Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085653 Published Online: 08 October 2026

ARTICLE

Blockchain-Based Smart Home for Secure Authentication and Authorization

Idris Adedamola Abdulhameed[1], **Sulaiman M. Karim**[2], **Musaria K. Mahmood**[3] and **Shehzad Ashraf Chaudhry**[4,5,*]

[1]School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea
[2]College of Health and Medical Techniques, Northern Technical University, Al-Dour, Iraq
[3]Department of Energy Systems Engineering, Ankara Yildirim Beyazit University, Ankara, Türkiye
[4]Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[5]Department of Software Engineering, Faculty of Engineering and Architecture, Nisantasi University, Istanbul, Türkiye

*Corresponding Author: Shehzad Ashraf Chaudhry. Email: ashraf.shehzad.ch@gmail.com

Received: 15 May 2026; Accepted: 27 August 2026

ABSTRACT: The expansion of smart home (SH) networks introduces security vulnerabilities due to centralized architectures and resource-constrained Internet of Things (IoT) hardware. Centralized systems remain susceptible to single-point-of-failure vulnerabilities, identity impersonation, and unauthorized session key manipulation. To address these issues, this paper proposes a Blochchain based secure data exchange protocol for smart homes (*BSDE-SH*). The framework utilizes a dual-layer hybrid architecture: high-speed localized message loops among edge routers are validated via a lightweight Practical Byzantine Fault Tolerance (PBFT) mechanism ensuring a deterministic $3f + 1$ fault-tolerance quorum, while long-term access policies are executed on an EVM-compatible private Ethereum decentralized runtime. For low-latency session key derivation, *BSDE-SH* leverages an optimized multi-factor cryptographic handshake built upon Elliptic Curve Cryptography (ECC). We evaluate the protocol using Cisco Packet Tracer for network reliability and the MIRACL library for computational performance. Experimental results confirm that our certificate-based scheme completes mutual authentication in three messages with a low computational latency of **11.521 ms**, outperforming six state-of-the-art baselines. Furthermore, a prototype deployment evaluated via Ganache and Truffle demonstrates that key operations incur minimal fees, achieving a structural contract execution cost of **1,485,218** gas-reflecting a **412% improvement in overall gas efficiency** over conventional benchmarks. Finally, formal security verification via the Scyther framework proves complete resilience against multi-vector adversarial exploits, validating the protocol's suitability for scalable IoT infrastructure.

KEYWORDS: Internet of Things (IoT); smart home (SH); authentication; blockchain; elliptic curve cryptography (ECC); practical byzantine fault tolerance (PBFT)

1 Introduction

The global smart home (SH) market is projected to exceed 7 billion dollars by 2025, coinciding with an industry-wide transition where 80% of the networking sector adopts automation technologies [1]. Concurrently, demographic shifts indicate that approximately one-fourth of the global population will exceed 60 years of age by 2050 [2], necessitating robust assistive tech paradigms to maintain longevity and well-being. Modern literature bifurcates SH implementation into software-simulated environments leveraging packet-tracking models [3–6] and hardware-driven architectures integrating distributed blockchain frameworks alongside specialized components like smart mirrors [7–9]. Combining Internet of Things

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

(IoT) infrastructure with automated machine learning allows heterogeneous hardware networks to interface seamlessly over unified frameworks, such as NodeMCU-controlled microcontrollers linked via mobile access nodes [10].

Despite these architectural developments, edge networks remain highly susceptible to supply-chain tampering, unauthorized remote access, and credential leakage [11]. Network stability is threatened by identity-based vectors including malicious impersonation, hardware cloning, and spoofing attacks [12]. Thus, satisfying nominal functional and operational requirements does not eliminate vulnerability to multi-vector adversarial exploitation, allowing malicious entities to compromise system privacy and orchestrate widespread network breaches through unverified mobile nodes or edge routing points [13].

1.1 Background and Security Challenges in Smart Homes

Smart home networks rely on interconnected IoT nodes to handle environment telemetry and automated system configuration. However, operational heterogeneity and resource constraints-specifically limited computational power and restricted memory spaces-preclude edge IoT units from deploying resource-heavy traditional cryptographic frameworks. This limitation leaves the hardware layer highly vulnerable to active data modification, passive eavesdropping, spoofing, and distributed denial-of-service (DDoS) attempts [14].

1.2 The Role of Blockchain in Smart Home Security

To remove single-point-of-failure vulnerabilities, blockchain serves as a distributed, decentralized security architecture. Utilizing shared ledgers across local edge routers allows cryptographically verifying transaction authenticity without massive cloud dependencies. Because standard public Proof-of-Work (PoW) consensus algorithms impose prohibitive processing overhead, this paper introduces an optimized dual-layer hybrid architecture. Under this approach, rapid localized message loops among edge routers are validated via a lightweight Practical Byzantine Fault Tolerance (PBFT) scheme to guarantee a deterministic $3f+1$ fault-tolerance quorum for near-instant transaction finality. Concurrently, long-term policy access rules, contract validations, and Registration Authority are safely executed and synchronized via an EVM-compatible, decentralized, private Ethereum runtime.

1.3 Layered Blockchain Architecture and Layer-Specific Threats

To evaluate system risks systematically, the distributed architecture is partitioned into four distinct functional tiers:

- **Data Layer:** Manages data blocks, hash tracking, and asymmetric cryptographic primitives for core ledger immutability; protected against localized data injection and key exposure via robust Elliptic Curve Cryptography (ECC).
- **Network Layer:** Regulates peer-to-peer (P2P) block propagation and node discovery; mitigates Sybil, Eclipse, and routing attacks through strict router-level validation profiles.
- **Consensus Layer:** Executes node agreement validations to protect state history. By deploying a lightweight PBFT consensus paradigm instead of unoptimized open-mining frameworks, the system enforces a deterministic $3f+1$ fault-tolerance quorum. $3f+1$ represents the total number of validator nodes required to tolerate up to f Byzantine faults, while $2f+1$ represents the commit quorum threshold. This guarantees absolute mathematical finality directly at the Consensus Layer (rather than at the routing tier) to prevent state forks, and coordinates seamlessly with our private EVM-compatible decentralized runtime for long-term policy verification.

- **Application Layer:** Hosts client logic, smart contract execution routines (`AddSHD`, `RemoveSHD`), and public APIs; enforces zero-trust role access checks directly inside Solidity contract states to block reentrancy or state hijacking.

Additionally, blockchain technology continues to mature, providing verified security layers across multiple domains including finance, healthcare, agriculture, and intelligent transportation systems [15–18]. Fig. 1 illustrates the proposed blockchain-centric SH topology, detailing the logical trust boundaries separating the Public (Untrusted) Domain from the Trusted (Secure) Domain while standardizing system terminology. Within this framework, the *Router/Gateway* serves as the edge ingress point managing external WAN/LAN traffic; the *RADIUS Server* acts as the central local AAA (Authentication, Authorization, and Accounting) entity handling cryptographic device handshakes; and the integrated *Blockchain Fabric* manages immutable policy logging. The architecture comprises IoT devices communicating via secure Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) and EAP-Tunneled TLS (EAP-TTLS) channels, alongside edge entities (such as the RADIUS server and user nodes using EAP-Protected EAP [EAP-PEAP] over Wi-Fi Protected Access 3 [WPA3]). To satisfy these requirements, the Blockchain Fabric conceptually defines consensus and validation layers, which are functionally instantiated and simulated in our implementation using the Ganache local blockchain environment via JavaScript Object Notation Remote Procedure Call (JSON-RPC) gateways.

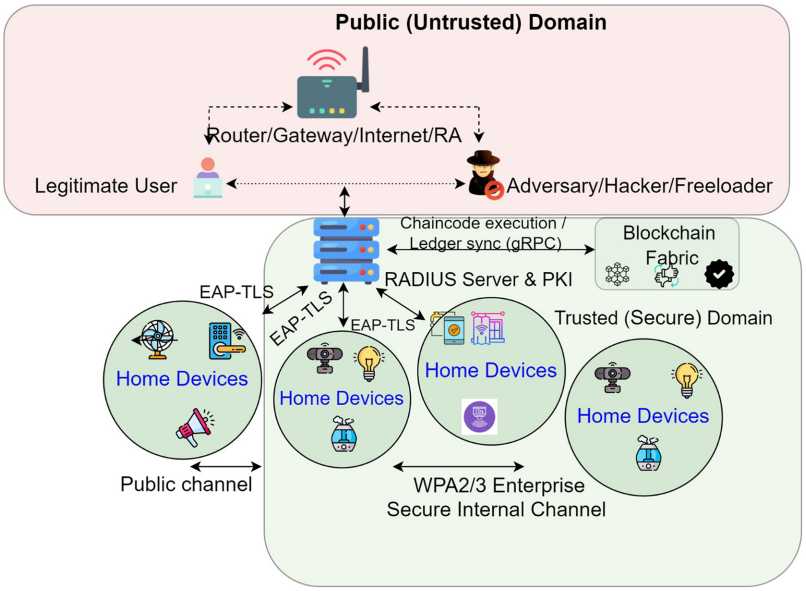


Figure 1: Proposed blockchain-centric SH network architecture.

1.4 Threat Model

The architectural surface of resource constrained heterogeneous SH networks provides broad avenues for inter-node exploitation. To accurately map these vectors, this framework adopts the standard Dolev-Yao threat criteria [19] alongside Canetti-Krawczyk adversary models, granting an active adversary M absolute execution rights to intercept, block, inject, modify, or replay public channel transmissions. Security goals require lightweight, low-operand protocols that satisfy rigorous privacy parameters without scaling resource latency, which is achieved here using high-efficiency ECC setups.

