Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025660
Article

Tech Science Press

Blockchain-Based Robust Data Security Scheme in IoT-Enabled Smart Home

Anusha Vangala[1], Ashok Kumar Das[1], YoungHo Park[2,*] and Sajjad Shaukat Jamal[3]

[1]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, 500032, India
[2]School of Electronics Engineering, Kyungpook National University, Daegu, 41566, Korea
[3]Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia
*Corresponding Author: YoungHo Park. Email: parkyh@knu.ac.kr
Received: 01 December 2021; Accepted: 09 February 2022

Abstract: The recent surge in development of smart homes and smart cities can be observed in many developed countries. While the idea to control devices that are in home (embedded with the Internet of Things (IoT) smart devices) by the user who is outside the home might sound fancy, but it comes with a lot of potential threats. There can be many attackers who will be trying to take advantage of this. So, there is a need for designing a secure scheme which will be able to distinguish among genuine/authorized users of the system and attackers. And knowing about the details of when and what IoT devices are used by the user, the attacker can trace the daily activities of user and can plan an attack accordingly. Thus, the designed security scheme should guarantee confidentiality, anonymity and un-traceability. Most of the schemes proposed in the literature are either non-blockchain based which involves inherent problems of storing data in a single-server or assuming weaker attack models. In this work, we propose a novel scheme based on blockchain technology, assuming a stronger Canetti and Krawczyk (CK)-threat model. Through the formal and informal security, and comparative analysis, we show that the proposed scheme provides a superior security and more functionality features, with less communication cost and comparable computational cost as compared to other competent schemes. Moreover, the blockchain based simulation study on the proposed scheme has been conducted to show its feasibility in real-life application.

Keywords: Internet of things (IoT); smart home; ubiquitous computing; blockchain; security

1 Introduction

Now a days, almost all the utilities are enabled with the Internet connectivity. It benefits users of multiple domains to operate the resources remotely and know their status. Smart home is one such application of the Internet of Things (IoT), where the users can monitor the home appliances and can take decisions accordingly. The system helps in tackling numerous real-life problems, like robberies (by

collecting the data from surveillance) and fire accidents (by collecting data from fire detecting sensors) and so on. However, it brings in several challenges which are to be handled and so, there is a need for designing a security scheme very carefully by considering all such challenges and making sure that no sensitive information is compromised.

Since there are many schemes in the literature which ensure secure communication between home gateway and IoT smart devices, we confine ourselves in designing the secure scheme which is subjected to the security aspects/hurdles that either of a user or the home gateway faces when they try to communicate with each other. The blockchain technology has been employed in order to enhance the security of a system. The blockchain has several inherent properties like decentralization, immutability and transparency. Thus, the blockchain technology provides high data integrity, and resilience against Denial-of-Service attacks (DoS) and several cyber security attacks when the data is simply stored in centralized databases. The immutability of the blockchain allows no adversary can update a block, insert fake information into a block or even delete a block once a block is added into the blockchain after running a consensus protocol among a group of nodes, called peer nodes, in a distributed Peer-to-Peer (P2P) blockchain network. To mitigate these issues, we aim to proposed a novel blockchain-based robust security scheme in an IoT-enabled smart home environment, which is light-weight by making the authentication and key establishment between the two authorized parties with minimal resources faster, and it will make a user in smart home deployment to feel that all the resources are at his disposal.

1.1 Network Model

A blockchain-based smart home system is shown in Fig. 1. In this system, we have a trusted registration authority (RA), which takes the responsibility of registering all the authorized entities by providing the corresponding secret parameters of that entity in a secure channel. We maintain a gateway node which acts as a mediator between the network of devices at home and remote users operating through some smart devices with smart cards installed in it. Once the home gateway and the users register with the RA, the users will be installing a smart card in their device, which has some set of parameters (credentials) through which their secrets are hidden. A set of consensus nodes are maintained on a cloud that are responsible for verifying the blocks and adding the blocks into the blockchain stored using these cloud nodes.

1.2 Attack Model

The proposed scheme is evaluated under "Dolev-Yao (DY) threat model" [1], which considers an adversary has capabilities of modifying, replaying, dropping the transactions and requests in transit. The adversary can also impersonate the authorized user(s) and home gateway. The smart devices of the devices with installed smart cards can be lost and all the information from them can be extracted by the adversary. The scheme also adopts the "Canetti and Krawczyk (CK-adversary model)" [2], where the adversary has all the capabilities as in the DY threat model, and in addition, the session state information and previous session keys of the entities in the system can be leaked through the session hijacking attack. The home gateway node can be put under a locking system as in [3] in order to avoid physical capture attack by the adversary. In addition, some smart home IoT devices can be physically capture by the adversary as these devices cannot be always monitored in 24×7 time. The adversary can then extract all the credentials stored in a captured smart device using the power analysis attacks [4].

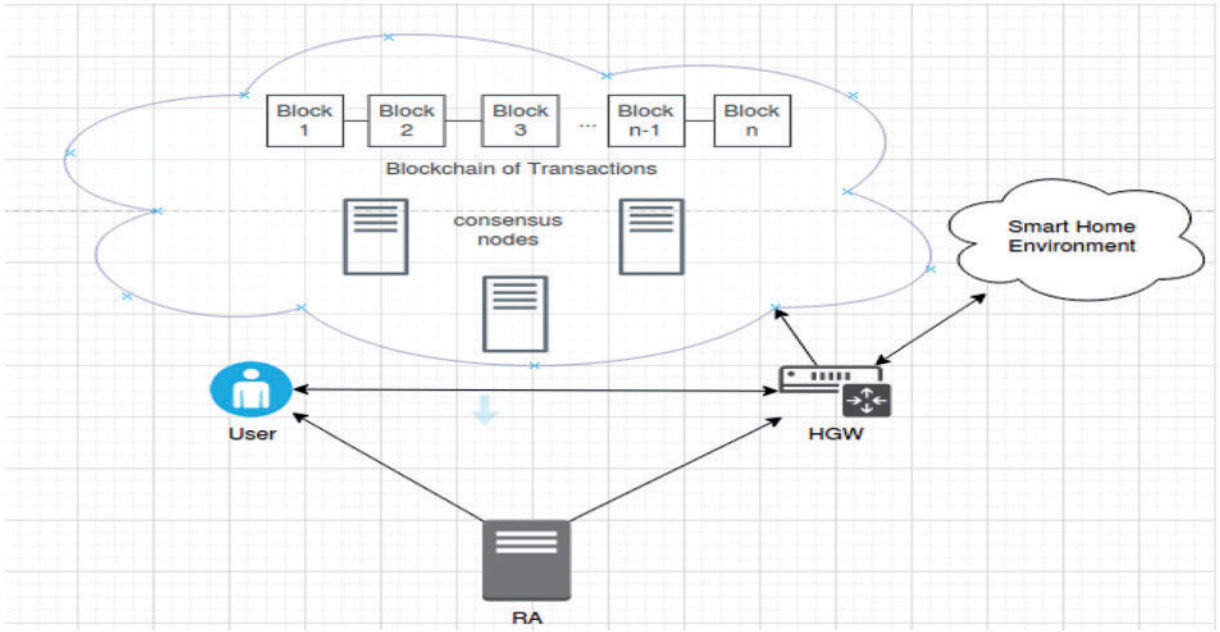

Figure 1: A general IoT network model of blockchain based smart home system

1.3 Research Contributions

In the following, some important contributions made in the research paper are listed:

- The proposed scheme is light-weight, which makes the authentication and key establishment between the two authorized parties with minimal resources faster, and it will make a user to feel that all the resources are at his disposal. The proposed scheme makes sure that the anonymity of a user is never compromised. In addition, the blockchain technology has been adopted with the scheme to make it highly robust for data storage in P2P blockchain network to provide high data integrity, and resilience against DoS and several cyber security attacks.
- The proposed scheme handles the challenging and risky smart card stolen attacks, and thus, a user needs not to be panic if he lost his smart card. The scheme is designed in such a way that it will resist many attacks, like *stolen smart card attack, user impersonation attack, replay attack, gateway impersonation attack, man-in-the-middle attack, and Ephemeral Secret Leakage (ESL) attack,* which make the scheme robust. We also consider a stronger notion of user anonymity where even gateway will not know the user's actual identity.
- We show how the proposed scheme is resistant to various potential attacks under the DY and CK-adversary models. A detailed comparative analysis among the proposed scheme and other existing schemes show that the proposed scheme provides a superior security and more functionality features, with less communication cost and comparable computational cost as compared to other competent existing schemes in the literature.
- The practical implementation of the proposed scheme using the blockchain-based experimental study has been shown to exhibit the feasibility in real-life application.

