Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6458461, 13 pages
https://doi.org/10.1155/2022/6458461

Research Article

Certificateless Authentication Scheme Based on Blockchain in Smart Home Network

Xingang Zhang,[1] **He Li**,[1] **Xiao Tian**,[2,3] **Rui Zhang**,[4] **and Qinglei Qi**[1]

[1] *Henan Engineering Research Center of Intelligent Processing for Big Data of Digital Image, School of Computer Science and Technology, Nanyang Normal University, Nanyang 473061, China*
[2] *Department of Health Management, Nanyang Medical College, Henan, Nanyang 473061, China*
[3] *Xi'an Hengpin Electronic Technology Co., Ltd, Xi'an 710086, China*
[4] *Henan Costar Group Co., Ltd, Nanyang 473003, China*

Correspondence should be addressed to He Li; lihe@nynu.edu.cn

Received 15 July 2022; Accepted 8 August 2022; Published 26 August 2022

Academic Editor: Zaoli Yang

The traditional centralized intelligent home network security authentication schemes have security problems such as integrity and confidentiality, while the distributed schemes have the problem of delayed authentication. These schemes are not suitable for the intelligent home environment of edge computing. To solve these problems, a certificateless smart home network authentication scheme based on blockchain and certificateless cryptosystem is proposed to realize mutual authentication among users, intelligent terminals (ITs), and intelligent gateways (IGs). The aggregation signature scheme of certificateless identity is introduced into the authentication of intelligent terminals in smart home network. The IG only needs to generate a signature to complete the identity authentication of multiple ITs. Compared with other authentication schemes, the security verification and performance analysis show that the proposed scheme uses less computation overhead and achieves more security features.

1. Introduction

Smart home is an important application of the Internet of Things (IoT) under the development of mobile Internet [1]. Smart home network is a network that interconnects all types of intelligent terminals in the home through IoT technology. In recent years, with the development of artificial intelligence and big data technology, it has made home networks increasingly intelligent. However, due to the limitation of hardware resources of home smart terminal devices, cloud computing is usually required to meet personalized smart services in the home [2]. The characteristics of strong real-time interactivity and low latency of edge computing will become the standard for home network intelligent services [3]. However, the wireless local area network used in smart homes has serious security risks and is easily attacked by hackers. The interception of wireless information also makes the wireless communication of smart terminals useless [4], which brings many security threats to the access and transmission of home data. Therefore, new requirements and challenges are put forward for the identity authentication scheme of the smart home network. At present, IoT authentication technology is mainly based on cryptographic systems and is divided into two main methods: symmetric and public key. Symmetric key authentication is computationally small and fast, but there are security issues arising from key management and distribution [5]. Therefore, it is necessary to design an authentication scheme that not only meets the needs of the future smart home network architecture but also is safe and efficient.

In recent years, there has been a lot of research in the field of smart home network authentication. In order to deal with the malicious attacks of radio frequency identification (RFID) in radio communication, a new trust-based authentication scheme is proposed in the literature [6] for smart home systems. The evaluation results of the scheme show that interference attacks and cloning attacks can be effectively countered. Fog computing smart home system, the literature [7] proposes an authentication model in a fog environment (SecFHome) that includes an edge negotiation phase and an identity verification phase. SecFHome adds updated information to the authentication side and can verify the synchronization of messages while authenticating, improving the efficiency of authentication. To protect smart home data security, a new secure privacy protection scheme is proposed in the literature [8] to resist ephemeral secret leakage (ESL) simulation attacks. The security of the scheme against different known attacks is demonstrated using the stochastic model. To reduce smart home network security threats, literature [9] proposes a smart card-based secure addressing and authentication scheme by modifying the IPv6 protocol. This scheme uses a secret session key to prevent unauthorized access to the network. However, this solution is less versatile and more difficult to apply in practice. Considering the limited resources of smart home devices, literature [10] proposes a lightweight short token authentication scheme. However, the scheme has security loopholes, which cannot resist Dos attacks, communication between devices is limited, and the registration process is complicated. The literature [11] designs a permission access control method based on risk. It uses groups (grouped according to risk similarity) for setting permissions and authorizing access. The method is similar to the above methods, in that they all use a centralized architecture and suffer from a single point of failure.

To solve the problems in the traditional centralized authentication architecture, the literature [12, 13] introduces blockchain technology to smart home network authentication based on cloud computing to achieve decentralized network authentication. But this scheme uploads data to the cloud, which has the problems of data leakage, high computational effort, and communication overhead. In order to solve these problems, the literature [14] introduced edge computing to realize the authentication and control of the smart home network based on the blockchain technology. Therefore, to solve this problem, this paper combines a certificateless cryptosystem with blockchain and proposes a blockchain-based certificateless authentication scheme (BCAS), which realises mutual authentication of users, smart terminals, and smart gateways. At the same time, to cope with the growth in the number of smart terminals in a smart home network, one-to-one smart terminal authentication may lead to system resource consumption and signalling congestion problems [15]. This paper also introduces a secure and efficient aggregated signature scheme for certificate-free identities to the authentication of smart terminals in smart home networks. Realizing the authentication of multiple smart terminals can be completed with a single signature authentication. The security and performance analysis illustrates that the proposed solution uses less computational overhead while ensuring security. Compared with existing certificateless authentication Scheme of smart home network, our proposed algorithm contributes the following improvements:

The smart home authentication network is constructed based on blockchain that includes the IT, intelligent app, user, IG, and ES.

A user authentication scheme based on blockchain and tripartite key negotiation is designed to improve the authentication security among users, IG, and ITs.

A single authentication scheme based on blockchain and public key signature is proposed to achieve secure and efficient authentication among IT, IG, and ES.

To improve the authentication performance of multiple ITs, the aggregation signature scheme of certificateless identity is introduced into the authentication of intelligent terminals in smart home network.

This paper is organized as follows. Section 2 presents preliminary knowledge of smart home security authentication. Section 3 introduces blockchain-based authentication scheme. Section 4 describes the security analysis of blockchain-based authentication scheme. Simulation results and performance analyses of the scheme are presented in Section 5. Finally, the conclusions are drawn in Section 6.

2. Preliminary Knowledge

2.1. Blockchain. Blockchain is a distributed ledger technology, proposed by selfproclaimed Satoshi Nakamoto in 2008. It is the core technology of Bitcoin. Different from the traditional banking transaction system, the blockchain can realize the verification and storage of transactions through distributed nodes without any trusted central authority [16]. Each node on the blockchain has a high degree of autonomy, and transactions on the blockchain network require consensus among all participants. Therefore, the blockchain has the characteristics of decentralization, data are difficult to tamper with, and traceable. It has these properties because the blockchain stores data through a chain structure, specifically as shown in Figure 1.

Each block is divided into a block header and a block body. Where the block header includes information such as version number, parent block hash, timestamp, and Merkle tree root value. The block body is used to store all types of transaction information. Where each transaction is permanently stored in the block body and requires a digital signature from the parties to the transaction; thus, ensuring that the data cannot be forged. Therefore, the data information recorded in the blockchain is extremely reliable and can solve the problem of entities not trusting each other in information systems.

2.2. Certificateless Trilateral Key Negotiation. Diffie and Hellman proposed the first two-party single-round key agreement protocol in 1976, which allows participating users to calculate a shared session key through public information. The shared session key is calculated from the secret value chosen by each of the participants. The literature [17] proposes a first tripartite key negotiation protocol in communication systems to enable three entities to construct a shared session key. The efficient certificateless tripartite key negotiation protocol proposed above is used here, and the specific tripartite key negotiation process is shown in Figure 2.