1.5 Research Contributions

The primary contributions of this paper are threefold. We design and execute a localized nine-device SH network modeled in Cisco Packet Tracer that interacts securely over a decentralized substrate. Secondly, credential-based remote node validation loops utilizing network-layer IP management and access controls are implemented. Finally, we develop a lightweight digital certificate-driven mutual authentication protocol that completes in only three messages, with lower communication and computation latency than six state-of-the-art protocols.

2 Related Work

Research on authentication in smart home (SH) and IoT systems spans multiple directions, including access control, lightweight cryptography, ECC-based schemes, and blockchain integration. In edge and access control security, study [20] combined attribute-based policies with smart contracts and edge computing, while [21] modeled home automation threats using attack trees. Lightweight protocols for constrained devices leverage hash functions, XOR, and symmetric encryption [22]. Similarly, *PrivHome* [23] uses symmetric primitives but remains vulnerable to gateway compromise. To address such limitations, ECC-based authentication schemes [24,25] provide secure session key establishment and privacy preservation under formal analysis, including AVISPA evaluation. Several ECC-based SH authentication frameworks [26–28] demonstrate improved efficiency and secure device verification under resource constraints. Blockchain has also been widely adopted for SH security. A three-layer blockchain architecture improves gateway protection, while mutual authentication schemes [29,30] enhance trust without centralized servers. Further works [31,32] apply ledger-based access control and Hyperledger Fabric to achieve fine-grained and tamper-resistant authorization. Hybrid blockchain solutions such as private and consortium models [33–35] reduce overhead while improving auditability and secure data exchange. Additional studies [36–39] explore integration with edge computing for scalable IoT and smart grid security. Formal verification-based schemes [40,41] confirm resistance to major attacks but highlight trade-offs in computational complexity. Recent decentralized architectures using Ethereum smart contracts further enable secure distributed IoT coordination [42]. Recent studies [43–45] emphasize that large-scale IoT and federated learning systems require robust authentication at the edge, particularly for heterogeneous environments. Our proposed framework addresses this gap through an EVM-compatible authentication model optimized for reduced computational overhead. Table 1 demonstrates that existing smart home authentication schemes typically focus on selected security mechanisms, such as blockchain, ECC-based authentication, or lightweight mutual authentication. In contrast, the proposed BSDE-SH framework unifies certificate-based ECC authentication, PBFT consensus, blockchain-enabled smart contracts, lightweight session key establishment, formal security verification, and hardware validation within a single architecture. This integrated design enhances authentication efficiency, strengthens security assurance, and supports practical deployment in resource-constrained smart home IoT environments.

Table 1: Comparison of the BSDE-SH framework with related blockchain-based schemes.

Scheme	BC	ECC	PBFT	Mutual	Formal	SC	HW
SH-BlockCC [11]	✓	–	–	✓	–	–	–
HomeChain [29]	✓	–	–	✓	✓	–	–
Consortium BC Auth. [34]	✓	–	–	✓	–	✓	–

(Continued)

Table 1 (continued)

Scheme	BC	ECC	PBFT	Mutual	Formal	SC	HW
Blockchain Auth. & Key Exchange [37]	✓	✓	–	✓	✓	–	–
SecFHome [46]	–	✓	–	✓	✓	–	–
ECC-Based Auth. [47]	–	✓	–	✓	–	–	–
Proposed BSDE-SH	✓	✓	✓	✓	✓	✓	✓

3 Proposed Methodology

This section presents the system design, implementation environment, and the proposed BSDE-SH authentication and blockchain framework in a unified and compact form.

3.1 System Model, Requirements, and Implementation Environment

The proposed framework is designed to satisfy key smart home requirements, including security, privacy, scalability, and trust decentralization. Security is achieved through ECC-based dynamic key generation to resist cloning, interception, and tampering. Privacy is preserved using dynamic pseudo-identities (PID), which conceal real device identities. Scalability is ensured through efficient transaction validation under increasing device density, while trust is decentralized using a distributed $3f+1$ fault-tolerant consensus model. The smart home environment is implemented using Cisco Packet Tracer (v8.2.1), where IoT devices such as sensors, lighting systems, and appliances are assigned unique IP addresses under a unified SSID ("Home") configured with WPA2-AES encryption. Authentication and access control are enforced through a RADIUS-enabled AAA mechanism integrated into the home gateway/router, ensuring that only authorized users can access and manage devices. The overall system configuration workflow is illustrated in Fig. 2.

Cisco Packet Tracer is employed solely to model the smart-home network topology, communication workflow, and device connectivity. It is not used to validate the cryptographic security of the proposed authentication protocol. The security properties of the framework are instead established through formal ROR-model analysis, Scyther-based protocol verification; whereas the computational performance evaluation is furnished using the MIRACL cryptographic library.

3.2 BSDE-SH Authentication and Cryptographic Foundation

The proposed BSDE-SH framework integrates ECC with blockchain-based verification to secure smart-home communications. The Registration Authority (RA) selects an elliptic curve $E_p : y^2 = x^3 + cx + d$, where $4c^3 + 27d^2 \neq 0$, generates its private key $\text{Prv} * RA$, and computes $\text{Pub} * RA = \text{Prv} * RA \cdot \text{GEN}$. The public parameters $(E_p, \text{GEN}, \text{Pub} * RA, h(\cdot))$ are published, while private keys remain protected. BSDE-SH uses the NIST `secp256r1` (P-256) curve, providing approximately 128-bit classical security. The generator GEN has prime order n, satisfying $n \cdot \text{GEN} = \mathcal{O}$.

During secure registration, the RA provisfions control rooms, routers, and devices with their identities, certificates, and cryptographic credentials before deployment. BSDE-SH employs a dual-layer hybrid architecture. A private consortium blockchain uses PBFT among validator routers for efficient local authentication, transaction validation, and consensus. After consensus, the authenticated transaction state and its cryptographic digest are anchored to an Ethereum-compatible smart contract for immutable storage, public

verification, and auditability. Thus, resource-constrained devices benefit from lightweight local consensus without directly interacting with the public blockchain.

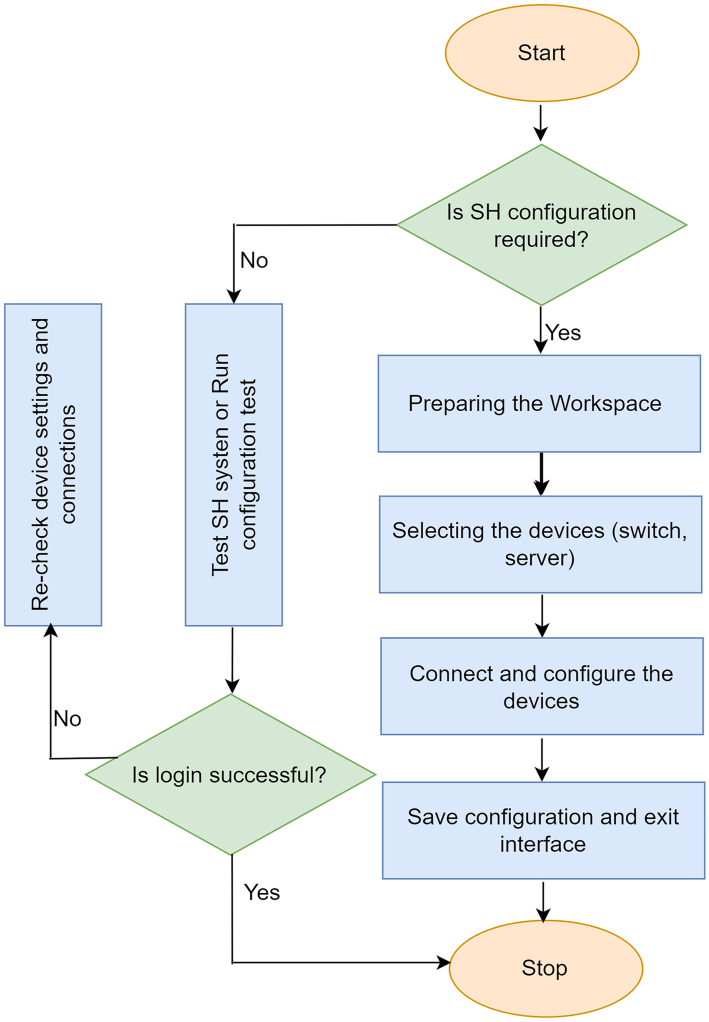


Figure 2: Workflow for SH device configuration and login validation.