1.4 Paper Outline

The remainder of this article is as follows. Section 2 describes an in-depth literature study on the related authentication protocols in an IoT environment. Various phases related to the proposed scheme have been discussed in Section 3. A detailed security analysis using both formal and informal security study has been done in Section 4 to show the robustness of the proposed scheme against various types of attacks in an IoT-enabled smart home environment. Section 5 provides a detailed comparative study on various security and functionality features, and communication and computational costs among the proposed scheme and other existing schemes. A blockchain-based implementation on the proposed scheme has been shown in Section 6. Finally, Section 7 provides some concluding remarks on this article.

2 Literature Survey

There has been extensive research on authentication schemes for smart home systems in the last decade. We study some of the most recent schemes relevant to the model developed in our proposed system.

Fakroon et al. [5] proposed an authentication scheme for a smart home system with smart devices to be accessed by users via gateway nodes. This scheme makes use of the user's location that restricts users beyond distance from accessing the smart devices by considering it a malicious access. This scheme has low computation cost, but higher communication cost. It does not consider usage of blockchain.

Shuai et al. [6] proposed an authentication scheme based on elliptic curve cryptography for smart home environment. This scheme is vulnerable to privileged insider attack that can be launched using stolen smart card and offline password guessing attack, user impersonation attack, parallel session attack and password change attack. This scheme was improved by Banerjee et al. [7] to overcome its limitations by considering a stronger notion of threat model, called Canetti and Krawczyk model (CK-adversary model), where the session secret leakage is possible. They adopted smart card-based authentication for user and assumed a central registry to supply secrets to the gateway and the users. This scheme is not block-chain based, and hence, it has all inherent problems of storing the transactions in a single server and also this scheme does not provide anonymity of user at the gateway node. However, this scheme is secure against several attacks such as ESL attack and stolen smart card attack. The same scheme by Shuai et al. [6] was also cryptanalyzed by Kaur et al. [8] by exposing its vulnerability to insecure session key agreement, replay attack, and gateway node bypass attack in addition to the attacks identified by Banerjee et al. [7]. The scheme by Shuai was also considered to be vulnerable against offline dictionary attack and it lacks forward secrecy as identified by Xu et al. [9].

Lin et al. [10] proposed a block-chain based mutual authentication scheme for smart home, where authentication of the users is done by using group signatures and session key establishment between user and the gateway with the help of the elliptic curve integrated encryption scheme (ECIES), and the gateway verification is done using keyed message authentication code (MAC) function generated using ECIES scheme. They confined the scope of the work to secure communication between a user and the gateway, and the same between the gateway and IoT devices are declared beyond the scope of their work. The attack model considered in their work is something like the DY threat model and it does not consider the scope for session secret leakage. Hence, their scheme is vulnerable to ESL attack under the CK-adversary model.

Naoui et al. [11] proposed a user authentication scheme, called LRP-SHAP, for smart home system with a user having a smartphone in a remote environment accessing the smart devices in the smart home via home energy manager (HEM). It uses a Wi-Fi or 4G/5G network for communication. It applies the elliptic curve and hashed message authentication code (HMAC) cryptographic operations in addition to hash functions and cookies for the authentication purpose. However, it has the extra overhead of storing cookies in addition to the parameters required for authentication.

From the above discussion, it is clear that most of the existing schemes are either vulnerable to various attacks including the ESL attack under the CK-adversary model or they are expensive in communication/computational costs. Thus, most of the schemes used in the literature for smart homes consider somewhat weaker notion of security and they are prone to ESL attack under the CK-adversary model, while some of the recent works consider the CK-adversary model but they are either non-blockchain based approaches which will have an inherent disadvantage of storing transactions in a single server or there is no stronger notion of anonymity of user (in the gateway). Moreover, the discussed existing schemes do not adopt the blockchain technology. As a result, the existing schemes lack in providing high data integrity, and they are not resilience against DoS and several cyber security attacks.

To deal with this, we propose a security scheme which can solve these major issues, by adopting the prominent solutions of existing schemes as building blocks with the blockchain technology. The proposed scheme is based on block chain technology and assumes a stronger CK-adversary model. It makes a user anonymous even to the gateway, anonymous and un-traceable to the intruder, and it is also resistant to several attacks for smart home environment.

3 The Proposed Scheme

In this section, we first discuss the main motivation behind the proposal of our blockchain-based security scheme for smart home environment. We then discuss different phases relevant to the proposed scheme.

3.1 Notations

Various notations and their significance are provided in Tab. 1, which are used throughout the paper.

Table 1: Notations and their significance

Notation	Significance
q	A large prime number (at least 160 bits prime)
GF(q)	Galois (finite) field of prime order q
$E_q(a, b)$	A non-singular elliptic curve over GF(q) of the form: $y^2 = x^3 + ax + b \ (mod\ q)$ with $4a^3 + 27b^2 \neq 0 \ (mod\ q)$
G	A base point $E_q(a,b)$ whose order is n_G as big as q
$\cdot$	Elliptic curve point multiplication operation
$x \cdot G$	Elliptic curve point (scalar) multiplication: $x \bullet G = G + G + \cdots + G$ (x times)
$A + B$	Elliptic curve point addition of points $A, B \in E_q(a, b)$

(Continued)

Table 1: Continued

Notation	Significance
RA	Trusted registration authority
U	A user in smart home
HGW	Home gateway
RTS_X	Registration timestamp issued by the RA to an entity
AID_{RA}, AID_U, AID_{HGW}	Actual identities of RA, U and HGW, respectively
$PRID_{RA}$, $PRID_U$, $PRID_{HGW}$	Pseudo-random identities of RA, U and HGW, respectively
SK_{RA}, SK_U, SK_{HGW}	Secret keys of RA, U and HGW, respectively
SID_U, Pwd_U, bio_U	Temporal session identity, secret password and biometric template of U, respectively
pr_U, Pub_U	Private and public keys of U, respectively; $Pub_U = pr_U \bullet G$
pr_{HGW}, Pub_{HGW}, LTS_{HGW}	Private key, public key and long-term secret of HGW, respectively
r_U, α_U	User U's random secrets
$\|\|$	Concatenation operation
$*$	Ordinary multiplication in finite field $GF(q)$
$\oplus$	Exclusive-OR operation
T_U, T_G, T_V	Current timestamp generated by the entities
ΔT	Maximum transmission delay associated with a message
$H(\cdot)$	A collision-resistant one-way cryptographic hash function
$Gen(\cdot)$	Fuzzy extractor probabilistic (randomized) generation function
$Rep(\cdot)$	Fuzzy extractor deterministic reproduction function
σ_U, τ_U	Biometric secret key and public parameter associated with U's biometrics bio_U, respectively
et	An associated error tolerance threshold value used in $Rep(\cdot)$

3.2 Description of Various Phases

In the following subsection, we now discuss the details of the various phases involved in the proposed blockchain-based security scheme for ubiquitous computing environment, called BlockUbiHome.

3.2.1 Setup Phase

In this phase, the trusted RA sets up the system by generating various public and private parameters by executing the below steps:

- **Step 1.** The RA selects a non-singular elliptic curve $E_q(a, b)$ of the form: $y^2 = x^3 + ax + b$ over a finite (Galois) field $GF(q)$, where a and b are constants chosen from $Z_q = \{0, 1, 2, \ldots, q-1\}$, $4a^3 + 27b^2 \neq 0 \pmod q$ and q is a large prime such that the "Elliptic Curve Discrete Logarithm Problem (ECDLP)" is intractable, a base point G in $E_q(a, b)$ and a "collision-resistant one-way hash function" $H(.)$.
- **Step 2.** The RA then randomly picks its actual identity $\mathrm{AID_{RA}}$ and a unique secret key ($\mathrm{SK_{RA}}$) from $\boldsymbol{Z}_{\boldsymbol{q}}^{*} = \{1, 2, \ldots, \boldsymbol{q} - 1\}$. The RA generates a pseudo-random identity ($\mathrm{PRID_{RA}}$) = H($\mathrm{AID_{RA}}$

|| SK_{RA}) and makes it available to all participating entities in the system. The also makes the public information $\{E_q(a, b), G, H(\bullet)\}$ as public.

3.2.2 One-time Registration Phase

This phase is executed by the RA before the actual communication procedure with both gateway node and the user.