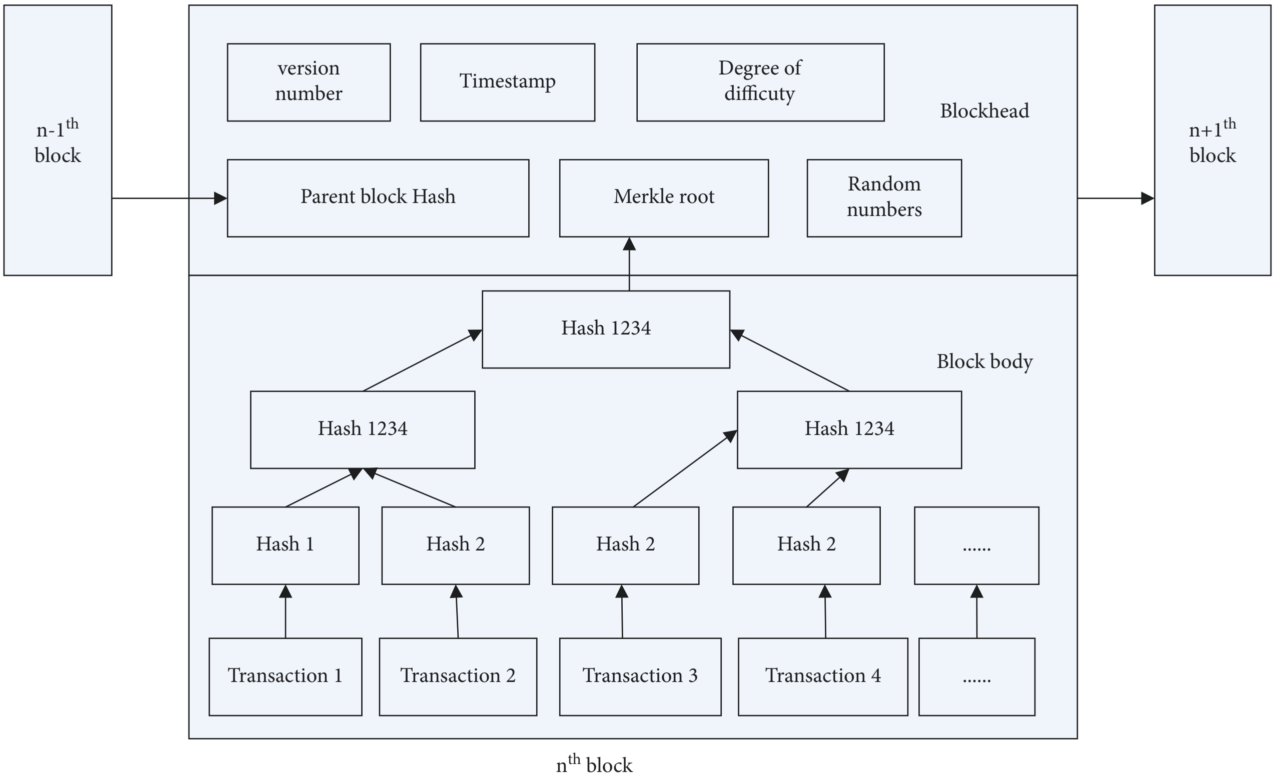

Figure 1: Blockchain structure.

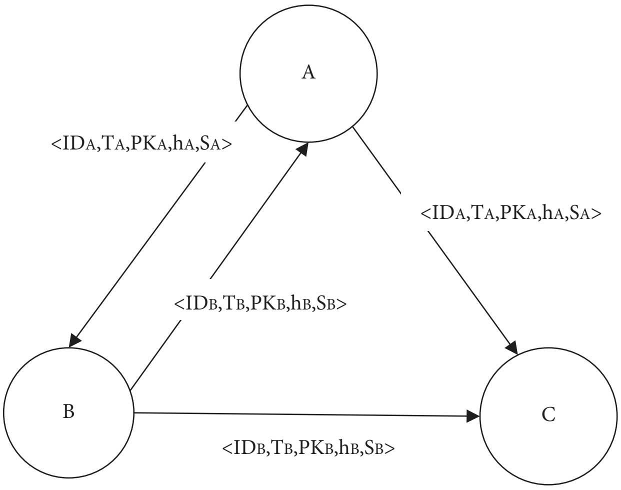

Figure 2: Tripartite key agreement negotiation process.

ID_A, ID_B, and ID_C are the respective identities of users A, B, and C. They submit their identities to the key generation center to generate their respective partial private keys, D_A, D_B, and D_C. PK_A, PK_B, and PK_C are the respective private keys of users A, B, and C. T_A, T_B, and T_C are the random numbers secretly chosen by each of users A, B, and C. $T_A = g^a$, $T_B = g^b$, and $T_C = g^c$. $<h_C,\ s_C>$ is the signature generated by user C, where h_C is the hash value of user identity information, random number, and information to be signed. The s_C is part of the private key output by user C. After the tripartite has completed the exchange of information and authentication in compliance with the rules of the protocol, they can each obtain a common key K by calculation and use this key K for secure communication in future communications.

2.3. *Certificateless Aggregate Signature.*

The first identity-based aggregated signature scheme using fork priming was proposed by Cheon et al. in 2004. It is proposed to facilitate the one-time verification of multiple signatures of multiple users. The signed message can only be known by the signer and the key generation center, and the signature private key is associated with the identity information. The so-called aggregated signature is to aggregate the signatures of n signers to n different messages into one signature, so that the verification equation of n signatures can be reduced to one verification equation. Therefore, the aggregated signature greatly improves the efficiency of signature verification and transmission.

2.4. *Difficult Questions and Hypothesis*

Definition 1. Diffie–Hellman (CDH) problem.

Let p and q be two prime numbers and satisfy $q|(p-1)$, let g be a generating element of order q in Z_p^*, $a, b\in Z_p^*$, given definite $g, g^a, g^b \in Z_p^*$, and calculate $g^{ab} \in Z_p^*$. The probability of the CDH problem being successfully calculated by algorithm A is

$$\mathrm{Succ}^{C\ DH}_{Z_p^*,A} = \Pr\left[g^{ab} \leftarrow A\left(g, g^a, g^b\right)\right], \qquad (1)$$

where the calculated probabilities are based on the random selection of algorithm A and the random selection of a and b on Z_p^*.

Hypothesis 1. CDH probability hypothesis: $\mathrm{Succ}^{C\,DH}_{Z^*_{p,A}}$ is negligible under any polynomial algorithm A.

Definition 2. Discrete logarithm (DL) problems: let p and q be two prime numbers and satisfy $q|(p-1)$, let g be a generating element of order q in Z_p^*, a, $b\in Z_p^*$, given definite $g, g^a \in Z_p^*$, and calculate $a\in Z_p^*$. The probability of the DL problem being successfully calculated by algorithm A is

$$\mathrm{Succ}^{DL}_{Z_p^*,A} = \Pr[a \leftarrow A(g, g^a)], \tag{2}$$

where the calculated probabilities are based on the random selection of algorithm A and the random selection of a on Z_p^*.

Hypothesis 2. DL probability hypothesis: $\mathrm{Succ}^{C\,DH}_{Z^*_{p,A}}$ is negligible under any polynomial algorithm A.

3. Blockchain-Based Authentication Scheme

3.1. Network Authentication Model. The smart home network built in this paper includes intelligent terminal (IT), intelligent application, user, intelligent gateway (IG), and edge server (ES), as shown in Figure 3. The IT mainly includes smart lights, smart audio, smart access control, smart air conditioning, home security systems, and various sensor devices. These terminal devices form a heterogeneous Internet of Things that collects and monitors various types of data in the home environment. The IG is an important communication device in the home network, with functions such as computing and storage, and acts as a full node of the blockchain, enabling all types of transaction operations (such as creation, validation, and query). The IG stores various types of data collected by IT, and the user (family members in the smart home) can query and manipulate these data as required. Edge computing refers to a new model of performing distributed computing at the edge of the network [18], with ES acting as an edge cloud responsible for smart home big data processing.

First, the network system initialization is performed by the ES, and the initialization generates the public parameters required for network authentication. Secondly, IT can collect environmental information in the family and use its own ID to complete registration and exchange information with IG. Among other things, IT should also create and store blockchain power tokens and smart contracts (smart contracts define permissions, policies, and constraints associated with the service) on the blockchain. Then, the smart application authenticates the user by verifying the information entered by the user. Finally, after passing the identity verification, users can request blockchain power tokens and smart contracts from IT and then request access to relevant services from IG, and only after the access service request is approved, they can operate on IT with corresponding rights.

The detailed description of the proposed blockchain-based authentication scheme is as follows. The scheme includes system initialization, tripartite key negotiation, IT identity binding, and user identity binding.

3.2. System Initialization. System initialization is performed by ES, and a public key cryptosystem is constructed using the Elliptic Curve Integrated Encryption Scheme (ECIES) [13] with the following key generation process:

3.2.1. System Creation. Enter the security parameter k, and ES produces two large prime numbers p and q and satisfies $q|(p-1)$. Randomly choose a generating element g of order q of Z_p^*, the subgroup generated by is G. ES arbitrarily choose $x\in Z_p^*$, and calculate $y = g^x$. Select hash function:

$$\begin{aligned} &H_1\colon \{0,1\}^* \times Z_p^* \longrightarrow Z_q^*,\\ &H_2\colon \{0,1\}^* \times Z_p^* \times Z_p^* \longrightarrow Z_q^*,\\ &H_3\colon \{0,1\}^* \times \left(Z_p^*\right)^4 \times Z_p^* \times Z_p^* \longrightarrow Z_q^*. \end{aligned} \tag{3}$$

System public parameters $= \{p,\ q,\ g,\ y,\ H_1,\ H_2,\ H_3\}$, Master Key (msk) $= x$, and $x \in Z_p^*$.

3.2.2. Partial Private Key Generation. After entering the user's identity IDi, the KGC first randomly selects $x_i \in Z_q^*$, then calculate $R_i = g^{r_i}$, $Q_1 = H_1(ID_i, R_i)$, $D_i = xH_1(ID_i, R_i) = xQ_i$, return D_i to user i through a secure channel, and use D_i as part of user i's private key.

3.2.3. Secret Value Setting. User ID_i randomly selects $x_i \in Z_q^*$ as the long-term private key.

3.2.4. Private Key Setting. User ID_i enters the parameters params, a partial private key and D_i, output the S_i of the private key at the client, where $S_i = x_i Q_i$, and generates the full private key $SK_i = \ <S_i, D_i>$.