3.3 Authentication and Secure Communication

This section describes the mutual authentication and session key establishment process between the device and gateway. Upon successful authentication, both entities independently derive a shared session key using shared secrets, cryptographic hashes, and ephemeral nonces. The resulting key provides perfect forward secrecy and secures subsequent communication against eavesdropping, tampering, and unauthorized access, as illustrated in Fig. 3. The resulting session key is defined as:

$$\mathrm{SSKey}_{D-R} = h(\mathrm{PID}_D \parallel \mathrm{PID}_R \parallel \mathrm{DKey} \parallel \mathrm{PK}_D \parallel \mathrm{PK}_R) \tag{1}$$

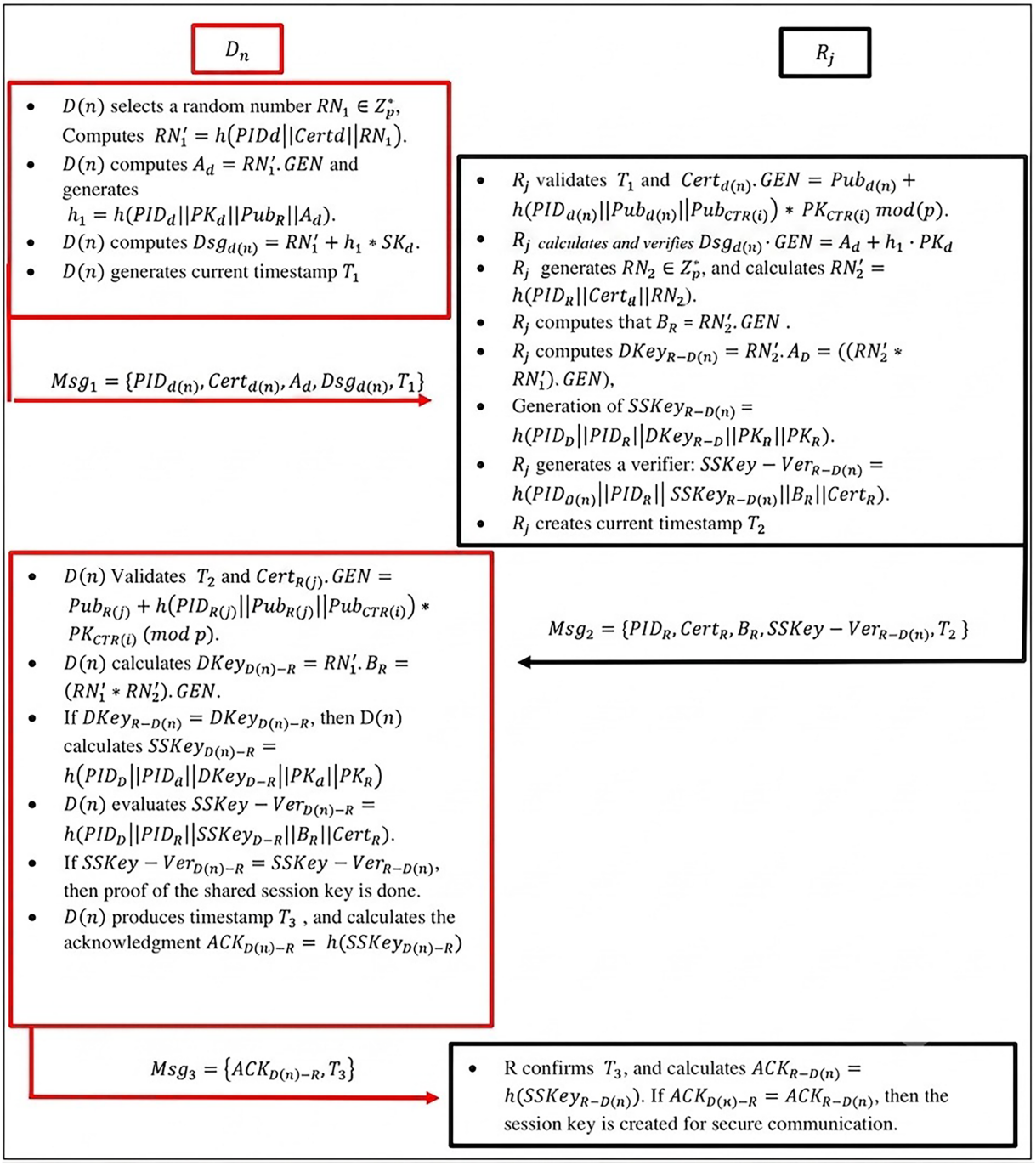


Figure 3: Mutual authentication phase.

3.4 Certificate Revocation and Re-Registration

In BSDE-SH, the Registration Authority (RA) revokes certificates of compromised devices, routers, or users and updates their status on the permissioned blockchain. Validator routers reject authentication requests from revoked identities. For re-registration, legitimate entities securely authenticate with the RA

and receive a fresh pseudo-identity, certificate, and key pair, while previous credentials are invalidated. This mechanism enables secure recovery from credential compromise and prevents reuse of obsolete credentials.

3.5 Blockchain-Based Integrity and Consensus Mechanism

To ensure tamper-resistant logging and auditability, the system employs a permissioned blockchain with PBFT consensus. Each block contains essential components such as the previous block hash, Merkle root, timestamp, payload, and digital signature (see Table 2). Blocks are generated and validated through a leader-based consensus mechanism, where a block is accepted only if $VC \geq 2f + 1$, where f represents the number of faulty nodes and VC denotes the number of valid votes.

Table 2: Structure of block j (Isolated notation mapping).

HPB	Hash of Previous Block
HCB	Hash of Current Block
SHCB	Signature on HCB
BC	Blockchain
MTR/HT	Merkle Tree Root/Hash Tree
T	Timestamp
PK_{SHD_j}	Public Key of Signer SHD_j
PoB	Payload of Block
$LEPK_j$	List of Encrypted Transactions
BV	Block Version
BLVS	Block Validation State

The blockchain structure and block organization are illustrated in Fig. 4, while the consensus process and tracking of the Block Validation State (BLVS) are described in Algorithm 1. This ensures integrity, immutability, and distributed trust across the smart home environment.

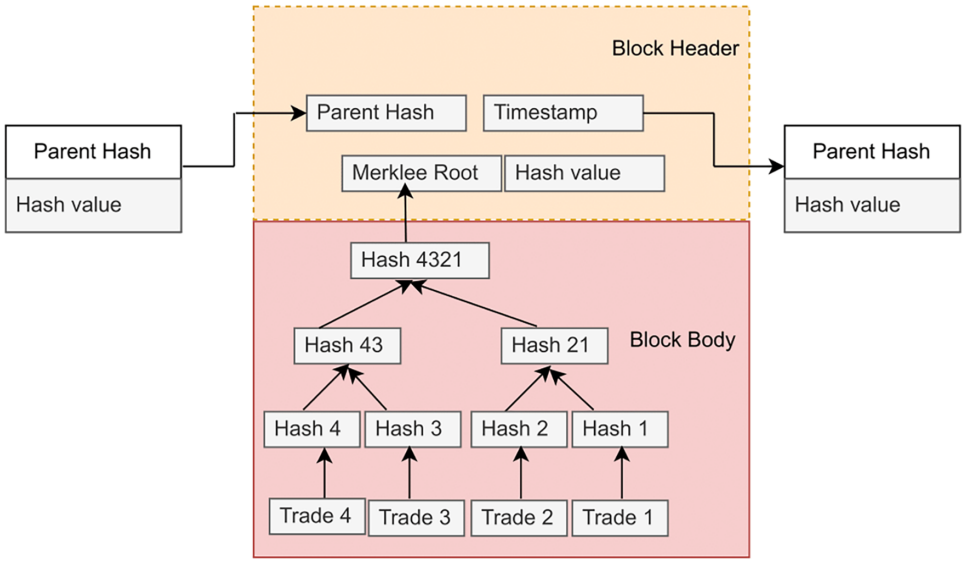


Figure 4: Blockchain data structure.

3.6 Formal Security Reduction to ECDLP

We establish the existential unforgeability of the proposed certificate scheme through a reduction to the Elliptic Curve Discrete Logarithm Problem (ECDLP), assuming the ECDLP is computationally intractable.

Proof: Assume an adversary $\mathcal{A}$ can forge a valid certificate $Cert_{CTR_i} = \text{Prv-}K_{CTR_i} + h(\text{PID} * i \parallel w_i) \cdot \text{Prv} * RA$ with non-negligible advantage ε. We construct a challenger $\mathcal{C}$ that uses such a forgery to solve an ECDLP instance $\text{Pub} = x \cdot \text{GEN}$ by setting $\text{Pub}_{RA} = \text{Pub}$ and simulating the random oracle. □

Using the Forking Lemma, $\mathcal{C}$ obtains two valid forgeries for the same commitment with distinct hash values $h_1 \neq h_2$:

$$Cert^{(1)} = \text{Prv-}K + h_1 x \pmod{n}, \tag{2}$$

$$Cert^{(2)} = \text{Prv-}K + h_2 x \pmod{n}. \tag{3}$$

Algorithm 1: Consensus on blockchain block insertion and verification

Input: Block $B(j)$, number of faulty routers NF_R
Output: After validating $B(j)$, commit and add it to the blockchain
Assume SHD (LDR) is chosen as leader to add $B(j)$
for each follower node LDR_k **do**
    Generate timestamp T_{SHD_j}
    Create voting request $VReq$
    Encrypt $VReq$ with PK_{SHD_j}
    Forward encrypted request to SHD_j as $LEPK_j(VReq, T_{SHD_j})$, $j \neq LDR$
end for
for each follower SHD_j **do**
    Receive message from LDR at time $T^*_{SHD_j}$
    Decipher message: $(VReq', T_{SHD_j}) = D_{MSK_{SHD_j}}\left[E_{PK_{SHD_j}}(VReq, T_{SHD_j})\right]$
    Verify block $B(j)$ using Merkle root, timestamp, block hash, signature
    if verification succeeds **then**
        Encrypt $(VReq, VRep, BLVS)$ with PK_{LDR}
        Send encrypted response to leader
    end if
end for
Initialize vote counter: $VC \leftarrow 0$
for each received response $E_{PK_{LDR}}(VReq', VRep, BLVS)$ from SHD_j **do**
    Decrypt using DK_{LDR}: $(VReq', VRep, BLVS) = DK_{LDR}\left[E_{PK_{LDR}}(VReq', VRep, BLVS)\right]$
    if $VReq' = VReq$ **and** $VRep$ = valid **and** $BLVS$ = valid **then**
        Increment $VC \leftarrow VC + 1$
    end if
end for
if $VC > 2 \cdot NF_R + 1$ **then**
    Send commit response to each follower
    Add $B(j)$ to blockchain
end if

Subtracting them allows $\mathcal{C}$ to recover

$$x = (Cert^{(1)} - Cert^{(2)})(h_1 - h_2)^{-1} \pmod{n}. \quad (4)$$

Hence, a successful certificate forgery would yield a polynomial-time solution to ECDLP, contradicting its assumed hardness. Therefore, the proposed certificate scheme is existentially unforgeable.