1) User registration phase: The user registration phase occurs as follows.

- **Step 1.** A user U selects an actual identity (AID_u), a secret key SK_U, a password Pwd_U, temporal identity SID_U and registration timestamp RTS_U, and calculates pseudo-random identity of the user U as $PRID_U = H(AID_u || SK_U || Pwd_U|| RTS_U)$ and sends the credentials $\{PRID_U, SID_U\}$ to the RA via secure channel.
- **Step 2.** After receiving the credentials of the user U, the RA computes a secret $X_{RA} = H(SID_U || PRID_U || SK_{RA} || AID_{RA})$, stores it in a smart card and returns the smart card to the user U via secure channel.
- **Step 3.** After receiving the smart card from the *RA*, the user U enters personal biometric bio_U at the sensor of a particular terminal and obtains (σ_U, τ_U) using the "fuzzy extractor probabilistic generation function *Gen(.)*", that is, $Gen(bio_U) = (\sigma_U, \tau_U)$. U computes $Y_U = X_{RA} \oplus H(Pwd_U || \sigma_U|| AID_U)$ and generates a random $\alpha_U \in Z_{q\ \text{to derive}}^*$ $Rpwd_U = H(\alpha_U || Pwd_U || AID_U)$ and $Rbio_U = H(\sigma_U || AID_U || \alpha_U)$. Next, the following parameters are generated: $\alpha_U' = \alpha_U \oplus H(Pwd_U || \sigma_U)$, $Rpwd_U' = Rpwd_U \oplus H(PRID_U || \sigma_U ||Pwd_U)$ and $Rbio_U' = Rbio_U \oplus H(\sigma_U || PRID_U || AID_U ||Pwd_U)$. The local authentication parameter is computed as $Z_U = H(\alpha_U || Rpwd_U || Rbio_U || AID_U)$. The user U's private key is generated as $pr_U \in Z_q^*$ and the corresponding public parameter as $Pub_U = pr_U \bullet G$, where $x \bullet G = G + G + \ldots + G$ (x times) represents the elliptic curve point (scalar) multiplication, that is, the point G added to itself x times. The user U then stores the parameters $\{\alpha_U', Rpwd_U', Rbio_U', AID_U, Gen(.), Rep(.), H(.), E_q(a, b), G, , \tau_U, et\}$ into a smart card, where *et* is an "error tolerance threshold value used in the fuzzy extractor deterministic reproduction function *Rep(.)*".

2) Gateway registration phase: Like the user registration phase, the gateway registration phase occurs as follows.

- **Step 1.** The RA selects an actual identity (AID_{HGW}) of the home gateway, registration timestamp RTS_{HGW} and secret key $SK_{HGW} \in Z_q^*$, and calculates pseudo-random identity of gateway ($PRID_{HGW}$) as $PRID_{HGW} = H(AID_{HGW} || SK_{RA} || RTS_{HGW})$. It also computes a long term secret of the home gateway (LTS_{HGW}) as $LTS_{HGW} = H(AID_{HGW} || SK_{HGW} || SK_{RA} || RTS_{HGW})$. The RA sends the registration credentials $<PRID_{HGW}, LTS_{HGW}, (PRID_U, SID_U)>$ to the home gateway via secure channel.
- **Step 2.** After receiving the registration credentials from the RA, the gateway selects its private key as $pr_{HGW} \in Z_q^*$, generates the corresponding public parameter as $Pub_{HGW} = pr_{HGW} \bullet G$ and stores $\{PRID_{HGW}, LTS_{HGW}, (PRID_U, SID_U), (pr_{HGW}, Pub_{HGW})\}$in its secure database in order to avoid stolen verifier attack.

3.2.3 Login Phase

To login to the smart home network, a registered user U needs the following steps:

- **Step 1**. U inserts his smart card and initiates the process by providing his actual identity AID_U, secret password Pwd_U, his/her bio-metrics bio_U' at the sensor of a specific terminal, say mobile device.
- **Step 2.** The smart card regenerates the biometric secret key σ_U using "fuzzy extractor deterministic reproduction function *Rep(.)*" as $Rep(bio_U', \tau_U) = \sigma_U$ provided that the "Hamming distance between the original registered biometric and current entered biometric is less than or equal to the error tolerance threshold value *et*", and computes $\alpha_U = \alpha_U' \oplus H(Pwd_U \,||\, \sigma_U)$, $Rpwd_U = Rpwd_U' \oplus H(PRID_U \,||\, \sigma_U \,||Pwd_U)$, $Rbio_U = Rbio_U' \oplus H(\sigma_U \,||\, PRID_U \,||\, AID_U \,||Pwd_U)$ and $Z_U' = H(\alpha_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, AID_U)$. The smart then checks if $Z_U' = Z_U$. If it is so, the user login is successful. Otherwise, this phase is terminated.
- **Step 3.** U generates a random secret $r_U \in Z_q^*$, a timestamp T_U, and computes $R_U = H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet G$ and $S_U = H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \oplus H(PRID_U \,||\, R_U \,||\, Pub_U \,||\, H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet Pub_{HGW})$ along with a signature $Sign_{rU} = H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) + H(S_U \,||\, PRID_U || R_U) * pr_U \pmod q$ and sends the login request message $Msg_1 :< SID_U, S_U, R_U, Sign_{rU}, T_U >$ to the gateway HGW via public channel.

This phase is summarized in Fig. 2.

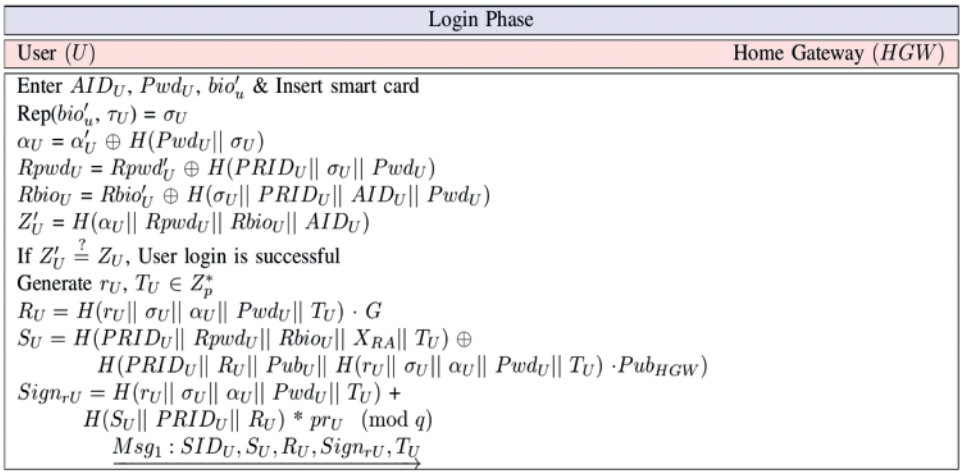

Figure 2: Login phase of a user U

3.2.4 Authentication and Key Establishment Phase

In this phase, both the home gateway, HGW and the user, U mutually authenticate each other and set up a common session key between them for secure data exchange. This is achieved by executing the following steps:

- **Step 1.** The gateway (HGW) receives Msg_1 at T_U' and verifies the timestamp $|T_U' - T_U| \leq \Delta T$. If it is valid, it extracts $PRID_U$ using the received SID_U from its memory and verifies the signature as $Sign_{rU} \bullet G = R_U + H(S_U \,||\, PRID_U \,||\, R_U) \bullet Pub_U$. If it is so, the HGW extracts $H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) = S_U \oplus H(PRID_U \,||\, R_U \,||\, Pub_U \,||\, pr_{HGW} \bullet R_U)$.
- **Step 2.** The HGW then generates a random secret $r_G \in Z_q^*$ and current timestamp T_G to compute $R_G = H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G) \bullet G$ and the session key $SK_{GU} = H(H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \,||\, H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G) \bullet R_U)$. A signature is also generated as $Sign_{rG} = H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G) + H(R_G \,||\, Pub_{HGW} \,||$

$PRID_{HGW} || SK_{GU} ||T_G) * pr_{HGW}$ (mod q). A new temporal session identity SID_U^{new} is generated for the user U and encrypted as $SID_U^* = SID_U^{new} \oplus H(R_G || R_U || SK_{GU} || Pub_{HGW} || Pub_U)$. The gateway HGW finally sends the message $Msg_2 :< R_G, SID_U^*, Sign_{rG}, T_G >$ to the user U via public channel.