3.2.5. Public Key Setting. The user selects a random secret value x_i, $X_i = g^{x_i}$, to generate the public key $PK_i = \ <X_i, R_i>$.

3.3. Tripartite Key Negotiation. In the first use of IT, the user's participation is required, and the key negotiation between the three is completed through the interaction of IG. Here, $\mathrm{ID_{IT}}$, $\mathrm{ID_{IG}}$, and ID_U are the respective identities of IT, IG, and user U. They submit their identities to ES, which completes the system initialization and generates their respective partial private keys, $\mathrm{SK_{IT}}$, $\mathrm{SK_{IG}}$, and SK_U.

$\mathrm{PK_{IT}}$, $\mathrm{PK_{IG}}$, and PK_U are the respective public keys of IT, IG, and user U. N_{IT}, N_{IG}, and N_U are random numbers a, b, c, chosen secretly by each of IT, IG, and user U, then calculate $N_{\mathrm{IT}} = g^a$, $N_{\mathrm{IG}} = g^b$, and $N_U = g^c$, and $<h_U,\ s_U>$ is the signature generated by user U:

(1) IG sends $<\mathrm{ID_{IG}}, \mathrm{N_{IG}}, \mathrm{PK_{IG}}>$ to IT and U.

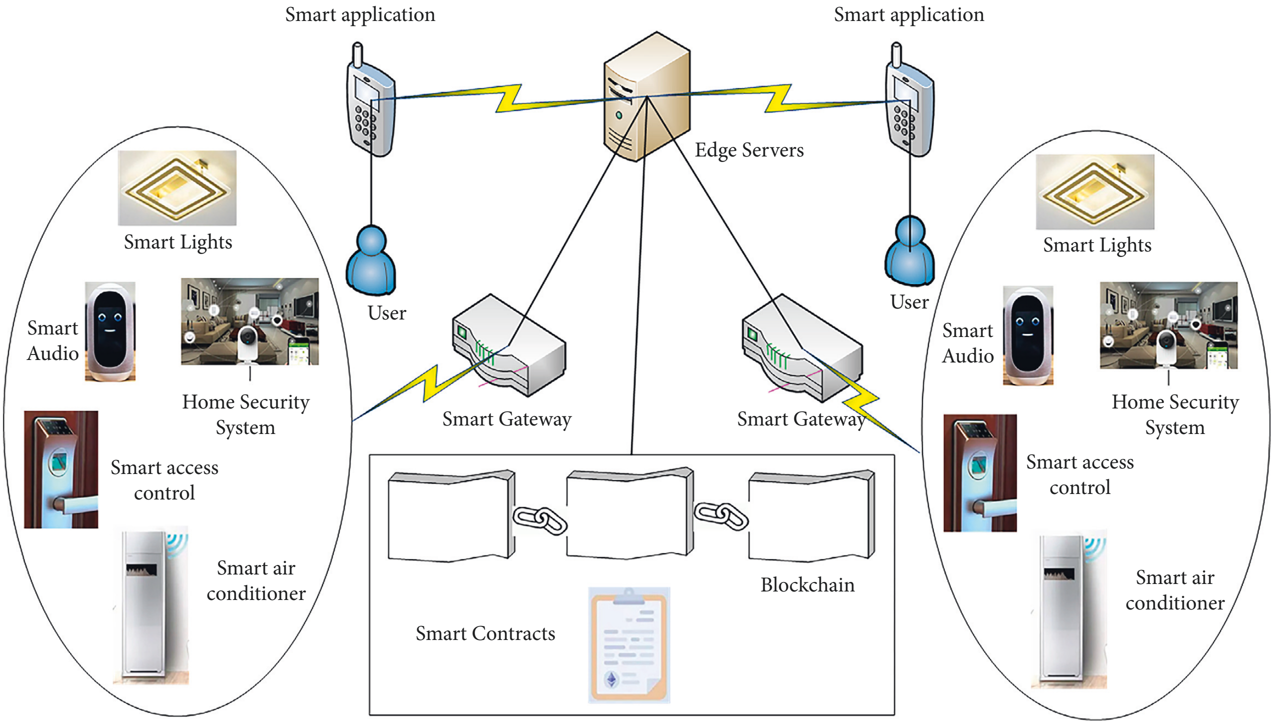

Figure 3: Smart home network structure.

(2) User U receives a message from the IG and generates a key negotiation response message < ID_U, N_U, PK_U>, which is also sent to the IG. IT receives a message from IG and generates a key negotiation response message < $\mathrm{ID}_{\mathrm{IT}}$, N_{IT}, $\mathrm{PK}_{\mathrm{IT}}$,>, at the same time sending the message to IG.

(3) After IG receives messages from user U and IT, calculate $K_{IG} = g^{cx+x_c}(N_U PK_U)(N_{IT} PK_{IT})$, $h_{IG} = H_2(T_{IG}, ID_{IG}, m)$, $s_{IG} = c + H_{IG} + x_{IG}$, generate signature <$h_{\mathrm{IG}}$, s_{IG}>, and send message <$\mathrm{ID}_{\mathrm{IG}}$, N_{IG}, $\mathrm{PK}_{\mathrm{IG}}$, h_{IG}, s_{IG} > to users U and IT.

(4) As in Figure 4, after receiving the identity information <$\mathrm{ID}_{\mathrm{IG}}$, N_{IG}, $\mathrm{PK}_{\mathrm{IG}}$, h_{IG}, s_{IG} > from the IG, the user U and IT verify the signature of the IG, respectively. If user U and IT pass the authentication to the IG, then user U and IT send their identity information <ID_U, N_U, PK_U, h_U, s_U>, <$\mathrm{ID}_{\mathrm{IT}}$, N_{IT}, $\mathrm{PK}_{\mathrm{IT}}$, h_{IT}, s_{IT}>, respectively, to the IG.

(5) If the IG passes the authentication of user U and IT, it first forwards <$\mathrm{ID}_{\mathrm{IT}}$, N_{IT}, $\mathrm{PK}_{\mathrm{IT}}$, h_{IT}, s_{IT}> to user U and <ID_U, N_U, PK_U, h_U, s_U> to IT. Then, the user U and IT authenticate each other. Finally, after authentication, user U and IT can each obtain a common session key $K_S = K_{\mathrm{IG}} = K_{\mathrm{IT}} = K_U$ by calculation and use this key KS for secure communication in future communications. If the identity information cannot be authenticated, key negotiation needs to be performed again.

3.4. *Intelligent Terminal Identity Binding.*

IT needs to initiate the binding of its own ID to IG to generate a unique ITID and store it on the blockchain, and the specific IT binding process is shown in Figure 5:

(1) IT first hashes its identity $\mathrm{ID}_{\mathrm{IT}}$ with SHA256 function to get $H(\mathrm{ID}_{\mathrm{IT}})$, then encrypts it with the common session key K_S to get $E(K_S, H(\mathrm{ID}_{\mathrm{IT}}))$, and finally sends it to IG.

(2) The IG receives the message, decrypts it with the common session key K_S to get $D(K_S, H(\mathrm{ID}_{\mathrm{IT}}))$, and queries whether the device is in the device list.

(3) If present, a device unique identification ITID(i) is generated for it and stored on the blockchain. If it does not exist, it indicates that the device does not exist or is illegal and the identity binding is rejected.

3.5. *User Identity Binding.*

As with IT, users need to be bound to the ES to ensure their legitimacy. The specific user binding process is shown in Figure 6:

(1) User U sends <ID_U, N_U, PK_U, h_U, s_U> to ES for key negotiation.

(2) After receiving the message from U, ES verifies the user's signature and calculates $K_{ES-U} = g^{bx+x_b}(N_U PK_U)$, $h_{ES} = H_2(T_{IG}, m)$, and $s_{ES} = c + H_{ES} + x_{ES}$; m is the message to be signed. ES generates signatures <h_{ES}, s_{ES}> and send the information <$\mathrm{ID}_{\mathrm{ES}}$,$N_{\mathrm{ES}}$, $\mathrm{PK}_{\mathrm{ES}}$, h_{ES}, s_{ES}> to U.