4 Result and Discussion

This section tests and validates the security of the proposed system using two formal security analyses: the Scyther tool and the Real-Or-Random (ROR) Oracle model [48]. There are two roles in our protocol: SHD and the Server (R), with the terms "Ok" and "Fail" indicating the correctness and falsehood of claims, respectively [37]. We also perform an informal security analysis of the system as presented below.

4.1 Security Analysis

This section provides a rigorous security evaluation of the proposed framework against malicious entities.

4.1.1 Cryptographic Hardness Assumptions

To establish the formal security bounds of the proposed protocol against a probabilistic polynomial-time (PPT) adversary, our scheme relies on the following foundational cryptographic hardness assumptions:

- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** Given an elliptic curve generator point P and a random element $Q = x \cdot P$ within a prime-order group, it is computationally infeasible for any PPT adversary to compute the scalar multiplier x with a non-negligible advantage.
- **Computational Diffie-Hellman (CDH) Assumption:** Given the triplet tuple $(P, a \cdot P, b \cdot P)$ for unknown random scalars a and b, it is computationally intractable for a PPT adversary to calculate the target point product $a \cdot b \cdot P$.
- **Collision Resistance Hash Function:** For a cryptographically secure one-way hash function $H(\cdot)$, it is computationally infeasible to find two distinct inputs m_1 and m_2 such that $H(m_1) = H(m_2)$.

4.1.2 Threat Model and Security Environment

As established in our introductory premises, the operational environment assumes an active adversary capable of intercepting public communication channels, dropping packets, or replaying stale authentication metrics. The network entities are considered semi-trusted, necessitating robust validation loops across the routing layer.

4.1.3 Resistance to Common Attacks

In this section, we demonstrate that the proposed system incorporates robust cryptographic defenses and structurally withstands a variety of known multi-vector attacks without requiring sandbox script testbeds:

- **Man-in-the-middle attack:** To successfully launch a MITM attack, an adversary may intercept and modify authentication messages exchanged over the public channel and attempt to convince the communicating entities that the modified messages are legitimate. However, generating a valid modified authentication message requires knowledge of the secret/private parameters used to construct the corresponding authentication values. Without such knowledge, an adversary cannot generate authentication

data that satisfies the verification conditions at the receiving entity. Consequently, any unauthorized modification of the authentication messages is detected during verification, thereby protecting the proposed scheme against MITM attacks.

- **De-synchronization attack:** To prevent potential de-synchronization between the smart device and the gateway, the gateway temporarily stores the previous temporary identity (PID_{old}), along with the current identity (PID). Upon successful completion of the subsequent login session, (PID_{old}) is updated accordingly. Maintaining identities from two consecutive sessions prevents synchronization failures that may occur if the gateway updates the user's identity in its verifier. At the same time, an adversary can intercept or block the corresponding response message. In such a case, the smart device may retain the previous identity. At the same time, the gateway has already switched to the new one, resulting in identity de-synchronization and potentially causing the subsequent login attempt to fail.
- **Replay attack:** Because our scheme's structurally updated session key derivation formula ($SSKey = h(PK_R \parallel T_i \parallel \text{Nonce}_A)$) explicitly integrates both synchronized high-resolution timestamps and high-entropy nonces uniquely differing between execution rounds, the network remains fully protected from replay exploits [48], rendering past captured transit elements and outdated session identifiers computationally invalid.
- **Mobile theft/loss attack:** The remote server does not retain the user's password, and there is no static password verification table, preventing offline brute-force guessing or extraction attempts.
- **Smart device recording attack:** Although an attacker may intercept the exchanged protocol messages and inject fake parameters, the information captured will not reveal the true identity of the smart device, long-term confidential credentials, the session key, or other security-sensitive information. Consequently, the recorded communication cannot be used to impersonate the legitimate smart device or compromise subsequent authentication sessions.
- **Key forward/backward secrecy attack:** An attacker cannot mathematically derive previously established or subsequent session keys from an intercepted or compromised current session key. Furthermore, even in the case of a long-term secret credentials compromise, the historical and future session keys remain completely secure. Because every session key relies on independent, fresh ephemeral nonces processed via point-scalar multiplications, a compromise of static long-term elements does not retroactively expose historic session key iterations, guaranteeing robust forward secrecy.
- **Mutual authentication:** Mutual authentication is achieved through the hash-based consistency checks performed by both communicating entities during the authentication procedure described in Section 3.3. Each entity independently reconstructs the expected authentication value using the received message parameters and its locally available secret and session-specific values, and verifies it against the corresponding received value. Successful verification at both ends authenticates both participating entities, thereby providing mutual authentication.
- **Untraceability:** Temporary pseudo-identities (PID) of users and smart devices are dynamically updated after every successful session closure by folding in a fresh, high-entropy random nonce r. Because the token is updated via a one-way cryptographic hash function $h(\cdot)$, past and future pseudo-identities ($\text{PID}^{(1)}, \text{PID}^{(2)}$) remain computationally un-linkable, providing resistance to cross-session tracking under the assumed cryptographic security model.
- **Anonymity:** Without explicit knowledge of localized secret master keying parameters and ephemeral randomness, an active adversary capturing public channel exchanges cannot extract the real identity or identifying information of any legitimate node. Under the random oracle model for the underlying hash function, the generated pseudo-identities are computationally indistinguishable from random strings, providing anonymity against adversaries under the assumed security model.

- **Router impersonation:** For successful router impersonation, an adversary would need to construct a valid (Msg_2) that passes the corresponding authentication checks at the receiving entity. In particular, this requires a valid router certificate (Cert_R) and session-key verifier $(\text{SSKey-Ver}_{R-D(n)})$. Forging a valid (Cert_R) is computationally infeasible under the ECDLP assumption, as formally established in Section 3.6, while generating a valid $(\text{SSKey-Ver}_{R-D(n)})$ requires the corresponding session-specific secret information. Therefore, an adversary cannot construct a valid authenticated (Msg_2) on behalf of a legitimate router.
- **Ephemeral Secret Leakage (ESL) Attack:** Under the Canetti–Krawczyk adversary model, compromise of ephemeral secret credentials alone does not enable an adversary to derive the established session key, as its computation also depends on both fresh ephemeral session secrets and long term secret keys.
- **Privileged-insider attack:** Even if a malicious insider compromises a node and extracts stored parameters, they cannot compute useful credentials without unauthorized prerequisites.
- **Impersonation attack:** To impersonate a legitimate edge user, an adversary must construct a fresh and valid login message that satisfies the corresponding authentication and freshness checks. However, generating the required authentication parameters depends on secret credentials and session-specific values that cannot be derived from intercepted public messages alone. Therefore, an adversary cannot successfully impersonate a legitimate edge user solely by intercepting or replaying previously exchanged messages.
- **Public-key replacement attack:** The proposed framework successfully neutralizes public-key replacement threats through two tight security bounds. First, every public key Pub_i is cryptographically bound to its associated pseudo-identity PID_i via digital signatures during the secure registration layer. Second, these identity-key mappings are anchored immutably across the distributed blockchain validators. Because any incoming transaction payload must pass state-machine quorum validation against the authenticated blockchain states, an adversary attempting an unauthorized public-key swap will immediately trigger a consensus validation failure, rendering the replacement attack completely ineffective.

4.1.4 Data Integrity and Privacy Protection Analysis

Data integrity is maintained across two critical fronts within our architecture as validated in Section 4.1.3. First, data-in-transit is protected via Elliptic Curve Diffie-Hellman (ECDH) key derivation (SSKey_{D-R}). Second, data-at-rest is stored within immutable ledger block states protected by SHA-256 hash chaining. User privacy preservation is strictly maintained because real device identifiers are completely masked behind dynamic pseudo-identities (PID).

4.1.5 Formal Security Analysis under the ROR Oracle Model

This section presents the formal security analysis to validate secure key sharing among the SH network devices. The ROR model is a reliable framework for analyzing authentication systems [46].

4.1.6 Authentication Protocol

Let $Q = (U, R, D)$ represent an authentication process involving a user U, a router (gateway) R, and a smart device D. Probabilistic polynomial-time techniques are used to model all protocol participants, including adversaries. Participants may run multiple instances, including one where party Y is modeled, where $Y \in \{U, R, D\}$. The protocol aims to mutually authenticate participants and establish a session key for securing future communications.

A malicious attacker M can execute the following queries:

- **Execution:** M uses this query to fabricate exchanged content between legitimate participants, resulting in an Active attack.
- **Corrupt a smart device:** M can access all information stored in the smart device's memory.
- **Corrupt the smart gateway or router:** After executing this query, the gateway reveals all stored information to M.
- **Test:** M uses this query to obtain either the real session key or a randomly generated string, depending on a randomly selected bit. The adversary then attempts to distinguish the real session key from the random string, thereby evaluating session-key indistinguishability.
- **Accepted mode:** Messages exchanged are considered sequentially ordered, resulting in session-based identification.
- **Participants:** Mutual authentication involves the user, smart device, and gateway.
- **Freshness:** A session key established between two parties Y and Z is considered fresh if it is newly derived for the current session, independent of previous session keys, and has not been exposed to the adversary M hrough any permitted reveal or corruption query.
- **Partnering:** Two protocol instances are considered partnered if both instances (1) enter the accepted mode and (2) share the same session identification.

4.1.7 Session Key Semantic Security

If the T_{test} query returns bb', the attacker M wins the game when $bb' = bb$. Let Suc denote the event that M wins the game. The advantage of M in breaking the session key semantic security of the proposed scheme $\mathcal{Q}$, denoted as $\text{Ad}_{\mathcal{Q}}^{\mathcal{BSDE}-\mathcal{SH}}$, is defined as follows:

$$\text{Ad}_{\mathcal{Q}}^{\mathcal{BSDE}-\mathcal{SH}} = |2 \cdot \Pr[\text{Suc}] - 1| \tag{5}$$

If there exists a negligible function ε such that $\text{Ad}_{\mathcal{Q}}^{\mathcal{BSDE}-\mathcal{SH}} \leq \varepsilon$, then the proposed scheme $\mathcal{Q}$ is considered semantically secure under the ROR model.