- **Step 3.** The user U receives Msg_2 at time T_G' and verifies the timestamp by $|T_G' - T_G| \leq \Delta T$. If it is satisfied, U computes the session key as $SK_{UG} = H(H(PRID_U || Rpwd_U || Rbio_U || X_{RA} || T_U) || (H(r_U || \sigma_U || \alpha_U || Pwd_U || T_U) \bullet R_G))$. The signature is then verified as $Sign_{rG} \bullet G = R_G + H(R_G || Pub_{HGW} || PRID_{HGW} || SK_{UG} ||T_G) \bullet Pub_{HGW}$. If the signature is valid, the new session identity of the user is extracted as $SID_U^{new} = SID_U^* \oplus H(R_G || R_U || SK_{UG} || Pub_{HGW} || Pub_U)$ and it is then updated in the user U's smart card corresponding to SID_U.
- **Step 4.** The user U generates current timestamp T_V and computes a session key verifier as $SKV_{UG} = H(SK_{UG}||T_V)$ and sends the acknowledgment message $Msg_3 :< SKV_{UG}, T_V >$ to the home gateway, HGW via open channel.
- **Step 5.** When HGW receives the Msg_3 at time T_V', it verifies the timestamp as $|T_V' - T_V| \leq \Delta T$. If this is verified as correct, the HGW computes its own session key verifier as $SKV_{GU} = H(SK_{GU}||T_V)$. If it is found that $SKV_{UG} = SKV_{GU}$, the session key SK_{GU} is stored in the home gateway HWG's secure memory. Simultaneously, the session key SK_{UG} is also stored by the user U's smart card memory.

This overall phase is finally briefed in Fig. 3.

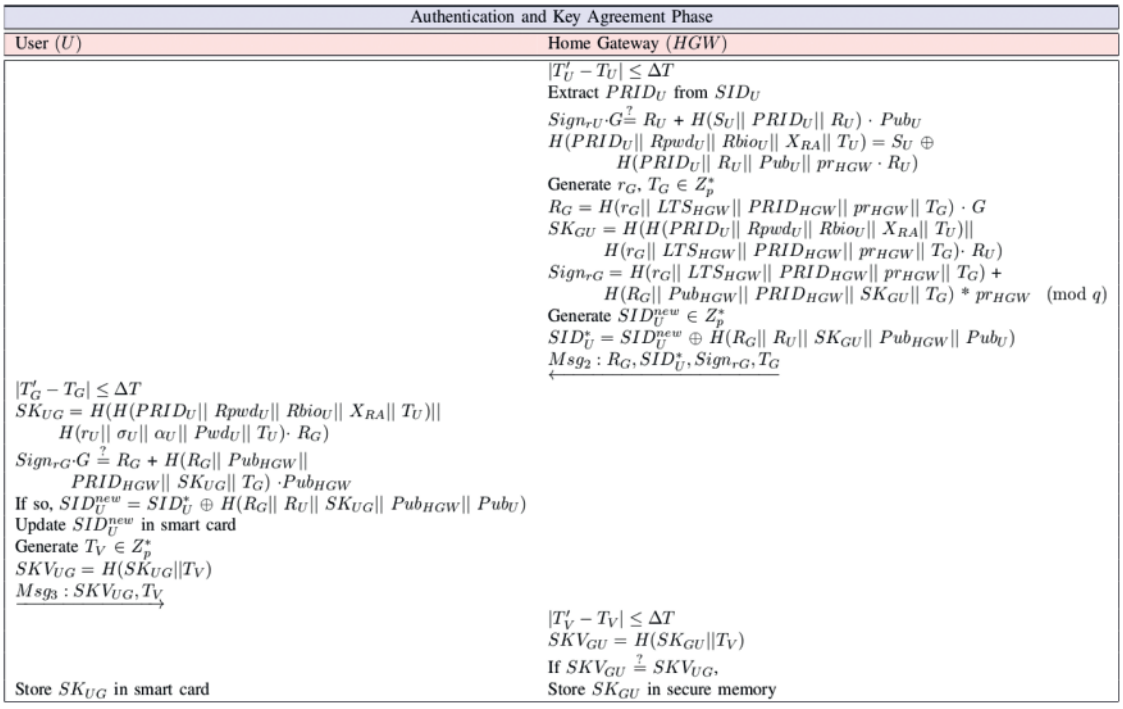

Figure 3: Authentication and key agreement among a user (U) and the home gateway node (HGW)

3.2.5 Secure Data Aggregation Phase

After the successful authentication and key establishment between a user and the gateway node, the transactions can be now sent in encrypted way using the established session key, along with the hash of transactions which will ensure privacy, integrity, and authentication between the communication parties. In this way, the in-charge gateway HGW will securely gather all the transactions, which will be used in forming the blocks for mining into the blockchain.

3.2.6 Block Addition Phase

Once the home gateway (HGW) receives the transactions from the authorized users, it will be routing then to one of the consensus nodes of private blockchain, and once the transactions threshold is reached, the consensus node, being the leader (miner) node in the Peer-to-Peer (P2P) blockchain network, creates a block with the useful parameters which make it immutable. The transactions (Tx_i) in the block are encrypted with the public key of HGW in order to ensure privacy. Then using the voting-based Practical Byzantine Fault Tolerance (PBFT) consensus algorithm [12], the leader determines whether to add or discard the block. The structure of a block is shown in Fig. 4. A constructed block has two parts: a) block header and 2) block payload. The block header contains the following fields:

- **Block version (BV):** It is a unique version number attached to each created block into the blockchain.
- **Previous block hash (PBH):** For creating the chain of blocks into the blockchain, the hash value (using Secure Hash Algorithm (SHA-256) [13] which produces 256-bit hash output) of the previous block of a current block is calculated.
- **Merkle Tree Root (MTR):** It is created as the hash value of all the transactions present in a block. Note that the Merkle tree is considered as a **hash tree** is a tree in which each "leaf node" is labelled with the cryptographic hash value of a data block containing the digital information. On the other side, each node that is not a leaf node is also labelled with the cryptographic hash value of the labels of its child nodes. Thus, a hash tree (Merkle tree) is treated as a generalization of a "hash list" and a "hash chain".
- **Timestamp (T):** The time when a block was created.
- **Owner of block (BO):** It represents an entity in the network who has created the block.
- **Public key of signer (Pub_{HGW}):** The public key of the home gateway being the signer is used to verify the signature on the current block hash (CBlock). CBlock is used to create the signature using the "Elliptic Curve Digital Signature Algorithm (ECDSA) signature verification function" with the help of the private key pr_{HGW} of HGW.
- **List of encrypted transactions:** A list of n_t encrypted transactions is inserted as block payload. Here, the encryption is done using the public key Pub_{HGW} to provide confidentiality or privacy of the transactions which can be viewed by the HGW only.

To verify a block, we need three level verification process: 1) Merkle tree root verification helps to checks whether any transactions in the block are modified or not; 2) current block hash verification helps to check whether any block header fields, and the transactions are proper or not; and 3) block signature verification using the ECDSA verification algorithm helps in verifying whether the overall block is authentic or not.

When an authorized user wants to remotely operate the smart devices at his home, he must provide his secret credentials and biometric details, through which the smart card authenticates him as a genuine user and will allow him in exchanging a secret session key. The user then passes all the essential parameters needed to the home gateway for constructing a session key. Through the authentication and

key establishment phase of the scheme, the home gateway and the user will share a secret session key. Using the shared session key, the user can encrypt the transactions (with the necessary details of IoT device and commands to it) and can send the transactions to the home gateway securely through the secure data aggregation phase. The home gateway decrypting the transactions will send the commands to the appropriate IoT device(s) and through the block addition phase of the scheme, the transactions will be added into the blockchain. The results from IoT devices are also encrypted with the same key and will be passed to the user.

Block Header	
Block Version (BV)	Unique block version number
Previous Block Hash (PBH)	Hash value of previous block
Merkle Tree Root (MTR)	Merkle tree root on transactions
Timestamp (T)	Time of block creation
Owner of Block (BO)	Home Gateway Node (HGW)
Public key of Signer	Pub_{HGW}
Block Payload (Encrypted Transactions)	
List of Transactions	$(E_{Pub_{HGW}}(Tx_l)\mid l= 1,2,3,\ldots,n_t$ $\vdots$
Current Block Hash ($CBlock$)	Hash of current block
Signature on Transactions	$ECDSA_{pr_{HGW}}$

Figure 4: Structure of a block in blockchain

The overall working flow diagram of the proposed smart home system including the block addition phase is summarized in Fig. 5. In the process flow diagram of the proposed scheme, two types of communication are performed through a) private/secure channel and b) public (open) channel. Here, the private channel is used for registration of the entities in a smart home environment. For instance, a user in smart home can register with the trusted RA in person by submitting the registration credentials to the RA and then receive a smart card from the RA. On the other side, the public channel is used for all other communications excluding the registration phase.