(3) After receiving the signature information <N_{ES}, $\mathrm{PK}_{\mathrm{ES}}$, h_{ES}, s_{ES}> of the ES, U verifies the signature of the ES, and if user U passes the authentication of the signature of the ES, user U computes $K_{U-ES} = g^{cx+x_c}(N_{ES} PK_{ES})$. Finally, after

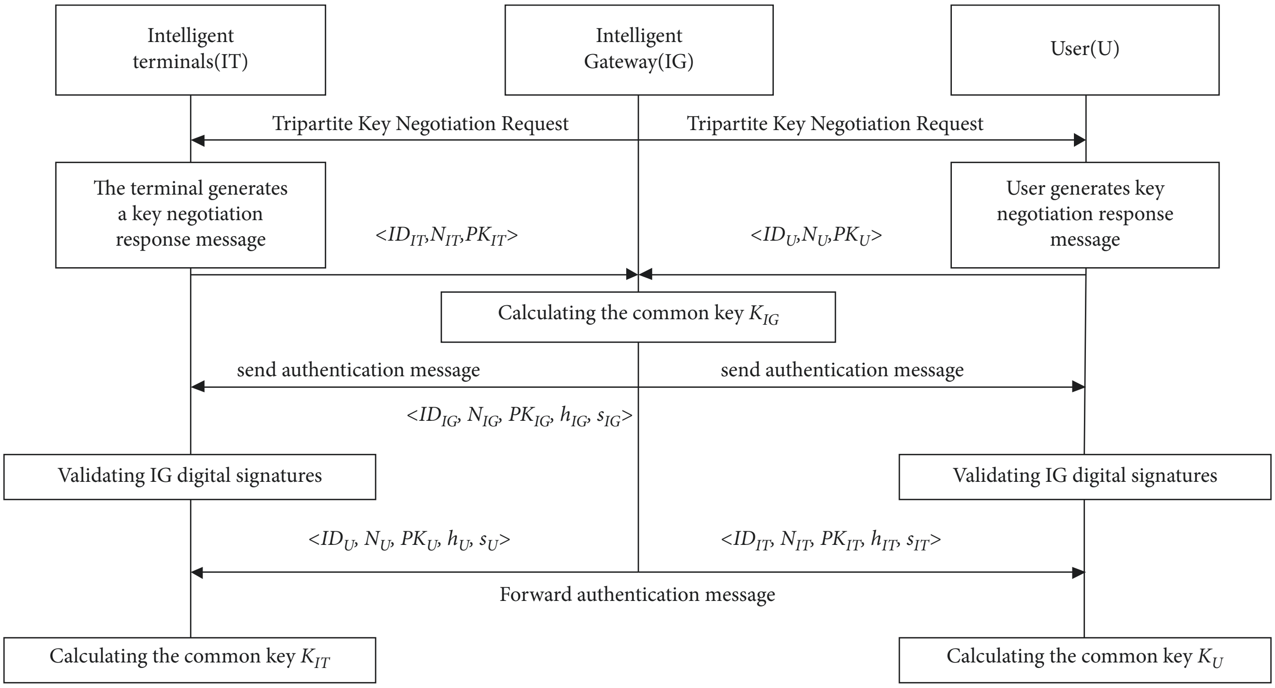
Intelligent terminals(IT)
Intelligent Gateway(IG)
User(U)
Tripartite Key Negotiation Request
Tripartite Key Negotiation Request
The terminal generates a key negotiation response message
<ID_{IT},N_{IT},PK_{IT}>
<ID_U,N_U,PK_U>
User generates key negotiation response message
Calculating the common key K_{IG}
send authentication message
send authentication message
<ID_{IG}, N_{IG}, PK_{IG}, h_{IG}, s_{IG}>
Validating IG digital signatures
Validating IG digital signatures
<ID_U, N_U, PK_U, h_U, s_U>
<ID_{IT}, N_{IT}, PK_{IT}, h_{IT}, s_{IT}>
Forward authentication message
Calculating the common key K_{IT}
Calculating the common key K_U

Figure 4: Tripartite key negotiation between IT, IG, and user U.

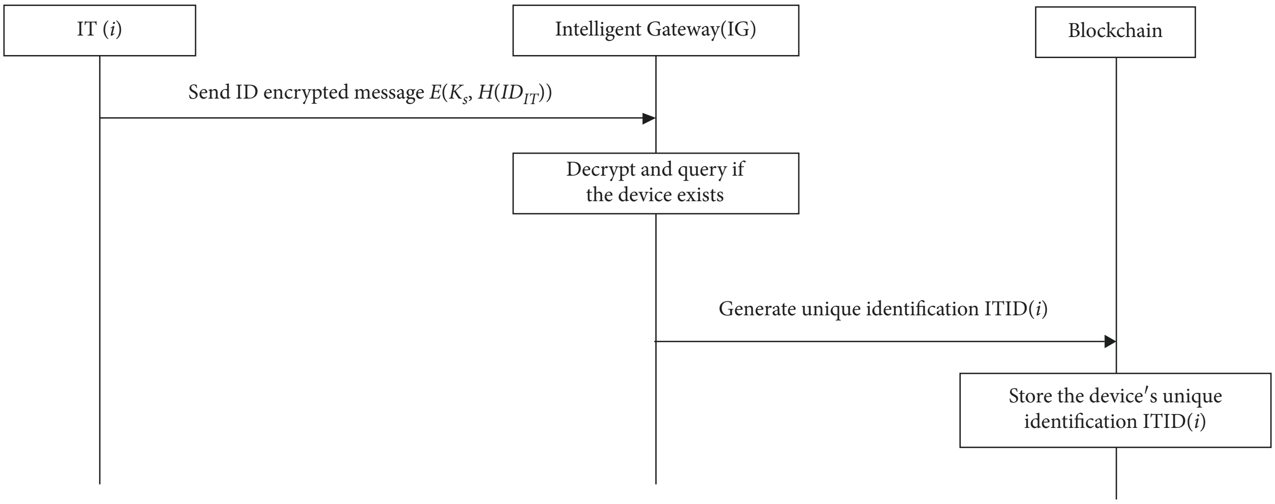
IT (i)
Intelligent Gateway(IG)
Blockchain
Send ID encrypted message $E(K_s, H(ID_{IT}))$
Decrypt and query if the device exists
Generate unique identification ITID(i)
Store the device′s unique identification ITID(i)

Figure 5: IT identity binding process.

authentication, the users U and ES can each obtain a common session key $K_{\mathrm{UE}} = K_{\mathrm{ES}} - U = K_U - E_S$ by calculation to use this key KUE for secure communication in future communications.

(4) The user chooses their identity ID_j to be hashed using the SHA256 function to obtain $H(\mathrm{ID}_j)$. Next, the encryption $E(K_{\mathrm{UE}}, H(\mathrm{ID}_j))$ is performed with the common session key KUE, and finally, this encrypted message is sent to the ES.

(5) After receiving the message, the ES decrypts $D(K_{\mathrm{UE}}, H(\mathrm{ID}_j))$ with the common session key K_{UE} and queries whether the user is a legitimate smart home family member.

(6) If legitimate, a user unique identifier UUID(j) is generated for it and added to the blockchain. If it is not legal, the user does not exist or is illegal and the identity binding is rejected.

3.6. User Authentication Access Mechanism. When a user needs to access an IT or access a specific service, user authentication needs to be completed. As shown in Figure 7, the specific authentication process is as follows:

(1) the user uses the APP input $U(j)$ to generate the user authentication request message r_U, which consists of an encrypted random number and signature

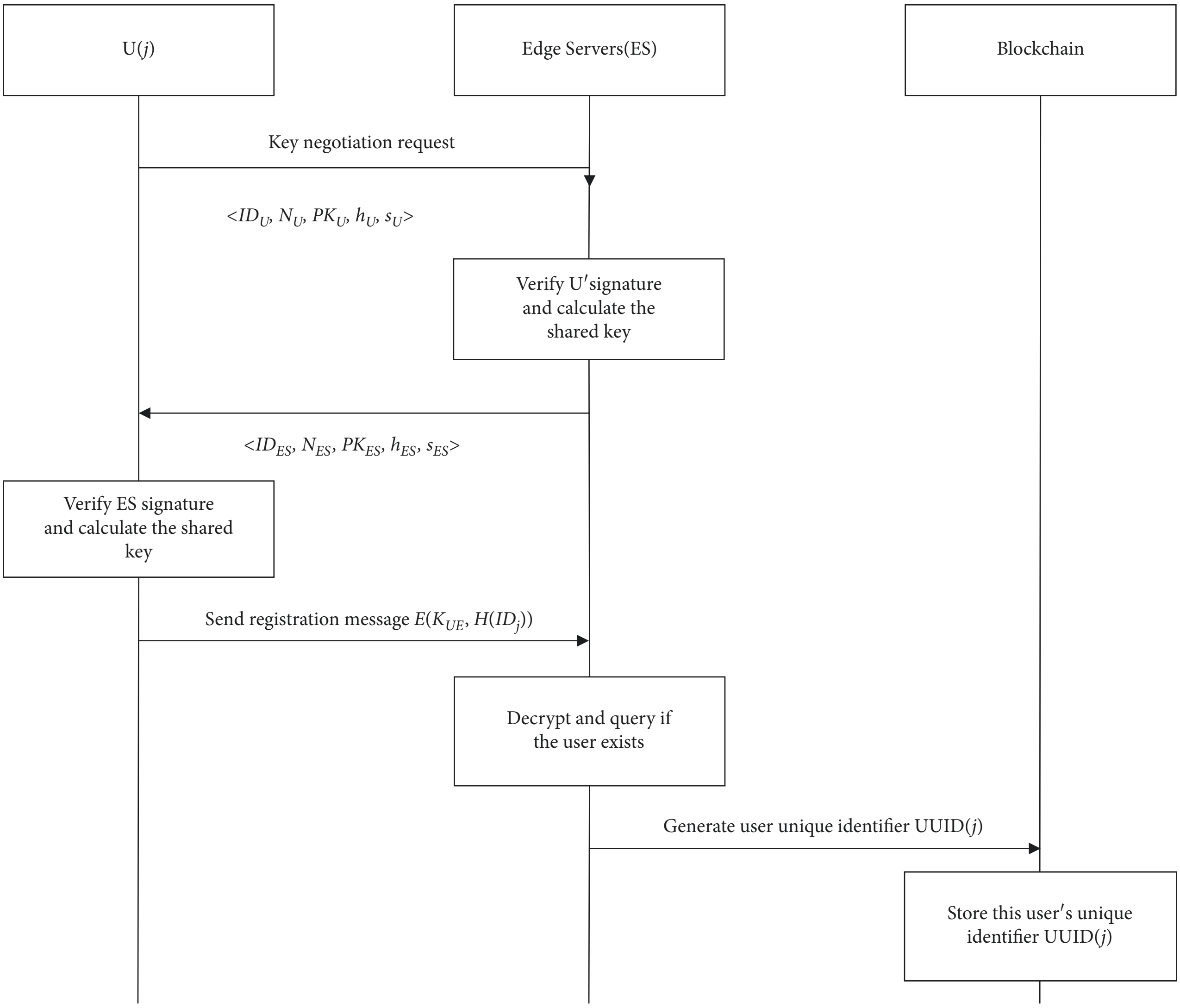
U(j)
Edge Servers(ES)
Blockchain
Key negotiation request
<ID_U, N_U, PK_U, h_U, s_U>
Verify U′signature and calculate the shared key
<ID_ES, N_ES, PK_ES, h_ES, s_ES>
Verify ES signature and calculate the shared key
Send registration message E(K_UE, H(ID_j))
Decrypt and query if the user exists
Generate user unique identifier UUID(j)
Store this user′s unique identifier UUID(j)