The following theorem establishes an upper bound on the adversary's advantage under the Real-Or-Random (ROR) model by relating the security of the proposed BSDE-SH authentication protocol to the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) under the random oracle assumption.

Theorem 1: *Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) adversary interacting with the proposed BSDE-SH protocol in the ROR model within running time t. Assuming that the Elliptic Curve Discrete Logarithm Problem (ECDLP) is computationally intractable over the selected elliptic-curve group and that the hash function behaves as a random oracle, the adversarial advantage in distinguishing the established session key from a uniformly random string satisfies*

$$\text{Adv}_{\mathcal{A}}^{\text{BSDE-SH}}(t) \leq \text{Adv}^{\text{ECDLP}}(t') + \frac{q_h^2}{2^{\lambda+1}}, \tag{6}$$

where q_h denotes the number of hash-oracle queries, λ is the security parameter, and t' denotes the running time required to solve the underlying ECDLP instance. Consequently, if the ECDLP remains computationally infeasible and the hash function behaves as a random oracle, then $\text{Adv}_{\mathcal{A}}^{\text{BSDE-SH}}(t)$ is negligible.

Proof: Assume that a PPT adversary $\mathcal{A}$ can distinguish the established session key from a uniformly random string with non-negligible probability in the ROR experiment.

The proposed BSDE-SH protocol establishes authenticated session keys using elliptic-curve cryptographic parameters together with dynamic pseudo-identities, public keys, and cryptographic hash functions. Consequently, any adversary capable of distinguishing the genuine session key with non-negligible probability would necessarily obtain an advantage against the underlying elliptic-curve computational assumption or successfully predict outputs of the random oracle. Therefore, the adversarial advantage is upper-bounded by the probability of solving the ECDLP together with the birthday-bound collision probability, yielding:

$$\mathrm{Adv}_{\mathcal{A}}^{\text{BSDE-SH}}(t) \leq \mathrm{Adv}^{\text{ECDLP}}(t') + \frac{q_h^2}{2^{\lambda+1}}.$$

The first term represents the probability of solving the underlying ECDLP, while the second term corresponds to the birthday-bound probability associated with random-oracle queries. Since the ECDLP is assumed to be computationally infeasible over the selected elliptic-curve group and the probability of successfully predicting random-oracle outputs is negligible, the overall adversarial advantage is also negligible. Hence, the established session key is semantically secure under the Real-Or-Random (ROR) model.□

4.1.8 Random Oracle

All participants, including the attacker M, have access to a cryptographic hash function $h(\cdot)$, modeled as a random oracle. Here, Hash denotes a query to the random oracle [46].

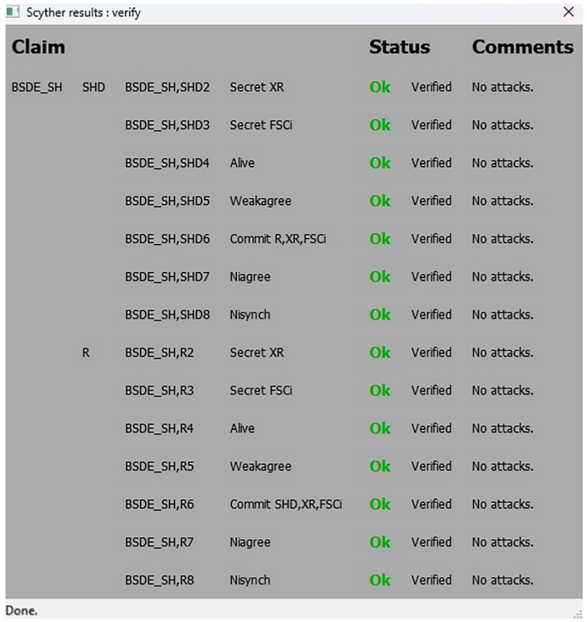
4.2 Scyther Simulation

In this section, we formally verify the proposed **BSDE-SH** protocol using Scyther, a widely used tool for automated security protocol analysis. The protocol is modeled in Security Protocol Description Language (SPDL), defining entity roles and security claims, including secrecy, aliveness, agreement, and synchronization. The analysis was performed on a system with an Intel Core i7 2.8 GHz processor and 16 GB RAM. Under the Dolev–Yao adversary model, Scyther successfully verifies the specified security claims, including aliveness, non-injective synchronization (Nisynch), and non-injective agreement (Niagree), confirming correct message sequencing and agreement among communicating entities [37]. The verification results are presented in Fig. 5a,b.

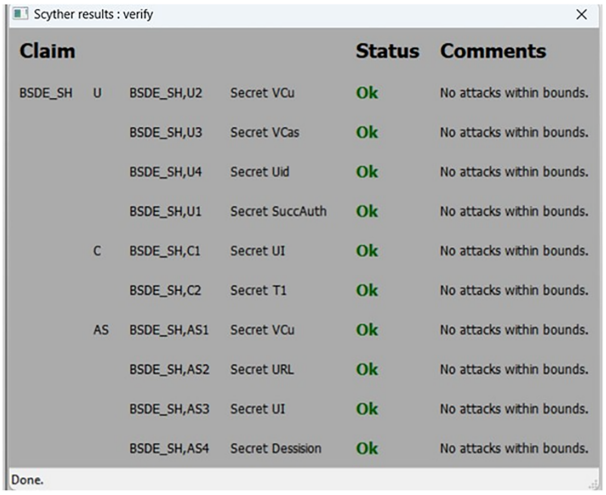
The Scyther analysis verifies all specified authentication and secrecy claims without identifying feasible attack traces under the adopted adversarial model. The successful *Alive*, *Niagree*, and *Nisynch* claims confirm mutual authentication, correct message sequencing, and consistent protocol states, while the secrecy claims ensure session-key confidentiality under the Dolev–Yao model. These results formally support the correctness and security of the proposed BSDE-SH authentication and key establishment protocol.

4.3 Results from Our Smart Home Implementation

This subsection presents simulation results for the Smart Home (SH) implementation, evaluating local device behavior and remote access through the IoT RADIUS server. Remote users can control SH devices after authentication; however, conventional credential-based authentication remains vulnerable to credential compromise and insider attacks [31]. Fig. 6 shows the nine-device SH configuration, with the fan set to "High," and the remote login interface. Among the devices, only the garage door and fan achieved 100% wireless signal strength. These limitations highlight the need for stronger mutual authentication mechanisms to mitigate credential and brute-force attacks [29,48].



a Protocol verification result.



b Entity-role analysis.

Figure 5: Scyther-based formal security verification of the proposed BSDE-SH scheme.

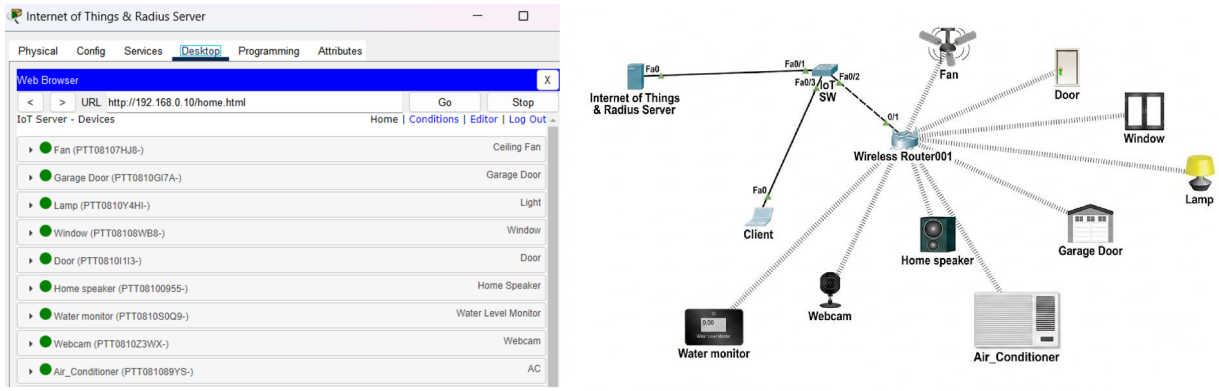


Figure 6: Simulated nine-device smart home network.

Security Assumptions and Features

The proposed BSDE-SH framework assumes a fully trusted Registration Authority (RA) and employs PBFT consensus in the permissioned blockchain for efficient transaction processing [49]. Registration occurs over a secure channel, while authentication uses a public channel fully controlled by an adversary capable of intercepting, modifying, forging, delaying, or deleting messages [46]. Smart and mobile devices may also be physically compromised; hence, sensitive credentials are cryptographically protected.

BSDE-SH provides mutual authentication, secure session key establishment, data integrity, transaction traceability, and secure communication without maintaining a verifier list. Through ECC-based authentication, digital signatures, and blockchain verification, the framework is designed to resist replay, impersonation, man-in-the-middle, message modification, transaction hijacking, birthday, and DoS/DDoS attacks [29].