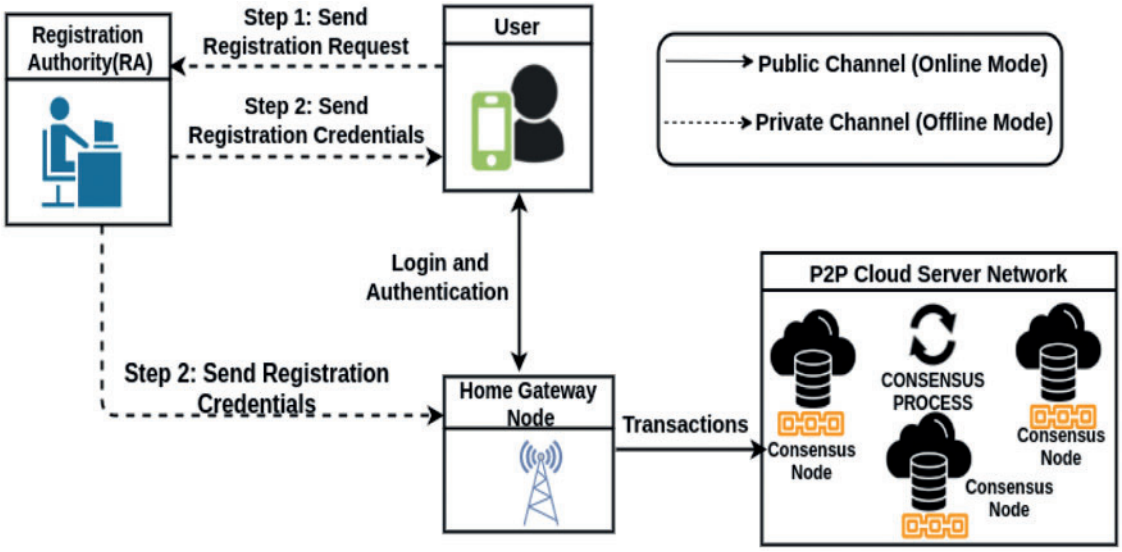

Figure 5: Process flow diagram of the proposed smart home system

4 Security Analysis

In this section, through both the formal and informal security analysis, we show the robustness of the proposed scheme against various potential attacks that are possible in case of a smart home environment.

4.1 Formal Security Analysis

This section analyses the proposed scheme using the widely adopted Real-Or-Random (ROR) oracle model [14] as follows.

The ROR model models a one-way cryptographic hash function H(•) as a random oracle *Hash*. Let $\Lambda_U{}^{w1}$ and $\Lambda_{HGW}{}^{w2}$ denote the instances of the participants user and the home gateway in the proposed scheme, respectively. The ROR analysis uses the queries defined in Tab. 2.

Table 2: Queries and their significance

Query	Purpose
Send(Λ^w, msg)	The participant Λ^w receives msg from an adversary A and responds to it
Execute($\Lambda_U{}^{w1}$, $_{HGW}{}^{w2}$)	A listens to the messages exchanged between $\Lambda_U{}^{w1}$ and $\Lambda_{HGW}{}^{w2}$
CorruptUD($\Lambda_U{}^{w1}$)	A extracts secrets from the lost or stolen user device UD
Reveal(Λ^w)	A session key between Λ^w and its partner is revealed to the adversary A
Test(Λ^w)	A checks if the derived session key is valid with a probability of a coin flip

Definition 1 (Semantic security): A polynomial time adversary A may have an advantage $Adv_A{}^{BlockUbiHome}(poly_t)$ to derive the session key $SK_{UG}(=SK_{GU})$ for the proposed scheme BlockUbiHome given by $Adv_A{}^{BlockUbiHome}(poly_t) = |2Pr[bit_g - bit_c] - 1|$ where Pr[E] represents the probability of event E and bit_g, bit_c are the guessed and correct bits.

Theorem 1. Let $Adv_A{}^{BlockUbiHome}(poly_t)$ be the advantage of a polynomial time adversary A in deriving the session key SK_{UG} ($= SK_{GU}$) for the proposed scheme BlockUbiHome such that $bits_{count}$ be the number of bits in the biometric secret key σ_U under the Zipf's law, then

$$Adv_A^{BlockUbiHome}(poly_t) \leq \frac{q_{hash}^2}{|Hash|} + 2[max\{C'.q_{send}^{s'}, \frac{q_{send}}{2^{bits_{count}}}\} + Adv_A^{ECDDHP}(poly_t)],$$

where $q_{hash,}$ q_{send} and $|Hash|$ represent "the number of hash queries, send queries and the range space of H(•)", $Adv_A^{ECDDHP}(poly_t)$ is the advantage of an adversary A in breaking the ECDDHP problem and C', s' are Zipf's parameters [15].

Proof. Four games $Game_i^A$, $i = 0, 1, 2, 3$ that involves the adversary A trying to win the game by correctly guessing random bit in game $Game_i^A$ denoted by the event $Success_{Game_i^A}^A$ and the corresponding advantage denoted by $Adv_{Game_i^A} = \Pr[Success_{Game_i^A}^A]$. The games and the corresponding probability of A in winning each game are defined as follows:

- ***Game*$_0^A$:** In the first game, A tries to pick a bit randomly bit_c randomly that gives the adversary advantage against the proposed scheme as

$$Adv_A^{BlockUbiHome}(poly_t) = |2\ Adv_{Game_0^A} - 1| \quad (1)$$

- ***Game***$_1^A$: In the second game, the adversary A attempts to derive the session key by running the *Execute* query to eavesdrop on the messages $Msg_1 = <SID_U, S_U, R_U, Sign_{rU}, T_U>$ and $Msg_2 = <R_G, SID_U^*, Sign_{rG}, T_G>$ during the authentication key agreement phase. The session key is constructed as $SK_{GU} = H(H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \,||\, H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G) \bullet R_U) = H(H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \,||\, H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet R_G) = SK_{UG}$. The session key is constructed using the temporal secrets r_U, r_G and the long-term secrets σ_U, α_U, Pwd_U, LTS_{HGW}, $X_{RA,}$ $Rpwd_U$, $Rbio_U$ and $pr_{HGW.}$ These credentials cannot be obtained from the eavesdropped messages. A uses the Reveal and Test queries to check if the derived session key is the original session key used by the entities or is a random key. Since the intercepted messages cannot help in deriving the session key $SK_{GU} = (SK_{UG})$, $Game_1^A$ and $Game_0^A$ are indistinguishable under the eavesdropping attack leading to the conclusion:

$$Adv_{Game_1^A} = Adv_{Game_0^A} \tag{2}$$

- ***Game***$_2^A$**:** In the third game, adversary A launches an active attack using the CorruptUD query, wherein the user device is stolen or lost and A has obtained all the secret user credentials α_U', $Rpwd_U$', $Rbio_U$', AID_U. However, these credentials alone cannot help in deriving the secret key as A would require the parameters r_U, $r_{G,}$ SK_U, Pwd_U, σ_U , α_U, $Rpwd_U$, $Rbio_{U.}$ At most $bits_{count}$ nearly random bits can be extracted using the fuzzy extractor method, which gives the approximate probability of guessing σ_U to be $\frac{1}{2^{bits_{count}}}$. Thus, in the absence of "password/biometrics guessing attacks", the games $Game_1^A$ and $Game_2^A$ becomes identical. Based on Zipf's law for user-selected passwords, the following relation may be derived:

$$|Adv_{Game_1^A} - Adv_{Game_2^A}| \leq \max\left\{C', q_{send}^{s'}, \frac{q_{send}}{2^{bits_{count}}}\right\} \tag{3}$$