Figure 6: User identity binding process.

generated by IT(i). That is, $K_U(R_{U(j)}, Sig_U)$, where $R_{U(j)}$ is a random number, Sig_U is its own signature, and K_U is the key negotiated by $U(j)$ with ES or IG.

(2) After IG or ES receives the message, it verifies the signature and establishes a connection with the blockchain network, requesting to obtain its unique identifier UUID(j). The ES or IG verifies the existence of the user by decoding the function $f(\text{UUID}(j)) = H(U'(j))$, if $H(U'(j)) = H(U(j))$, then the user is authenticated; otherwise, the authentication is denied.

(3) After the user establishes the initial trust relationship with the IG or ES, the user sends an access request message to the IT.

(4) IT(i) verifies the user's signature after receiving the access request message and generates a response message and sends it to the IG. After the IG receives the message, it verifies the validity of IT(i); if it is valid, it forwards the message; otherwise, it rejects the request.

(5) After receiving the response message from IT(i), the user verifies the legitimacy of IT(i) and generates control information and sends it to IT(i). Users can perform control operations on IT(i) or encrypt data interactions.

3.7. IT Authentication Mechanism

3.7.1. Single Intelligent Terminal Authentication. In order to avoid illegal intrusion into the smart home network via IT devices, when a single IT device needs to upload data to the ES via IG, the legitimacy of the end device needs to be authenticated. As shown in Figure 8, the specific authentication process is as follows:

(1) IT(i) sends the authentication request message r_{IT} to the IG, which consists of an encrypted random number and signature generated by IT(i). That is, $K_{IT\text{-}IG}$ $(R_{IT(i)}, Sig_{IT})$, where $R_{IT(i)}$ is a random number, Sig_{IT} is its own signature, and $K_{IT\text{-}IG}$ is the key negotiated between IT(i) and IG.

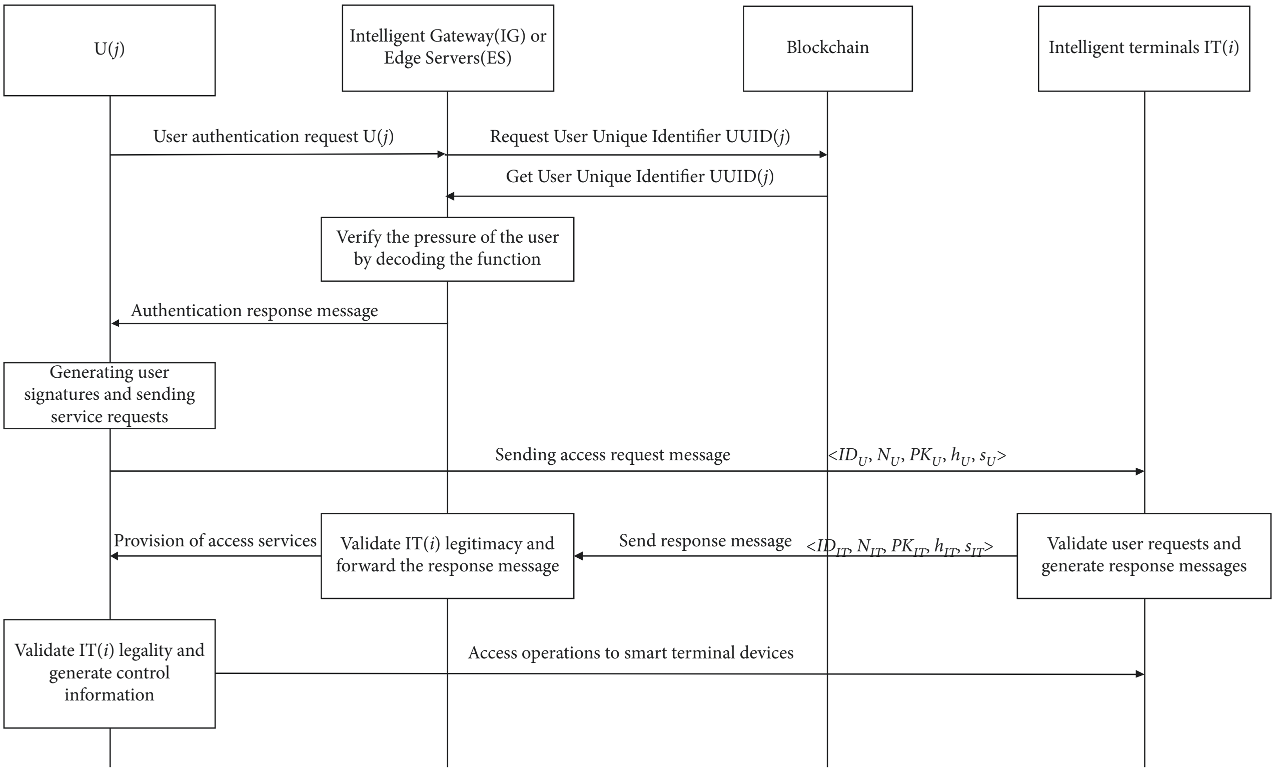
U(j)
Intelligent Gateway(IG) or Edge Servers(ES)
Blockchain
Intelligent terminals IT(i)
User authentication request U(j)
Request User Unique Identifier UUID(j)
Get User Unique Identifier UUID(j)
Verify the pressure of the user by decoding the function
Authentication response message
Generating user signatures and sending service requests
Sending access request message
<ID_U, N_U, PK_U, h_U, s_U>
Provision of access services
Validate IT(i) legitimacy and forward the response message
Send response message
<ID_{IT}, N_{IT}, PK_{IT}, h_{IT}, s_{IT}>
Validate user requests and generate response messages
Validate IT(i) legality and generate control information
Access operations to smart terminal devices

Figure 7: User authentication access process.

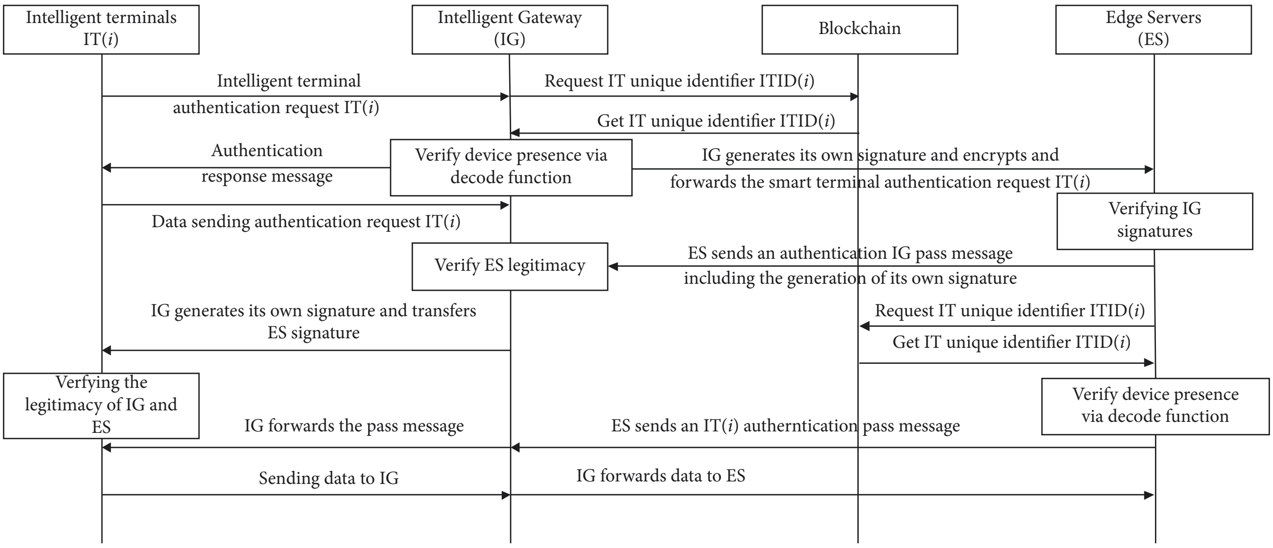
Intelligent terminals IT(i)
Intelligent Gateway (IG)
Blockchain
Edge Servers (ES)
Intelligent terminal authentication request IT(i)
Request IT unique identifier ITID(i)
Get IT unique identifier ITID(i)
Authentication response message
Verify device presence via decode function
IG generates its own signature and encrypts and forwards the smart terminal authentication request IT(i)
Verifying IG signatures
Data sending authentication request IT(i)
Verify ES legitimacy
ES sends an authentication IG pass message including the generation of its own signature
IG generates its own signature and transfers ES signature
Request IT unique identifier ITID(i)
Get IT unique identifier ITID(i)
Verfying the legitimacy of IG and ES
Verify device presence via decode function
IG forwards the pass message
ES sends an IT(i) authenrtication pass message
Sending data to IG
IG forwards data to ES