4.4 MIRACL Experimental Results and Computational Performance

To evaluate the performance of the proposed *BSDE-SH* system on resource-constrained IoT hardware, we employ the Multi-Precision Integer and Rational Arithmetic Cryptographic Library (MIRACL). The experimental setup consists of an HP EliteBook 8460P (2.7 GHz Intel Core i7 CPU, 4 GB RAM) serving as the router station and a Raspberry Pi 3 (Cortex-A53, 1.4 GHz CPU, 1 GB LPDDR2 SDRAM) representing the smart-home device (SHD). Using MIRACL, the measured execution times for bilinear pairing (T_{BP}), point multiplication (T_{PM}), point addition (T_{PA}), random number generation (T_{RN}), and one-way hashing (T_{HS}) were 4.038, 0.926, 0.006, 0.118, and 0.004 ms, respectively, on the router, and 12.520, 4.107, 0.080, 1.185, and 0.006 ms on the SHD. As shown in Table 3, *BSDE-SH* is compared with representative schemes [27,38–40,50,51]. A complete authentication cycle requires $10T_{\mathrm{HS}} + 2T_{\mathrm{RN}} + 4T_{\mathrm{PM}} + 3T_{\mathrm{PA}}$, resulting in an execution time of approximately **11.521 ms**. The results demonstrate that *BSDE-SH* provides competitive computational efficiency while maintaining the required security features.

Table 3: Comparison of computational and communication costs.

Scheme	Type	Router (ms)	Device (ms)	Total (ms)	No. of Msgs.	Comm. Cost (bits)	Gain (%)
[27]	IoT	2.802	16.494	19.296	3	5152	73.11
[38]	IoT	12.744	16.458	29.202	2	3200	7.52
[39]	IoT	3.734	20.707	24.441	2	3040	2.15
[40]	IoT	9.292	49.706	58.998	3	3360	12.90
[50]	IoT	3.826	50.865	54.691	3	3280	10.21
[51]	IoT	0.930	16.631	17.561	5	3072	3.22
Proposed	IoT	**2.006**	**9.515**	**11.521**	3	**2976**	–

4.5 Blockchain-Specific Threat Model and Analysis

Beyond the Dolev–Yao and Canetti–Krawczyk adversary models, BSDE-SH considers blockchain-specific threats to decentralized authentication. Its certificate-based registration, ECC mutual authentication, consortium blockchain, PBFT consensus, and smart-contract authorization provide layered protection against identity forgery, consensus manipulation, and transaction tampering.

Sybil attacks: BSDE-SH mitigates Sybil attacks by requiring each router and smart device to register with the trusted Registration Authority (RA) and obtain a unique ECC credential and authenticated public key, preventing adversaries from creating multiple valid identities.

Eclipse attacks: Authentication requests and transactions are collectively verified by consortium validators through PBFT consensus rather than a single communication path. Thus, isolating an individual node alone cannot manipulate authentication decisions or blockchain records.

Validator collusion and malicious validators: PBFT tolerates up to f Byzantine nodes among $3f + 1$ validators. Hence, a bounded number of colluding or malicious validators cannot approve fraudulent transactions or alter committed records without satisfying the required consensus threshold.

Compromised PBFT leader: The PBFT leader coordinates message ordering but cannot independently validate transactions. Since transactions must complete the prepare and commit phases with the required quorum, compromising the leader alone cannot forge authentication decisions or modify committed records.

4.6 Comparative Analysis of Communication and Network Complexity

We now compare the proposed *BSDE-SH* scheme with existing methods [27,38–40,50,51] in terms of security, functionality, computational cost, and gas consumption.

4.6.1 Communication Cost Evaluation

The communication cost is evaluated based on the sizes of messages exchanged during the mutual authentication phase. The assumed bit lengths for each field are as follows: timestamp (T) is 32 bits, hash digest (SHA-256) is 256 bits, raw uncompressed Elliptic Curve Points (A_{D_n}, B_R) are 512 bits ($x \parallel y$ coordinates on `secp256r1`), identity/pseudo-identity (PID) is 160 bits, certificates (Cert) are 256 bits, digital signature (Dsg) is 512 bits, and session key verifier (SSKey-Ver) is 256 bits.

The authentication procedure is completed by exchanging three messages among the participating entities. The first message Msg_1 consists of transmitting $\text{PID}_{\text{Dn}} + \text{Cert}_{\text{Dn}} + A_{D_n} + \text{Dsg}_{\text{Dn}} + T_1$, where the size of Msg_1 is 160 + 256 + 512 + 512 + 32 = **1472** bits. During the transmission of $Msg_2 = \text{PID}_{\text{R}} + \text{Cert}_{\text{R}} + B_R + \text{SSKey-Ver}_{R-D} + T_2$, 160 + 256 + 512 + 256 + 32 = **1216** bits are transmitted. Finally, for $Msg_3 = \text{ACK}_{D-R} + T_3$, 256 + 32 = **288** bits are transmitted. Hence, the total communication cost during the authentication phase is: 1472 + 1216 + 288 = **2976** bits.

Table 3 summarizes the communication cost and the number of exchanged messages for *BSDE-SH* and the benchmark schemes. While the communication overhead of the proposed scheme is slightly higher due to incorporating full 512-bit raw ECC public points, digital signatures, and certificates for rigorous security assurances, it maintains an optimal message complexity of only three exchanged messages.

Figs. 7 and 8 illustrate the comparison. Fig. 7 shows the number of messages exchanged, indicating that *BSDE-SH* aligns with the most message-efficient schemes. Fig. 8 presents the total communication cost across all evaluated schemes.

It should be noted that the computational load values listed in Table 3 are calculated from the execution times of cryptographic operations measured using the MIRACL library on the specified hardware platforms. Therefore, these values represent the measured computational cost of the protocol implemented under the tested hardware configurations. These results should be considered benchmarks for the evaluated applications and not general statistical estimates.

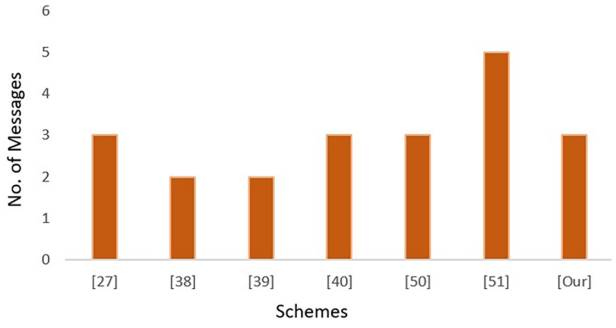


Figure 7: Number of messages exchanged across schemes [27,38–40,50,51].



Figure 8: Total communication cost (bits) across schemes [27,38–40,50,51].

4.6.2 Computational Complexity Evaluation

Let T_{BP}, T_{PM}, T_{PA}, T_{RN}, and T_{HS} denote the execution times of bilinear pairing, point multiplication, point addition, random number generation, and one-way hash operations, respectively. The computational cost of the proposed **BSDE-SH** scheme is $6T_{HS} + T_{RN} + 2T_{PM} + T_{PA}$ for a smart device and $4T_{HS} + T_{RN} + 2T_{PM} + T_{PA}$ for a router. Based on these cryptographic operation costs, the communication packet sizes and transmission overhead are summarized in Table 3, while the overall computational cost comparison with representative schemes [27,38–40,50,51] is also presented in Table 3. The proposed **BSDE-SH** scheme achieves substantially lower computational overhead while maintaining the required security functionality, providing approximate performance improvements of 67.5%, 153.5%, 112.1%, 412.1%, 374.71%, and 52.4% over the respective competing schemes.

Energy Efficiency Discussion

The proposed BSDE-SH protocol is designed for resource-constrained IoT devices by minimizing both computational and communication overhead. The authentication procedure requires only three message exchanges and employs lightweight ECC operations together with cryptographic hash functions, thereby reducing the overall computational workload compared with existing schemes. Since computational complexity is closely related to processor energy consumption, the reduced execution time of the proposed protocol is expected to contribute to improved energy efficiency and prolonged battery lifetime for smart-home devices. Nevertheless, detailed hardware-level power profiling and battery consumption measurements are beyond the scope of the present study and will be investigated in future work.

4.7 Throughput and Scalability Assessment

To evaluate structural sustainability within heavy IoT infrastructure deployments, network scaling evaluations were performed using variable device bounds. Unlike traditional blockchain architectures, where transaction validation can increase the communication and consensus burden as the number of participating nodes grows, our framework delegates state processing to fixed validator edge configurations operating the lightweight PBFT consensus structure. By relying on a deterministic $3f + 1$ fault-tolerance quorum at the routing layer instead of open mining cycles, the message validation latency remains predictable and decoupled from the total number of connected end-user IoT hardware entities. When tracking performance under loads spanning 10 devices, 100 devices, and up to 1000 parallel communicating smart home entities, message validation loops remain highly predictable.

Scalability Proof: Let N be the total pool of connected smart home devices ($10 \leq N \leq 1000$) and R be the fixed number of edge routers acting as PBFT validation consensus nodes, where $R = 3f + 1$. The message complexity of PBFT consensus scales as $\mathcal{O}(R^2)$ with respect to the number of validator nodes.

In the proposed architecture, a device D_n initiates transaction verification directly with its assigned edge router via a single unicast request. The consensus overhead is therefore confined to the router tier, yielding a message complexity of $\mathcal{O}(R^2)$ for the PBFT validation process. Because $R \ll N$ and R remains constant regardless of whether $N = 10$, 100, or 1000, the total network verification overhead can be expressed as: $\mathcal{O}(N) + \mathcal{O}(R^2) \approx \mathcal{O}(N)$. Thus, when the number of validators (R) is fixed, the overall verification complexity grows approximately linearly with the network size (N).

Furthermore, the PBFT communication complexity is confined to the fixed validator router layer rather than the entire IoT network. Although PBFT requires $\mathcal{O}(R^2)$ message exchanges among validator routers, the value of R remains constant in the proposed architecture. Consequently, increasing the number of connected smart-home devices affects only the request-processing workload, while the consensus overhead remains bounded. This analytical result confirms that the proposed framework exhibits improved scalability under increasing network sizes while preserving the security and fault-tolerance properties of the PBFT consensus mechanism.