• ***Game***$_3^A$**:** In this fourth game, the adversary A tries to emulate the hash operation and the elliptic curve multiplication operation so as to obtain the corresponding results during the authentication scheme. For this, A simulates the *Hash* random oracle and simulates hash queries. In addition, it needs to solve the elliptic curve decisional Diffie-Hellman problem (ECDDHP) to derive the session key. In the proposed scheme, $R_U = H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet G$ and $S_U = H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \oplus H(PRID_U \,||\, R_U \,||\, Pub_U \,||\, H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet Pub_{HGW})$ and $R_G = H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, prHGW \,||\, T_G) \bullet G$ are available in public channel. To derive the session key $SK_{GU} = H(H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \,||\, R_U \bullet H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G)) = H(H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U) \,||\, H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U) \bullet R_G) = SK_{UG}$, adversary A needs to solve ECDDHP to extract $H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U)$ and $H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G)$ from R_U, R_G and their product $R_U \bullet R_G$. Let the adversary A's advantage in solving ECDDHP be $Adv_A^{ECDDHP}(poly_t)$. Also, A needs to query the random oracle to obtain the collision of hash results $H(r_U \,||\, \sigma_U \,||\, \alpha_U \,||\, Pwd_U \,||\, T_U)$, $H(r_G \,||\, LTS_{HGW} \,||\, PRID_{HGW} \,||\, pr_{HGW} \,||\, T_G)$ and $H(PRID_U \,||\, Rpwd_U \,||\, Rbio_U \,||\, X_{RA} \,||\, T_U)$ using q_{hash} number of queries. Applying the birthday paradox, for the hash collision over the ECDDHP advantage of the adversary we obtain:

$$|Adv_{Game_2^A} - Adv_{Game_3^A}| \leq \frac{q_{hash}^2}{|Hash|} + Adv_A^{ECDDHP}(poly_t) \tag{4}$$

After the games are played, the adversary now guesses the correct bit bit_c , which has an advantage given by

$$Adv_{Game_3^A} = \frac{1}{2} \tag{5}$$

Considering Eqs. (1)–(5), we get

$$\frac{1}{2}Adv_A^{BlockUbiHome}(poly_t) = \left|Adv_{Game_0^A} - \frac{1}{2}\right|$$
$$= |Adv_{Game_1^A} - Adv_{Game_3^A}|$$

Now, we get

$$\frac{1}{2}Adv_A^{BlockUbiHome}(poly_t) = |Adv_{Game_1^A} - Adv_{Game_3^A}|$$
$$\leq |Adv_{Game_1^A} - Adv_{Game_2^A}| + |Adv_{Game_2^A} - Adv_{Game_3^A}| \quad (6)$$
$$\leq \max\left\{C' \cdot q_{send}^{s'}, \frac{q_{send}}{2^{bits_{count}}}\right\} + \frac{q_{hash}^2}{2|Hash|} + Adv_A^{ECDDHP}(poly_t)$$

Multiplying both sides by 2 in Eq. (6), we get

$$Adv_A^{BlockUbiHome}(poly_t) \leq \frac{q_{hash}^2}{|Hash|} + 2\left[max\left\{C' q_{send}^{s'}, \frac{q_{send}}{2^{bits_{count}}}\right\} + Adv_A^{ECDDHP}(poly_t)\right]$$

4.2 Informal Security Analysis

The informal security analysis proves that the proposed scheme is resistant to various known attacks.

Proposition 1. The proposed scheme *BlockUbiHome* is secure against ESL Attack.

Proof. In the proposed scheme, the computed session key is constructed as $SK_{GU} = H(H(PRID_U || Rpwd_U || Rbio_U || X_{RA} || T_U) || R_U \bullet H(r_G || LTS_{HGW} || PRID_{HGW} || pr_{HGW} || T_G)) = H(H(PRID_U || Rpwd_U || Rbio_U || X_{RA} || T_U) || H(r_U || \sigma_U || \alpha_U || Pwd_U || T_U) \bullet R_G) = SK_{UG}$. The session key is constructed using the temporal secrets r_U, r_G and the long term secrets σ_U, α_U, Pwd_U, LTS_{HGW}, X_{RA}, $Rpwd_U$, $Rbio_U$ and pr_{HGW}. If only the long-term secrets are revealed, then the short-term secrets protect the session key from compromise. Similarly, if the short-term secrets are compromised, then the long-term secrets ensure that the session key is protected.

Proposition 2. The proposed scheme *BlockUbiHome* is secure against privileged insider attack.

Proof. The home gateway node registration phase does not require the gateway to share any of its credentials. All the required parameters are pre-loaded by the registration authority into the gateway node's secure memory. The registration of the user ensures that AID_U, SK_U, and RTS_U are not revealed to the registration authority. The user credentials created do not use any parameters generated by the RA. The only parameter X_{RA} is used in the session key construction but not in any user credentials. In addition, usage of random secret α_U ensures that the identity and password of the user cannot be guessed by the attacker due to the one-way property of the collision-resistant hash function. Thus, privileged insider attack cannot be successful against the proposed scheme.

Proposition 3. The proposed scheme *BlockUbiHome* is resilient against replay attack.

Proof. Consider that the messages $Msg_1 :< SID_U, S_U, R_U, Sign_{rU}, T_U >$, $Msg_2 : < R_G, SID_U^*, Sign_{rG}, T_G >$ and $Msg_3 :< SKV_{UG}, T_V >$ are intercepted by the adversary A during the authentication phase of the scheme *BlockUbiHome*. A may try to replay the same messages by capturing these messages and resending them to the destination entity. In such as case, as per the designed scheme, the destination

entity will verify the freshness of the timestamp and find that these are replayed messages and discard them. Thus, the proposed scheme resists replay attack.

Proposition 4. The proposed scheme *BlockUbiHome* is resilient against man-in-the-middle attack.

Proof. Consider that the messages $Msg_1 :< SID_U, S_U, R_U, Sign_{rU}, T_U >$, $Msg_2 : < R_G, SID_U^*, Sign_{rG}, T_G >$ and $Msg_3 :< SKV_{UG}, T_V >$ are captured and tampered by the adversary A during transit in the public channel. In Msg_1 and Msg_2, the parameters are all verified by the signature $Sign_{rU}$ and $Sign_{rG}$, respectively. The message Msg_3 is intended to verify if the session keys computed at both sides are equal. Hence, the computation of the verifier itself ensures that MiTM fails as the session keys themselves are never exchanged directly.

Proposition 5. The proposed scheme *BlockUbiHome* is resilient against impersonation attacks.

Proof. Consider that the messages $Msg_1 :< SID_U, S_U, R_U, Sign_{rU}, T_U >$, $Msg_2 : < R_G, SID_U^*, Sign_{rG}, T_G >$ and $Msg_3 :< SKV_{UG}, T_V >$ are intercepted by the adversary A during the authentication phase of the scheme *BlockUbiHome*. The following cases are studied:

- ***User impersonation attack****:* To launch this attack, the adversary A tries to impersonate the user U. For this, A generates r_U^A, T_U^A and computes $R_U^A = H(r_U^A||\sigma_U||\alpha_U||Pwd_U||T_U^A) \cdot G$, $S_U^A = H(PRID_U||Rpwd_U||Rbio_U||X_{RA}||T_U^A) \oplus H(PRID_U||Pub_U||H(r_U^A||\sigma_U||\alpha_U||Pwd_U||T_U^A) \cdot Pub_{HGW})$ and $Sign_{rU}^A = H(r_U^A||\sigma_U||\alpha_U||Pwd_U||T_U^A) \oplus H(S_U||PRID_U||R_U) * pr_U (mod\ q)$. However, it can be observed that the long term secrets $Rpwd_U$, $Rbio_U$, X_{RA}, σ_U, α_U, Pwd_U should be known to A in order to order to produce a valid message $Msg_1^A :< PRID_U, R_U^A, S_U^A, Sign_{rU}, T_U^A$. Hence, the proposed scheme *BlockUbiHome* is resistant to user impersonation attack.
- ***Home gateway node impersonation attack****:* For the adversary A to impersonate the home gateway, it needs to first generate a random secret as $r_G^A \in Z_p^*$, and also to create a fresh timestamp T_G^A, and then to compute the parameters like $R_G^A = H(r_G^A ||LTS_{HGW}|| PRID_{HGW} ||pr_{HGW}||T_G^A) \cdot G$, $SK_{GU}^A = H(H(PRID_U ||Rpwd_U||Rbio_U||X_{RA}|| T_U^A) ||R_U||H(r_G||LTS_{HGW}|| PRID_{HGW} ||pr_{HGW}||T_G^A))$ and the signature to be calculated using the private key pr_{HGW} of the HGW as follows: $Sign_{rG}^A = H(r_G^A ||LTS_{HGW}|| PRID_{HGW} ||pr_{HGW}||T_G^A) + H\left(R_U^A ||Rub_{HGW}||RRID_{HGW}||SK_{GU}^A|| T_G^A\right) * pr_{HGW}\ (mod\ q)$ To fabricate the message Msg_3: $< R_G^A, Sign_{rG}^A, SID_U^{*A}, T_G^A >$, A requires the long-term secrets $PRID_{HGW}$, pr_{HGW} and LTS_{HGW}. Hence, the scheme *BlockUbiHome* is secure against this impersonation attack.

