies

The testing process begins with the creation of a comprehensive plan that outlines how the software

will be evaluated across multiple platforms and configurations. Effective testing methodologies rely on strict quality control procedures to verify that the application conforms to the specifications in the system requirements document. Key considerations in developing a testing framework include:

- Coverage of general functionality and special features.
- Validation across diverse hardware and software platforms.
- Verification that the software is free from defects and aligns with user expectations.
- Documentation of test strategies, objectives, and expected outcomes.

Types of Software Tests

Unit Testing

Unit testing focuses on individual components of the software, verifying that internal logic functions correctly and that inputs produce expected outputs. Each decision branch and internal flow is evaluated to ensure correctness. Conducted after the development of each unit, this type of testing is structural, relying on knowledge of the code, and is invasive in nature. Unit tests confirm that:

- Business processes execute according to specifications.
- All input scenarios are handled correctly.
- Each unique path produces the expected outcome.

Functional Testing

Functional testing evaluates whether the software performs as specified in business, technical, and user documentation. It focuses on validating:

System Testing

System testing assesses the behavior of the fully integrated software system. It ensures that all components interact as intended and that the system produces predictable and correct results. A typical example includes configuration-based system integration tests, emphasizing the verification of process flows and integration points.

Performance Testing

Performance testing measures the system's efficiency in producing outputs within specified time constraints. It evaluates metrics such as response time, throughput, and resource utilization during various operations, ensuring that the system meets performance expectations under expected load conditions.

Integration Testing

Integration testing examines the interactions between multiple software components or systems. The primary goal is to identify interface defects and ensure that integrated components function correctly together. This testing can occur at different levels, including module-to-module interactions and system-to-system communications.

Application and Future Enhancements

The proposed framework envisions several key improvements to enhance scalability, security, and real-world applicability:

1. **Pluggable Proof of Voting (PoV) Consensus:** A lightweight PoV consensus mechanism will be integrated into Hyperledger Fabric to improve efficiency in validating transactions. This approach has proven effective in maintaining low-latency and energy-efficient consensus operations.
2. **IoT Testbed Development:** A real-time IoT testbed will be designed to evaluate the framework under practical scenarios. This will facilitate rigorous testing of system performance, scalability, and resilience in dynamic environments.
3. **Off-Chain Transactions:** Scalability challenges will be addressed by leveraging off-chain transaction techniques, reducing the load on the main blockchain while maintaining security and integrity.
4. **Device Attestation and Identity Verification:** The framework will incorporate robust device attestation mechanisms, including cryptographic key protection, digital signature-based identity verification, secure key injection, and trustworthiness-based evidence generation. These measures ensure that only authenticated and trusted devices participate in the network.

Conclusion

This work presents a blockchain-enabled security framework for smart homes that integrates user/device registration, secure key management, encrypted storage, and multimedia handling into a unified system.

Key highlights include:

- **Data Security:** SHA-based hashing ensures data integrity, while AES-based symmetric encryption protects sensor data and video streams from unauthorized access.
- **Immutable Logging:** Blockchain provides a tamper-proof ledger for tracking device activity, access requests, and storage operations.
- **Automated Policy Enforcement:** Smart contracts enforce security policies without relying on third-party interventions, reducing operational risks.
- **Multimedia Security:** Video streams are processed by splitting them into frames, encrypting each frame, and distributing them across storage nodes. Authorized users can retrieve and decrypt these frames using blockchain-managed keys.
- **User Interface:** A smart home dashboard enables intuitive monitoring of devices, secure access management, and visualization of encrypted data.

Overall, this framework demonstrates that combining blockchain, SHA-based integrity verification, and AES-based encryption provides a privacy-preserving, tamper-proof, and efficient solution for smart home security. It offers a scalable and practical approach to safeguarding IoT-enabled environments.

References

[1] R. Chataut, A. Phoummalayvane, and R. Akl, "Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0," *Sensors*, vol. 23, no. 16, p. 7194, Aug. 2023.
[2] A. Khanna and S. Kaur, "Internet of Things (IoT), applications and challenges: A comprehensive review," *Wireless Pers. Commun.*, vol. 114, no. 2, pp. 1687–1762, Sep. 2020.
[3] M. Nassereddine and A. Khang, *Applications of Internet of Things (IoT) in smart cities*, CRC Press, 2024, pp. 109–136.
[4] R. Das and M. M. Inuwa, "A review on fog computing: Issues, characteristics, challenges, and potential applications," *Telematics Informat. Rep.*, vol. 10, Jun. 2023, Art. no. 100049.
[5] H. Lin and N. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, Jul. 2016.
[6] O. Popoola et al., "A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: Problems, challenges and solutions," *Blockchain: Res. Appl.*, vol. 5, no. 2, Jun. 2024, Art. no. 100178.
[7] M. Albany et al., "A review: Secure Internet of thing system for smart houses," *Proc. Comput. Sci.*, vol. 201, pp. 437–444, Jan. 2022.
[8] S. Nakamoto, *Bitcoin Whitepaper*, 2008. [Online].
[9] N. Adhikari and M. Ramkumar, "IoT and blockchain integration: Applications, opportunities, and challenges," *Network*, vol. 3, no. 1, pp. 115–141, Jan. 2023.
[10] Q. Ma, H. Tan, and T. Zhou, "Mutual authentication scheme for smart devices in IoT-enabled smart home systems," *Comput. Standards Interfaces*, vol. 86, Aug. 2023, Art. no. 103743.
[11] X. Xu, Y. Guo, and Y. Guo, "Fog-enabled private blockchain-based identity authentication scheme for smart home," *Comput. Commun.*, vol. 205, pp. 58–68, May 2023.
[12] M. Ammi, S. Alarabi, and E. Benkhelifa, "Customized blockchain-based architecture for secure smart home for lightweight IoT," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102482.
[13] A. Qashlan et al., "Privacy-preserving mechanism in smart home using blockchain," *IEEE Access*, vol. 9, pp. 103651–103669, 2021.
[14] A. Huszti, S. Kovács, and N. Oláh, "Scalable, password-based and threshold authentication for smart homes," *Int. J. Inf. Secur.*, vol. 21, no. 4, pp. 707–723, Aug. 2022.
[15] C. Lin et al., "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.
[16] P. Kumar et al., "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors J.*, vol. 16, no. 1, pp. 254–264, Jan. 2016.
[17] M. Saeed et al., "Trust management technique using blockchain in smart building," *Eng. Proc.*, vol. 20, no. 1, p. 24, Aug. 2022.
[18] I. Asghar et al., "Fortifying smart home security: A robust and efficient user authentication scheme to counter node capture attacks," *Sensors*, vol. 23, no. 16, p. 7268, Aug. 2023.
[19] R. Paul et al., "IoT-based secure smart city architecture using blockchain," in *Proc. 2nd Int. Conf. Data Sci. Bus. Analytics (ICDSBA)*, Sep. 2018, pp. 215–220.
[20] A. Bounceur et al., "CupCarbon: A platform for the design, simulation, and visualization of radio propagation and interferences in IoT networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–4.
[21] Hyperledger Caliper, 2022.