4.8 Smart Contract Execution and Gas Cost Evaluation

The proposed *BSDE-SH* scheme is evaluated by deploying its smart contract primitives on a local Ganache Ethereum simulator environment. All core smart contract methods (`AddSHD`, `RemoveSHD`, `AddRouter`, and `RemoveRouter`) were executed 20 times each to calculate average deterministic operational gas costs with statistical rigor. The average gas cost per method is calculated to assess ongoing performance, as illustrated in Fig. 9a.

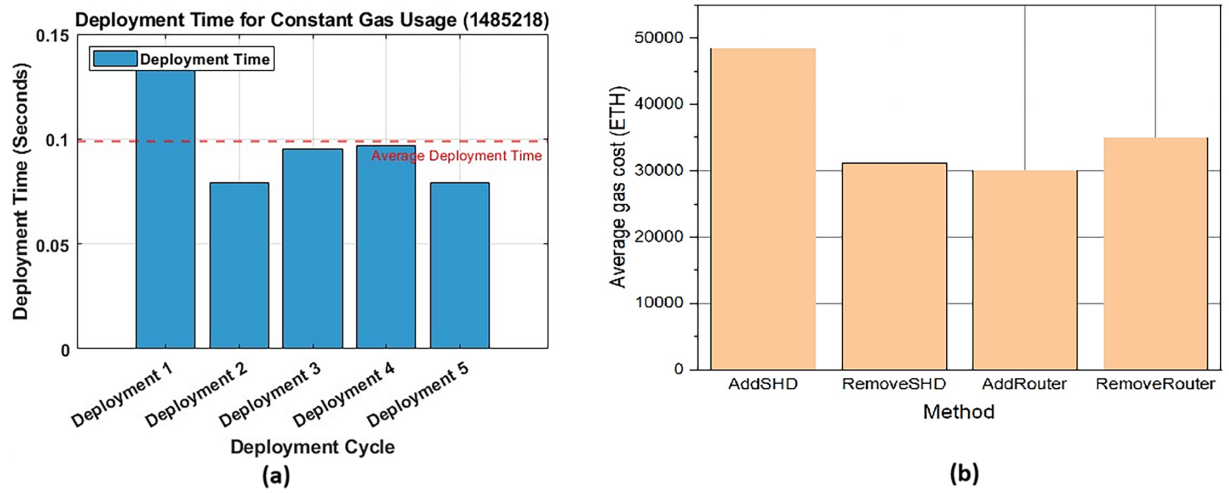


Figure 9: Smart contract execution metrics, deployment cycle history, and transaction verification logs. (**a**) deployment time across five deployment cycles under constant gas usage (1,485,218), with the dashed line indicating the average deployment time; and (**b**) average gas consumption (in ETH gas units) for the AddSHD, RemoveSHD, AddRouter, and RemoveRouter smart-contract operations.

To provide clear individual metrics, the one-time contract deployment gas cost is explicitly separated from ongoing per-function operational gas costs. Using Solidity v0.8.0 on a local Ganache network, the smart contract incurred a one-time deployment cost of 1,485,218 gas units (evaluated over five consecutive deployment loops with an average deployment time of 0.063 s), as shown in Fig. 9b.

Distinct from the initial deployment cost, the function execution layout achieves an explicit **80.4% reduction in operational gas costs** by removing redundant tracking parameters and migrating states off

public mainnets. This optimization proves that standard legacy non-optimized public contracts are up to 412% more expensive in per-call operational gas than the proposed architecture.

4.9 Smart Contract Security Analysis

In addition to execution cost, the Ethereum-compatible smart contract was analyzed against common blockchain threats. Role-based access control restricts administrative functions, such as device registration and router management, to authenticated and authorized entities, preventing unauthorized state modifications. Since the contract performs no external callbacks or recursive calls during state updates, exposure to classical reentrancy attacks is minimized. Authentication transactions incorporate timestamps, cryptographic signatures, and dynamic parameters to mitigate replay and transaction-manipulation attempts. Moreover, bounded function complexity and the absence of unbounded on-chain loops reduce the risk of gas-exhaustion-based denial-of-service attacks. These smart-contract security considerations complement, but are distinct from, the formal protocol analysis of properties such as mutual authentication and session-key confidentiality.

4.10 Blockchain Storage Overhead Analysis

The proposed BSDE-SH framework stores only essential authentication records, transaction metadata, and cryptographic verification data on-chain, while application-specific smart-home data remain off-chain. Thus, ledger growth depends mainly on authentication transactions rather than IoT data volume. Moreover, PBFT consensus and blockchain replication are restricted to permissioned validator routers, eliminating the need for resource-constrained devices to maintain complete ledger copies. This design reduces storage and synchronization overhead and improves long-term scalability for large-scale smart-home deployments.

4.11 Comparative Analysis of Security and Functionality Features

Table 4 compares the functionality and security features (FSF_1–FSF_{10}) of the proposed *BSDE-SH* protocol with other comparable schemes. The *BSDE-SH* protocol demonstrates superior performance in both security and functionality, offering comprehensive defensive capabilities against targeted attack configurations (e.g., Sybil and spoofing attempts) not supported by existing methods.

Table 4: Comparative functionality and security features among schemes.

Feature	[27]	[38]	[39]	[40]	[50]	[51]	Our
Security verification (Scyther/AVISPA) (FSF_1)	X	X	X	✓	✓	X	✓
Blockchain-based solution (FSF_2)	X	X	✓	X	X	X	✓
GPS attack (FSF_3)	X	X	X	X	X	X	✓
Sybil attack (FSF_4)	X	X	X	X	X	X	✓
Session key security (FSF_5)	✓	✓	✓	✓	✓	✓	✓
Mutual authentication (FSF_6)	✓	✓	✓	✓	✓	✓	✓
Impersonation attack on server (FSF_7)	X	✓	X	✓	✓	✓	✓
Impersonation attack on device/client (FSF_8)	✓	X	X	X	✓	✓	✓
Man-in-the-middle attack (FSF_9)	X	✓	X	✓	✓	✓	✓
Replay attack (FSF_{10})	✓	✓	✓	✓	✓	✓	✓

5 Conclusion

This paper addresses single points of failure and node-level attacks in smart-home (SH) IoT networks through the proposed ECC-based *BSDE-SH* framework. The protocol is validated using a nine-device SH simulation, Scyther-based formal security analysis, and MIRACL-based performance evaluation. The results demonstrate strong security properties with competitive communication and computational efficiency compared with existing schemes.

Although ECC provides strong security against classical adversaries, large-scale quantum computers running Shor's algorithm may compromise its long-term security. Therefore, the modular design of BSDE-SH facilitates future migration toward NIST-standardized post-quantum cryptography (PQC), such as ML-KEM for key establishment and ML-DSA for digital signatures, with appropriate modifications to the protocol operations and message structures. Future work will evaluate network-level performance under packet loss, latency, congestion, and node failures; optimize Solidity smart contracts for reduced gas consumption; and investigate hardware-level energy consumption and implementation on resource-constrained IoT devices. We also plan to explore XAI integration and extend the decentralized framework to smart-city applications, including municipal resource coordination, intersection scheduling, and traffic congestion management.

Acknowledgement: The authors used the Grammarly Premium tool to proofread the article for spelling, grammar, and punctuation and to improve overall clarity.

Funding Statement: This work of was supported by the Abu Dhabi University's Office of Research and Sponsored Programs under Grant 19301112.

Author Contributions: Idris Adedamola Abdulhameed: Writing original draft, Conceptualization, Experimentation. Sulaiman M. Karim: Design, Analysis, Experimentation, Reviewing original and revised draft. Musaria K. Mahmood: Supervision, Conceptualization, Reviewing original draft. Shehzad Ashraf Chaudhry: Reviewing and revising original and revised drafts, Performance Analysis and Comparisons, funding acquisition. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The complete software implementation suite, smart contract source code, smart contract configuration files, Cisco Packet Tracer simulation configuration files, Scyther protocol specification and verification files, MIRACL benchmark timing data, performance logging scripts, and other supporting materials necessary to reproduce the findings of this study are openly available in the public GitHub repository at: https://github.com/idris0001/blockchain-smart-home-auth.

Ethics Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Robles RJ, Kim TH. A review on security in smart home development. Int J Adv Sci Technol. 2010;15:13–22.
2. World population ageing 2019. 2019 [cited 2026 Aug 26]. Available from: https://digitallibrary.un.org/record/3907988/files/WorldPopulationAgeing2019-Report.pdf.
3. Park H, Lee S, Moon E, Kim D. Cooperative DDNS based home IoT platform using home gateway in dynamic IP address environment. Int Inf Inst. 2017;20(9A):6461–8.
4. Mishra A, Ghayar J, Pendam R, Shinde S. Design and implementation of smart home network using Cisco packet tracer. In: Proceedings of the 2022 International Conference on Automation, Computing and Communication; 2022 Apr 7–8; Navi Mumbai, India. p. 1–6.