Figure 8: Single intelligent terminal authentication access process.

(2) The IG receives the message, verifies the signature, and establishes a connection to the blockchain network, requesting its unique identification ITID(i). Verify the existence of IT(i) by decoding function $f(\text{ITID}(i)) = H(\text{ID}'_{\text{IT}}\ (\text{i}))$, if $H(\text{ID}'_{\text{IT}}\ (i)) = H(\text{ID}_{\text{IT}}(i))$, then IG passes the consent authentication and sends an authentication response message to IT(i); otherwise, IG rejects authentication.

(3) The IG encrypts its own signature with the public key of the ES and sends it to the ES together with the authentication request of IT(i).

(4) After receiving the signature and authentication request from the IG, ES first verifies the IG signature. After the ES verifies the IG, it will generate its own signature and sends an IG authentication pass message to the IG; otherwise, authentication is

denied. Then, the ES establishes a connection with the blockchain network and requests for the unique identification ITID(i) of that IT(i). ES obtains its unique identifier ITID(i) and then verifies the existence of IT(i) by decoding the function $f(\mathrm{ITID}(i)) = H(\mathrm{ID}_{\mathrm{IT}}'(i))$. If $H(\mathrm{ID}_{\mathrm{IT}}'(i)) = H(\mathrm{ID}_{\mathrm{IT}}(i))$, then IT($i$) is authenticated by the ES, which generates an IT(i) authentication pass message and sends it to the IG; otherwise, authentication is denied.

(5) After receiving the signature and authentication pass message from the ES, IG verifies the legitimacy of the ES. If the ES is legitimate, IG will generate its own signature and sends it to IT(i) together with the signature of ES. Otherwise, authentication is rejected.

(6) After receiving the signature and authentication pass message from the IG, IT(i) verifies the legitimacy of the IG and ES. If the IG and ES are legitimate, and the IT(i) receives the ES authentication pass message forwarded by IG, the authentication is passed and data can be sent to the ES via the IG. Otherwise, authentication is rejected.

3.7.2. Multiple Intelligent Terminal Authentications. In multiple IT concurrent authentications, identity-based aggregated signatures are introduced to smart home network authentication. Compressing the signatures of any number of entities into one signature reduces the storage space of the signature and at the same time simplifies the authentication of multiple entities to the verification of one signature. In multiple IT concurrent authentications, identity-based aggregated signatures are introduced to smart home network authentication. Compressing the signatures of any number of entities into one signature reduces the storage space of the signatures and simultaneously simplifies the authentication of multiple entities to the verification of one signature. Compared with other authentication schemes, this scheme reduces the burden of data storage space under the premise of ensuring security. The data traffic of interactive messages and signalling is reduced, and the computational workload of verifying the signature is also reduced. It can improve the efficiency of smart home network certification when multiple IT certifications are concurrent. As shown in Figure 9, the specific authentication process is as follows:

(1) the IG sends an aggregated signature authentication request to all ITs in the smart home network. This group of endpoints generates their own signature $\sigma(i)$ using their own private keys and sends an aggregated signature authentication response message including this signature to the IG.

(2) After IG receives these aggregate signature authentication response messages, it queries whether these IT devices exist. If they exist, the unique identification ASID(s) of the multiterminal is generated and stored in the blockchain. If one or more terminals do not exist, the authentication will be refused, and only the unique identification ASID(s) of the legitimate terminal will be generated and stored in the blockchain.

(3) The IG generates the aggregated signature of the legitimate endpoint and encrypts it with the public key of the ES and sends it to the ES together with the IGs aggregated authentication request.

(4) After the ES receives the aggregated signature and authentication request sent by the IG, it first verifies the aggregated signature of the IG. After the verification is passed, the ES establishes a connection with the blockchain network and requests to obtain the unique identification ASID(s) of multiple ITs.

(5) The ES obtains its unique identification ASID(s) and verifies the existence of IT(s) by decoding the function $f(\mathrm{ASID}(s)) = H(\mathrm{ID}_{\mathrm{IT}}'\ (s))$. If $H(\mathrm{ID}_{\mathrm{IT}}'(s)) = H(\mathrm{ID}_{\mathrm{IT}}(s))$, then IT($s$) is authenticated by the ES, which generates its own signature and sends an authentication pass message to the IG. Otherwise, the authentication is rejected.

(6) After receiving the signature and authentication pass message from the ES, the IG verifies the legitimacy of the ES. If the ES is legitimate, it generates its own signature and forwards the authentication pass message; otherwise, it rejects the authentication.

(7) After receiving the signature and authentication pass message from the IG, IT(s) verifies the legitimacy of the IG and ES. If the IG and ES are legitimate, then the authentication is passed and data can be sent to the ES via the IG; otherwise, the authentication is rejected.

4. Security Analysis

Lemma 1. *Assume that the cryptographic problem of the elliptic curve used in the scheme is secure, and the single-item security is that the adversary cannot obtain the corresponding plaintext from the ciphertext when he does not know the user's private key, that is, the adversary u can obtain the inverse from the ciphertext. The probability is negligible under any polynomial algorithm A (Hypotheses 1 and 2).*

$$\mathrm{Succ}_{Z_p^*,A}^{DL} = \Pr\left[x_i \leftarrow A\left(g, g^{x_i}\right)\right], \tag{4}$$

$$\mathrm{Succ}_{Z_p^*,A}^{DL} = \Pr\left[r_i \leftarrow A\left(g, g^{r_i}\right)\right]. \tag{5}$$

It follows from Hypothesis 2 and the Lemma that (4) and (5) successfully solved are negligible under any polynomial algorithm A.

In theory, hackers can simultaneously intercept the public keys PK_a, PK_b, and PK_c and the random numbers R_a, R_b, and R_c of users A, B, and C in the channel." If an attacker wants to obtain the shared key K, he must obtain the master key x and any one of the random numbers r_a, r_b, and r_c at the same time. Taking user A as an example, the attacker wants to obtain the shared key, and other information can be obtained by intercepting the information, but he must also obtain the master key x and the random number r_a at the same time. The formula is as follows:

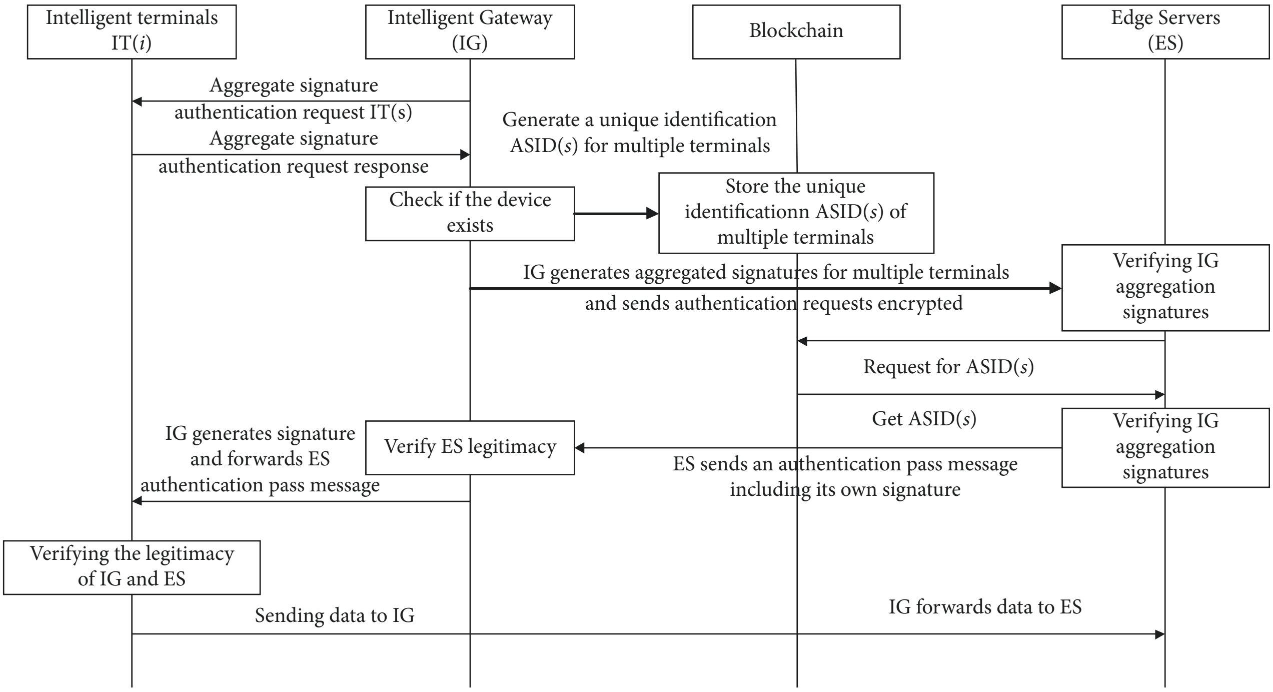
Intelligent terminals IT(i)
Intelligent Gateway (IG)
Blockchain
Edge Servers (ES)
Aggregate signature authentication request IT(s)
Aggregate signature authentication request response
Generate a unique identification ASID(s) for multiple terminals
Check if the device exists
Store the unique identificationn ASID(s) of multiple terminals
IG generates aggregated signatures for multiple terminals and sends authentication requests encrypted
Verifying IG aggregation signatures
Request for ASID(s)
Get ASID(s)
Verifying IG aggregation signatures
IG generates signature and forwards ES authentication pass message
Verify ES legitimacy
ES sends an authentication pass message including its own signature
Verifying the legitimacy of IG and ES
Sending data to IG
IG forwards data to ES

Figure 9: Multiple Intelligent terminal authentication access processes.

$$\mathrm{Succ}_{Z_p^*,A}^{C\ DH} = \Pr\left[g^{r_a x} \leftarrow A\left(g, g^{r_a}, g^{x}\right)\right]. \tag{6}$$

It follows from Hypothesis 1 and the Lemma that (6) successfully solved under any polynomial algorithm A is negligible. Therefore, the solution is monomial secure and the proof is over.

In addition to one-way security, the scheme also has security features such as antireplay attack, man-in-the-middle attack, denial of service attack, active attack, and passive eavesdropping.

4.1. Confidentiality. The user and IT use the shared key to encrypt the authentication request message to the IG or ES. When the IG or ES receives the authentication request, the shared key is used for decryption. Other communication entities cannot obtain the authentication because they do not have the decrypted shared key.

4.2. Freshness. The freshness of messages transmitted between users U, IG, and ES is guaranteed by random numbers. User U attaches a random number when sending an authentication data request, and IG and ES also attach a random number when sending an authentication response. By checking whether the received random number is the same as the one sent by itself, if it is different, the authentication fails. Therefore, it can be guaranteed that every authentication message sent has a freshness guarantee.

4.3. Unforgeability. When a forger pretends to be a signer u to forge a single signature σ_u=(U_u, V_u), although U_u can be calculated from the public key and random number, V_u cannot be calculated correctly because V_u=(h_u, R$_u$)Q$_{\mathrm{ID}u}$, and the forger does not know R_u, Q_{Idu}, and h_u, so it is impossible to forge σ_u to pass the signature authentication. According to the unforgeability of a single signature, while a forger can forge U, it cannot forge V so that it passes verification. Since the aggregated signature is jointly determined by multiple users, the counterfeiter needs to obtain the private key SK of all the signers to obtain the signature, which is obviously unrealistic. Even if the attacker obtains one or more private keys, it also cannot generate the correct aggregate signature without knowing the other private keys.

4.4. Antireplay Attack. When an attacker is replaying an old authentication data request message from a user or terminal, the IG or ES receives the message and verifies that the random number it contains is the same as its own current random number. If it is different, the authentication data request message is rejected. Similarly, when an attacker replayed an old authentication response message from IG or ES, it would be rejected by the user or endpoint.

4.5. Anti-Man-in-the-Middle Attack. When an attacker impersonates IG to eavesdrop and tamper with the authentication message between ES, IT, and users, it cannot be successful. Because the attacker does not know the private key of the ES, and there is signature authentication between the ES, the user, and the IG. If it cannot pass the signature authentication, the ES will reject the request when it receives the authentication data request message, and the user will also reject the request when it receives the authentication response message from the IG. Therefore, it can defend against man-in-the-middle attacks.

4.6. Resisting Distributed Denial of Service Attacks (DDos). The smart home authentication network is based on blockchain, which uses a distributed network. When a network communication node (such as IG) is attacked and the user cannot log in normally, the normal communication of other IGs is not affected. Therefore, it can effectively resist DDos.

Table 1 analyzes the authentication comparison between the three existing schemes and the newly proposed scheme. The scheme's authentication includes the following: ① ES to user authentication (ES-U): ② ES to IT authentication (ES-IT); ③ user to ES authentication (U-ES); ④ user to IT authentication (U-ES); and ⑤ IT to user authentication (IT-U).

From the above security analysis and the comparison of Tables 1 and 2, it can be concluded that the proposed solution has a higher level of security than these three existing solutions.

5. Performance Analysis

The smart home network architecture in this paper is built on the Ethernet platform. In the simulation testing and evaluation experiments, ES and IG were used as full nodes, and IT and users consisting of various sensors were used as light nodes. The Lenovo Yangtze P880 workstation was used as the Linux server. The configuration used for the ES was the Raspberry Pi 2020-02-05. IG uses a standalone virtual machine running the SDN controller in a Linux server. Smart contracts were coded using Solidity v0.5.0. The experimental environment is shown in Table 3.

5.1. Analysis of User Authentication. In the simulation evaluation, the proposed scheme is compared with the risk-based access control architecture Tyche from literature [11], the cloud and blockchain-based approach to smart home access control from literature [12], and the IoT authentication and access control approach from literature [13]. Tyche is a system that leverages the risk-asymmetry in physical device operations to limit the risk that apps pose to smart home users [11]. The risk-based groupings of device operations are defined in this system and applied to existing SmartThings apps. To counter the possible attacks on the gateway of smart homes, a blockchain-based smart home gateway network is proposed in literature [12]. To support decentralization and overcome the problem from traditional centralized architecture, this smart home network includes device, gateway, and cloud layers. The blockchain technology is employed at the gateway layer wherein data are stored and exchanged in the form blocks of blockchain. To monitor the activities that take place on particular data evidence, a novel Cloud framework based on Software Defined Network (SDN) is proposed in literature [13]. This framework includes IoT devices (100-mobile nodes), open flow switch, and blockchain-based controllers, cloud server, authentication server (AS), and investigator. In the mobile nodes, the packets are encrypted by using the ECIES algorithm and transferred to the cloud server.

Table 1: Comparison of authentication schemes.

Protocol	ES-U	ES-IT	U-ES	U-IT	IT-U
Tyche [11]	Y	N	Y	N	N
Literature [12]	Y	N	N	Y	N
Literature [13]	Y	Y	N	N	N
BCAS	Y	Y	Y	Y	Y

Table 2: Types of attacks that each of the compared schemes can resist.

Protocol	Replay	DDos	Man-in-the-middle	Confidentiality
Tyche [11]	Y	N	Y	N
Literature [12]	Y	N	N	Y
Literature [13]	Y	Y	N	Y
BCAS	Y	Y	Y	Y

Table 3: Experimental environment parameters.

Software/hardware	Parameter
Operating system	Ubuntu Linux 20.04LTS
CPU	Intel i7 10700 2.9 GHz
Programming languages	Solidity
Memory	16G
Raspberry Pi	4B+

The computational overhead mainly includes the encryption and decryption algorithms used for user registration and user authentication, signature authentication, hash functions, and decoding functions. The overhead calculated in the simulation experiments in this paper is the total time required for a user to initiate a registration and for the authentication to complete, as shown in Figure 10, and the simulation results are the average of 20 tests. It can be seen that the time spent in this scheme is less than in literature [11, 12] due to the shorter length of the ECIES key used and therefore less computational overhead. However, compared with the literature [13], the computational cost is slightly longer because the user authentications to IT and IT to user authentication are added to the user authentication scheme, which increases the computational cost but improves the security of authentication.