Proposition 6. The proposed scheme *BlockUbiHome* is resilient against DoS attack.

Proof. The fuzzy extractor method used in the scheme employs the Hamming distance concept that helps avoid false acceptance and false rejection rates. This method is better in comparison to the one-way hash functions, perceptual hashing and biohashing, which cannot generate unique results from user biometric data even though they have reduced output error. In addition, the usage of timestamps in ensures message freshness and does not allow the same message from the same sender to be repeatedly received. This ensures that safety against DoS attack.

Proposition 7. The proposed scheme *BlockUbuiHome* achieves anonymity and untraceability.

Proof. The messages $Msg_1 :< SID_U, S_U, R_U, Sign_{rU}, T_U >$, $Msg_2 : < R_G, SID_U^*, Sign_{rG}, T_G >$ and $Msg_3 :< SKV_{UG}, T_V >$ in the authentication phase use only pseudorandom and temporal identities $PRID_U$ and SID_U^* of the user and hence no message can be traced to the original sender. Hence, the proposed scheme achieves anonymity and untraceability.

Proposition 8. The proposed scheme is resilient against stolen smart card attack.

Proof. The user stores the credentials α'_U, $Rpwd'_U$, $Rbio'_U$, AID_U on the smart card during the registration phase. However, none of these credentials reveal the secret parameters $\sigma_U, \alpha_U, Pwd_U$, and SK_U directly to the adversary A. In addition, offline biometric guessing attacks are infeasible on the 160-bit long identities and secret credentials. Hence, A cannot obtain any information from the stolen smart card, making the scheme secure against such attacks.

5 Comparative Study

This section performs thorough analysis of the proposed scheme and compares it with other schemes, such as the schemes of Fakroon et al. [5], Shuai et al. [6] and Naoui et al. [11] for costs undergone like the analysis done in the existing schemes [16–20].

5.1 Testbed Experimentation with MIRACL

The proposed scheme is examined through its execution time with the help of the open source SDK "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [21] based on C/C++ that gives the execution times of all required cryptographic operations. The notations for representing the cryptographic operations as denoted as T_h for SHA-256 hashing, T_{ecm} for elliptic curve multiplication, T_{eca} for elliptic curve addition, T_{enc} for symmetric key encryption using AES-128, T_{decr} for symmetric key decryption using AES-128, T_{exp} for exponentiation, and T_{bp} for bilinear pairing operation.

Two scenarios have been considered as follows.

Scenario 1: The platform for a server is taken as "Ubuntu 18.04.4 LTS, with memory: 7.7 GiB, processor: Intel Core i7-8565U CPU @ 1.80 GHz X 8, OS Type: 64-bit and disk: 966.1 GiB". 100 runs have been executed for each cryptographic operation to record the "maximum, minimum and average run time in milliseconds" for each operation. The results are tabulated in Tab. 3.

Table 3: Execution costs using MIRACL library for cryptographic primitives

Primitive	Average time on Raspberry PI 3 (in milliseconds)	Average time on server (in milliseconds)
T_h	0.309	0.055
T_{exp}	0.228	0.072
T_{ecm}	2.288	0.674
T_{eca}	0.016	0.002
T_{enc}	0.018	0.001
T_{dec}	0.014	0.001
T_{bp}	32.084	4.603

Scenario 2: The platform for a user device/smart device is considered as follows: "Raspberry Pi 3 B + Rev 1.3, with CPU: 64-bit, Processor: 1.4 GHz Quad Core, 4-cores, Memory (RAM): 1GiB, and OS: Ubuntu 20.04 LTS, 64-bit" [22] Once again, 100 runs are executed for each operation to record the maximum, minimum and average run-time for each operation. The results are also tabulated in Tab. 3.

5.2 Computational Costs Comparison

The proposed scheme is evaluated to obtain a computation cost of $11T_h + 5T_{ecm} + 2T_{eca}$ at the user and $7T_h + 5T_{ecm} + 2T_{eca}$ at the home gateway node. The comparison of computation costs is shown in Tab. 4.

Table 4: Communicational costs comparison

Scheme	User	Gateway	Smart device
Fakroon et al.	$4T_h \approx 1.236$ ms	$5T_h \approx 0.275$ ms	$3T_h \approx 0.927$ ms
Shuai et al.	$6T_h + 2T_{ecm} \approx 6.43$ ms	$7T_h + T_{ecm} \approx 1.059$ ms	$3T_h \approx 0.927$ ms
Naoui et al.	$5T_h + 2T_{ecm} + 2T_{enc} \approx$ 6.157 ms	$8T_h + T_{ecm} + 3T_{enc} \approx$ 1.117 ms	$T_h + 2T_{dec}$ ≈ 0.323 ms
Proposed (BlockUbiHome)	$11T_h + 5T_{ecm} + 2T_{eca} \approx$ 14.871 ms	$7T_h + 5T_{ecm} + 2T_{eca} \approx$ 3.757 ms	_

5.3 Communication Costs Comparison

The result of hash function (using SHA-256) takes 256 bits, and the result of "symmetric key encryption/decryption function using AES-128" [23] takes 128 bits. Choice of ECC is taken such that 160-bit ECC provides the same security as 1024-bit RSA algorithm. A given point on the elliptic curve takes 320 bits with each coordinate taking 160 bits. The proposed scheme is evaluated to obtain a communication cost of 1984 bits in 3 messages. The comparison of communication costs is shown in Tab. 5. The proposed scheme *BlockUbiHome* is observed to take the least communication cost among all compared schemes.

Table 5: Communication costs comparison

Scheme	No. of required messages	Total bits needed
Fakroon et al.	4	3072
Shuai et al.	4	3193
Naoui et al.	3	6304
Proposed (BlockUbiHome)	3	1984

5.4 Security and Functionality Features Comparison

The proposed scheme is compared with the relevant schemes to understand how many of the required security features and functionality features are supported by each scheme. It can be concluded from Tab. 6 that even though the scheme has slightly higher computational cost, its communication cost is very low and it achieves more features compared to the other schemes.

Table 6: Security and functionality features comparison

Feature	Fakroon et al.	Shuai et al.	Naoui et al.	Proposed (BlockUbiHome)
Anonymity	√	√	√	√
Untraceability	√	√	√	√
User impersonation attack	×	×	√	√
Stolen smart card/mobile device attack	×	×	×	√
ESL attack under CK-adversary model	×	×	×	√
Privileged-insider attack	√	×	×	√
Replay attack	√	×	√	√
Man-in-the-middle attack	√	√	√	√
Mutual authentication	√	√	√	√
Unauthorized login detection	√	√	√	√
DoS attack	×	×	×	√
Offline guessing attacks	×	×	×	√
Blockchain support	×	×	×	√

6 Implementation of Blockchain: Simulation Study

The blockchain simulations were performed on a platform "Ubuntu 20.04.3 LTS (Focal Fossa), 64-bit OS with Intel® Core™ i7-6820HQ CPU @ 2.70 GHz, 32 GiB RAM" using the "Practical Byzantine Fault Tolerance (PBFT)" consensus algorithm [12]. The main advantages of using the blockchain simulations for the proposed scheme are to show the effects of computational time needed for three cases: a) a varied number of transactions per each block, b) a varied number of P2P nodes in the blockchain network, and c) a varied number of blocks to be mined in P2P blockchain network. These will measure the effectiveness of the proposed scheme with respect to utilization of the blockchain technology with the traditional user authentication and key agreement procedures in a smart home environment.

The home gateway after receiving a transaction from an authorized user will pass the transaction to a consensus Node for storing it into the blockchain. We can make the consensus node build a block,

only after receiving some threshold number of transactions, so that storage overhead will be decreased. It creates a block which has the parameters essential to make the system achieve immutable and non-tampering. The transactions in the block are encrypted with its public key to ensure privacy [24]. It now requests leader for adding the block to the Network. Then using the PBFT algorithm for consensus, the leader determines whether to add or discard the block.