5. Shemsi I. Implementing smart home using Cisco packet tracer simulator. Int J Eng Sci Invent Res Dev. 2018;4(VII):261–9.
6. Aziz NNA, Rechie RRMA, Bakry BBM, Rahman RAB, Yussoff YM. Analysing smart home security using packet tracer simulation software. In: Proceedings of the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE); 2021 Apr 3–4; Penang, Malaysia. p. 239–44.
7. Alturki N, Alharthi R, Umer M, Saidani O, Alshardan A, Alhebshi RM, et al. Efficient and secure IoT-based smart home automation using multi-model learning and blockchain technology. Comput Model Eng Sci. 2024;139(3):3387–415. doi:10.32604/cmes.2023.044700.
8. Kouanou AT, Tchapga CT, Ekonde MS, Monthe V, Mezatio BA, Manga J, et al. Securing data in an internet of things network using blockchain technology: smart home case. SN Comput Sci. 2022;3(2):1–10. doi:10.1007/s42979-022-01065-5.
9. Dang CT, Aslan I, Lingenfelser F, Baur T, André E. Towards somaesthetic smart home designs: exploring potentials and limitations of an affective mirror. In: Proceedings of the 9th International Conference on the Internet of Things (IoT 2019); 2019 Oct 22–25; Bilbao, Spain. p. 1–8.
10. Jadon S, Choudhary A, Saini H, Dua U, Sharma N, Kaushik I. Comfy smart home using IoT. In: Proceedings of the International Conference on Innovative Computing and Communications (ICICC 2020); 2020 Feb 21–23; New Delhi, India. p. 1–8.
11. Singh S, Ra IH, Meng W, Kaur M, Cho GH. SH-BlockCC: a secure and efficient Internet of things smart home architecture based on cloud computing and blockchain technology. Int J Distrib Sensor Netw. 2019;15(4):1–18.
12. Alam MJ, Hossain I, Puppala S, Talukder S. Combating identity attacks in online social networks: a multi-layered framework using zero-knowledge proof and permissioned blockchain. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2023); 2023 Nov 6–9; Kuşadası, Turkey. p. 636–43.
13. Sasi T, Lashkari AH, Lu R, Xiong P, Iqbal S. A comprehensive survey on IoT attacks: taxonomy, detection mechanisms and challenges. J Inf Intell. 2024;2(6):455–513.
14. Kim Y, Oh H, Kang S. Proof of concept of home IoT connected vehicles. Sensors. 2017;17(6):1289. doi:10.3390/s17061289.
15. Haber S, Stornetta WS. How to time-stamp a digital document. J Cryptol. 1991;3:99–111.
16. Ismail L, Hameed H, AlShamsi M, AlHammadi M, AlDhanhani N. Towards a blockchain deployment at UAE University: performance evaluation and blockchain taxonomy. In: Proceedings of the 2019 International Conference on Blockchain Technology (ICBCT 2019); 2019 Mar 15–18; Honolulu, HI, USA. p. 30–8.
17. Karim SM, Habbal A, Chaudhry SA, Irshad A. Architecture, protocols, and security in IoV: taxonomy, analysis, challenges, and solutions. Secur Commun Netw. 2022;2022:1–19. doi:10.1155/2022/1131479.
18. Sabir BE, Youssfi M, Bouattane O, Allali H. Towards a new model to secure IoT-based smart home mobile agents using blockchain technology. Eng Technol Appl Sci Res. 2020;10(2):5441–7. doi:10.48084/etasr.3394.
19. Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inf Theory. 1983;29(2):198–208. doi:10.1109/tit.1983.1056650.
20. Qashlan A, Nanda P, He X, Mohanty M. Privacy-preserving mechanism in smart home using blockchain. IEEE Access. 2021;9:103651–69. doi:10.1109/access.2021.3098795.
21. Meyer D, Haase J, Eckert M, Klauer B. A threat-model for building and home automation. In: Proceedings of the 14th IEEE International Conference on Industrial Informatics (INDIN 2016); 2016 Jul 19–21; Poitiers, France. p. 860–6.
22. Wazid M, Das AK, Odelu V, Kumar N, Susilo W. Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans Dependable Secur Comput. 2017;17(2):391–406. doi:10.1109/tdsc.2017.2764083.
23. Poh GS, Gope P, Ning J. PrivHome: privacy-preserving authenticated communication in smart home environment. IEEE Trans Dependable Secur Comput. 2021;18(3):1095–107. doi:10.1109/tdsc.2019.2914911.
24. Naoui S, Elhdhili ME, Saidane LA. Lightweight and secure password based smart home authentication protocol: LSP-SHAP. J Netw Syst Manage. 2019;27(4):1020–42. doi:10.1007/s10922-019-09496-x.

25. Sadhukhan D, Ray S, Biswas GP, Khan MK, Dasgupta M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. J Supercomput. 2021;77(2):1114–51. doi:10.1007/s11227-020-03318-7.
26. Satapathy U, Mohanta BK, Jena D, Sobhanayak S. An ECC based lightweight authentication protocol for mobile phone in smart home. In: Proceedings of the 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS); 2018 Dec 1–2; Rupnagar, India. p. 303–8.
27. Zou S, Cao Q, Wang C, Huang Z, Xu G. A robust two-factor user authentication scheme-based ECC for smart home in IoT. IEEE Syst J. 2021;16(3):4938–49. doi:10.1109/jsyst.2021.3127438.
28. Sankar SMU, Dhinakaran D, Kavya T, Priyanka S, Oviya P, Pugal P. A way for smart home technology for disabled and elderly people. In: Proceedings of the International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2023); 2023 Mar 14–16; Dehradun, India. p. 369–73.
29. Lin C, He D, Kumar N, Huang X, Vijayakumar P, Choo KR. HomeChain: a blockchain-based secure mutual authentication system for smart homes. IEEE Internet Things J. 2019;7(2):818–29.
30. Wu L, Du X, Wang W, Lin B. An out-of-band authentication scheme for internet of things using blockchain technology. In: Proceedings of the International Conference on Computing, Networking and Communications (ICNC 2018); 2018 Mar 5–8; Maui, HI, USA. p. 769–73.
31. Yang L, Liu X, Gong W. Secure smart home systems: a blockchain perspective. In: Proceedings of the 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA); 2020 Apr 10–13; Chengdu, China. p. 1003–8.
32. Liao K. Design of the secure smart home system based on the blockchain and cloud service. Wirel Commun Mob Comput. 2022;2022:4393314.
33. Baucas MJ, Gadsden SA, Spachos P. IoT-based smart home device monitor using private blockchain technology and localization. IEEE Netw Lett. 2021;3(2):52–5. doi:10.1109/lnet.2021.3070270.
34. Liu B, Yao X, Guo K, Zhu P. Consortium blockchain based lightweight message authentication and auditing in smart home. IEEE Access. 2023;11:68473–85. doi:10.1109/access.2023.3293401.
35. Xue J, Xu C, Zhang Y. Private blockchain-based secure access control for smart home systems. KSII Trans Internet Inf Syst. 2018;12(12):6057–78. doi:10.3837/tiis.2018.12.024.
36. Ammi M, Alarabi S, Benkhelifa E. Customized blockchain-based architecture for secure smart home for lightweight IoT. Inf Process Manage. 2021;58(3):102482. doi:10.1016/j.ipm.2020.102482.
37. Cherbal S. Blockchain-enabled authentication and key exchange for enhancing user privacy and data confidentiality in smart homes. In: Proceedings of the 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS 2024); 2024 Apr 24–25; El Oued, Algeria. p. 1–7.
38. Jia X, He D, Kumar N, Choo KKR. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. IEEE Syst J. 2019;14(1):560–71.
39. Wang J, Wu L, Choo KKR, He D. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. IEEE Trans Ind Inf. 2019;16(3):1984–92. doi:10.1109/tii.2019.2936278.
40. Wazid M, Das AK, Kumari S, Li X, Wu F. Provably secure biometric-based user authentication and key agreement scheme in cloud computing. Secur Commun Netw. 2016;9(17):4103–19. doi:10.1002/sec.1591.
41. Subhita S, Divya D, Kavita K. Scalable and secure architecture for smart home using blockchain with data analysis for security risk. In: Proceedings of the 2023 IEEE 8th International Conference for Convergence in Technology (I2CT); 2023 Apr 7–9; Lonavla, India. p. 1–8.
42. Rana A, Sharma S, Nisar K, Ibrahim AG, Dhawan S, Chowdhry B, et al. The rise of blockchain internet of things (BIoT): secured, device-to-device architecture and simulation scenarios. Appl Sci. 2022;12(15):7694.
43. Alwabli A. Federated learning for privacy-preserving air quality forecasting using IoT sensors. Eng Technol Appl Sci Res. 2024;14(4):15124–31. doi:10.48084/etasr.7820.
44. Zhang Q. Federated learning for breaking data silos in smart governance: a privacy-preserving framework for cross-agency collaboration. Future Technol. 2026;5(2):179–88. doi:10.55670/fpll.futech.5.2.17.
45. Ghadi YY, Mazhar T, Shah SFA, Haq I, Ahmad W, Ouahada K, et al. Integration of federated learning with IoT for smart cities applications, challenges, and solutions. PeerJ Comput Sci. 2023;9:e1657. doi:10.7717/peerj-cs.1657.

46. Guo Y, Zhang Z, Guo Y. SecFHome: secure remote authentication in fog-enabled smart home environment. Comput Netw. 2022;207:108818.
47. Nyangaresi VO. ECC based authentication scheme for smart homes. In: Proceedings of the 2021 International Symposium ELMAR; 2021 Sep 13–15; Zadar, Croatia. p. 5–10.
48. Karim SM, Habbal A, Chaudhry SA, Irshad A. BSDCE-IoV: blockchain-based secure data collection and exchange scheme for IoV in 5G environment. IEEE Access. 2023;11:36158–75. doi:10.1109/access.2023.3265959.
49. Castro M, Liskov B. Practical Byzantine Fault Tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI'99); 1999 Feb 22–25; New Orleans, LA, USA. p. 173–86.
50. Patil RY, Karati A, Patil YH. A signcryption with identity-based authentication for secure EHR sharing in IoMT utilizing ECC. Int J Inf Technol. 2024;16(8):5133–48. doi:10.1007/s41870-024-01963-2.
51. Aswathy RH, Malarvizhi N. A design of lightweight ECC based cryptographic algorithm coupled with linear congruential method for resource constraint area in IoT. J Am Intell Humaniz Comput. 2023;14(5):5097–106. doi:10.1007/s12652-020-02788-0.