The size of the block <BV, PBH, MTR, T, BO, Signer Public Key, List of Transactions, Block Sign, CBH> can be computed as <32, 256, 256, 32, 160, 160, n_t* 160, 160, 256> totaling to about 1472 + n_t* 160 bits. The simulation is conducted on VS CODE 2019 programming platform with Nodejs language. The following scenarios are taken into account:

Case 1: The number of peer nodes in the network is taken to be 10 and the no of transactions is taken to be 15 in each block. The time for computation as the number of blocks mined is increased from 5 to 25 in steps of 5 as shown in Fig. 6.

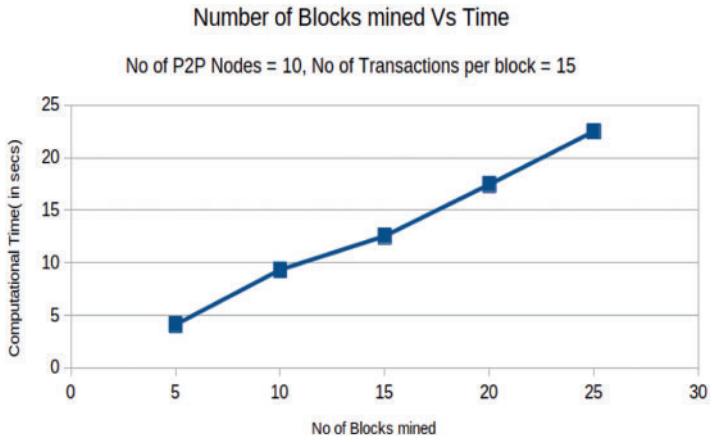

Figure 6: Simulation results: Number of blocks mined *vs.* computational time (in s)

Case 2: The number of peer nodes is taken to be 5 and the number of blocks mined is fixed at 10 while the number of transactions in each block is increased from 5 to 25 in steps of 5 to note the change in computation time as shown in Fig. 7.

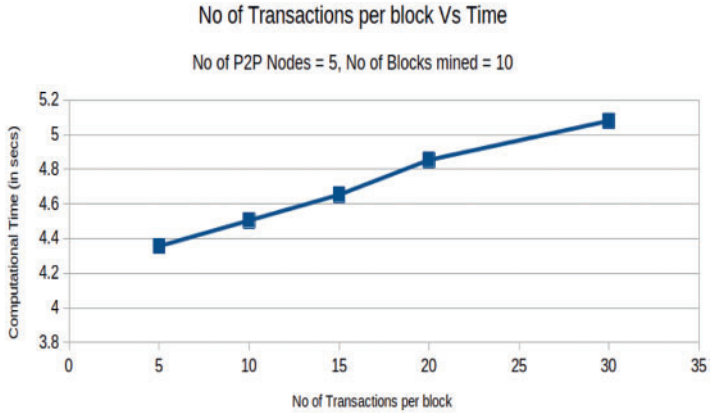

Figure 7: Simulation results: Number of transactions per block *vs.* computational time (in s)

Case 3: The number of peer nodes is varied from 10 to 30 in steps of 5 while the number of blocks mined is fixed at 10 and the number of transactions per block is fixed at 15. The respective computation time is noted during this variation as shown in Fig. 8.

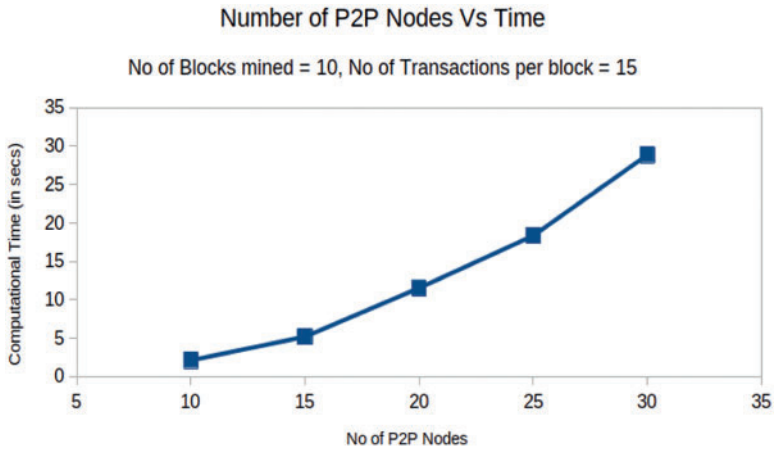

Figure 8: Simulation results: Number of P2P nodes *vs.* computational time (in s)

7 Conclusion and Future Work

The proposed research work presents a novel authentication protocol for smart home system using blockchain technology. A user can be remotely authenticated by the home gateway network using the proposed scheme. Once authenticated, the user may send any commands or instructions to be executed by a network of devices via the home gateway. Since the user is authenticated before sending commands, any malicious user is restricted from sending harmful commands to the home network. The proposed scheme is analyzed through a detailed analysis of the computation cost, communication cost and security features. In addition, an implementation of the blockchain shows only a linear variation in the increase of execution time with change in the peer nodes or number blocks or number transactions in each block. Future work includes lattice-based cryptographic techniques that can be embedded with the blockchain technology for smart home environment [25].

Acknowledgement: The authors would like to thank their universities for the support provided during this research. The authors would also like to thank the anonymous reviewers and the associate editor for their valuable feedback.

Funding Statement: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605. The authors also extend their gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under Grant Number R.G.P. 1/399/42.

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

References

[1] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[2] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, pp. 337–351, 2002, Springer, Berlin Heidelberg.

[3] M. Wazid, A. K. Das, V. Odelu, N. Kumar and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.

[4] T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[5] M. Fakroon, M. Alshahrani, F. Gebali and I. Traore, "Secure remote anonymous user authentication scheme for smart home environment," *Internet of Things*, vol. 9, pp. 100158, 2020.

[6] M. Shuai, N. Yu, H. Wan and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Computers & Security*, vol. 86, pp. 132–146, 2019.

[7] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay and Y. Park, "An efficient, anonymous and robust authentication scheme for smart home environments," *Sensors*, vol. 20, no. 4, pp. 1215, 2020.

[8] D. Kaur and D. Kumar, "Cryptanalysis and improvement of a two-factor user authentication scheme for smart home," *Journal of Information Security and Applications*, vol. 58, pp. 102787, 2021.

[9] M. Xu, Q. Dong, M. Zhou, C. Wang and Y. Liu, "Security analysis on anonymous authentication scheme for smart home environment with provable security," *Wireless Communications and Mobile Computing*, Vol. 2020, pp. 1–4. Article ID 8838363. 2020.

[10] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar *et al.,* "Homechain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2019.

[11] S. Naoui, M. E. Elhdhili and L. A. Saidane, "Lightweight and secure password based smart home authentication protocol: LSP-SHAP," *Journal of Network and Systems Management*, vol. 27, no. 4, pp. 1020–1042, 2019.

[12] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[13] W. E. May, "Secure hash standard," FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, accessed on January 2020.

[14] M. Abdalla, P. A. Fouque and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography (PKC 2005)*, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, vol. 3386, pp. 65–84, 2005.

[15] D. Wang, H. Cheng, P. Wang, X. Huang and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[16] A. Vangala, A. K. Sutrala, A. K. Das and M. Jo, "Smart contract-based blockchain-envisioned authentication scheme for smart farming," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10792–10806, 2021.

[17] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar *et al.,* "Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15824–15838, 2021.

[18] A. Vangala, A. K. Das and J. H. Lee, "Provably secure signature-based anonymous user authentication protocol in an internet of things enabled intelligent precision agricultural environment," *Concurrency and Computation: Practice and Experience*, pp. e6187, 2021, https://doi.org/10.1002/cpe.6187.

[19] B. Bera, A. Vangala, A. K. Das, P. Lorenz and M. K. Khan, "Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment," *Computer Standards & Interfaces*, vol. 80, pp. 103567, 2022.

[20] A. Vangala, A. K. Das, N. Kumar and M. Alazab, "Smart secure sensing for IoT-based agriculture: Blockchain perspective," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17591–17607, 2021.

[21] "MIRACL cryptograhic SDK: Multiprecision integer and rational arithmetic cryptograohic library," 2020, [Online]. Available: https://github.com/miracl/MIRACL, accessed on October 2021.
[22] "Raspberry Pi 3 model B+, " 2020, [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/, accessed on May 2021.
[23] "Advanced encryption standard," FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001, [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, accessed on June 2020.
[24] T. Li, H. Wang, D. He and J. Yu, "Permissioned blockchain-based anonymous and traceable aggregate signature scheme for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8387–8398, 2020.
[25] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das *et al.,* "LSCSH: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 24–32, 2018.