5.2. Analysis of IT Authentication. Intelligent terminal authentication simulation experiments use response time for performance analysis. The response time is the time taken from the initiation of an authentication request message to the receipt of an authentication pass message. Firstly, the response time in the case of a small number of concurrent authentication requests using a single intelligent terminal authentication method. As shown in Figure 11, the response time of schemes is increased as the number of IT increases. This is because as the number of nodes increases, the number of signatures to be verified increases and the computation delay becomes longer. Compared to other schemes, BCAS scheme has the shortest response time. This is because that the single IT authentication process is optimized in this

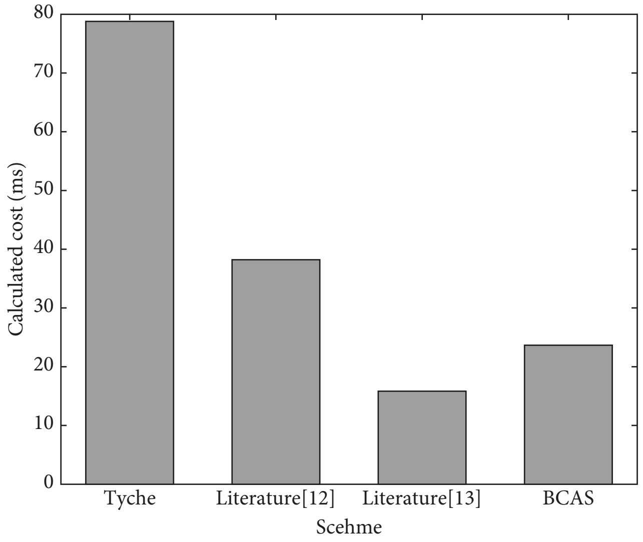

Figure 10: Comparison of user authentication calculation overheads.

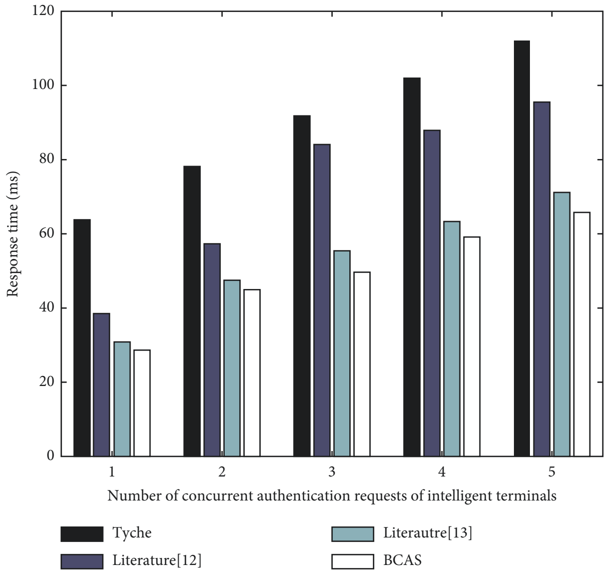

Figure 11: Comparison of response times when the number of IT concurrent is low.

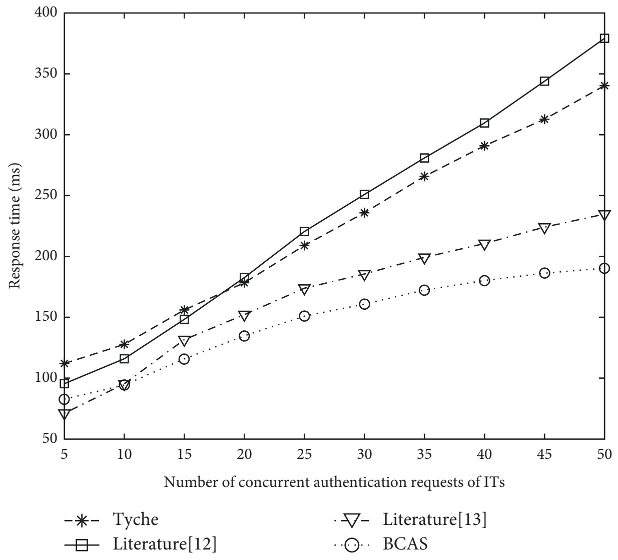

Figure 12: Comparison of response times when the number of IT concurrent is high.

paper. After the ES completes the IG authentication, it sends its own signature to the IG. Then, while the ES verifies the blockchain identity of the IT, the IG authenticates the effectiveness of the ES. This reduces the computational latency, which improves the response time. However, in Figure 11, the advantage of scheme BCAS is not obvious when the number of ITs is small, due to the fact that the computation time for signatures is a smaller proportion of the response time.

Figure 12 shows a comparison of the response time with different numbers of sensor nodes using the Tyche and BACS Literature [12, 13] schemes.

The response time of all schemes becomes longer as the number of sensor nodes increases. This is because as the number of nodes increases, the number of signatures to be verified increases and the computation delay becomes longer. In Figure 12, as the number of IT concurrent increases, the response time increases for all schemes. However, the BCAS method increases less quickly than the other methods. This is due to the fact that the BCAS scenario reduces the number of verification signatures when handling multiple IT certifications. With the increase of concurrent ITs authentication, BCAS has obvious advantages. It is due to the fact that the computation time of the signature occupies a large proportion of the response time. The average response time of BCAS is 12.03% lower than that of literature [13], 48.53% lower than that of Tyche, and 52.23% lower than that of literature [12]. Therefore, the proposed solution has greater advantages when the number of IT certifications increases dramatically (such as smart home network power outage restarts and IG disconnections).

6. Conclusions

As an important part of IoT applications, smart home networks can bring great convenience to family life. However, as remote users and the smart home network communicate through insecure channels, there are important security risks for sensitive information of users and intelligent devices. It is therefore necessary to design a secure and efficient authentication scheme to secure the communication. This paper proposes a certificateless authentication scheme for edge computing based on blockchain. The scheme has security features such as decentralization, unforgeability, resistance to replay attacks, and resistance to man-in-the-middle attacks. Performance simulation experiments show that this scheme has less computational overhead and shorter response time, especially in the case of high IT concurrent certification. At the same time, smart home data are stored in ES to reduce data security problems caused by IG being attacked. However, the proposed authentication scheme needs more control information. The computational cost of the BCAS method is still high for the ITs and users. In future research, the efficient and secure cryptographic signature methods will be considered to improve computational cost and optimize the authentication process. To reduce the response time, the offline computing will be introduced into the smart home authentication.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

The authors have read the manuscript and approved its submission.

Acknowledgments

This work was partly supported by National Natural Science Foundation of China (Grant no. 62002180), Natural Science Foundation of Henan Province (Grant no. 202300410301), General Project of Humanities and Social Sciences Research of Henan Institutions of Higher Learning (Grant No. 2021-ZZJH-262), the Scientific and Technological Project in Henan Province of China (Grant nos. 212102310481, 222102320369, and 212102210169), and the Key Scientific Research Projects of Colleges and Universities in Henan Province (Grant nos. 21A520033 and 22A520037).

References

[1] C. Liu, Z. F. Xu, and F. Q. Wang, "Application framework design of the Internet of Things operating system for smart appliances," *Modern Electronics Technique*, vol. 43, no. 23, pp. 143–145, 2020.

[2] P. T. Song, C. Li, and L. T. Xu, "Edge computing system for smart home based on personal computer," *Computer Engineering*, vol. 43, no. 11, pp. 1–7, 2017.

[3] M. Nasir, K. Muhammad, A. Ullah, J. Ahmad, S. Wook Baik, and M. Sajjad, "Enabling automation and edge intelligence over resource constraint IoT devices for smart home," *Neurocomputing*, vol. 491, pp. 494–506, 2022.

[4] N. Guo, C. Zhao, and T. Gao, "An anonymous authentication scheme for edge computing-based car-home connectivity services in vehicular networks," *Future Generation Computer Systems*, vol. 106, pp. 659–671, 2020.

[5] X. D. Hu and R. S. Zhao, "A lightweight mutual authentication protocol for smart home," *Chinese Journal of Sensors and Actuators*, vol. 29, no. 5, p. 7, 2016.

[6] B. Mbarek, M. Ge, and T. Pitner, "Trust-based authentication for smart home systems," *Wireless Personal Communications*, vol. 117, no. 3, pp. 2157–2172, 2021.

[7] Y. Guo, Z. Zhang, and Y. Guo, "SecFHome: secure remote authentication in fog-enabled smart home environment," *Computer Networks*, vol. 2022, no. 207, 7 pages, Article ID 108818, 2022.

[8] W. Liu, X. Wang, and W. Peng, "NCZKP based privacy-preserving authentication scheme for the untrusted gateway node smart home environment," in *Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, Rennes, France, July 2020.

[9] P. Kumar and L. Chouhan, "A secure authentication scheme for IoT application in smart home," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 420–438, 2021.

[10] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.

[11] A. Rahmati, E. Fernandes, K. Eykholt, and A. Prakash, "Tyche: risk-based permissions for smart home platforms," 2021, https://arxiv.org/pdf/1801.04609.pdf.

[12] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, "A blockchain-based smart home gateway architecture for preventing data forgery," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, p. 9, 2020.

[13] P. Velmurugadass, S. Dhanasekaran, S. Shasi Anand, and V. Vasudevan, "Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm," *Materials Today Proceedings*, vol. 37, no. 2, pp. 2653–2659, 2021.

[14] L. H. Zhang, H. Z. Zhang, and Y. Cao, "Authentication and access control scheme for smart home based on blockchain," *Application Research of Computers*, vol. 39, no. 3, pp. 863–867, 2022.

[15] Z. Chen, W. Hou, H. Wen, W. Lei, and H. Lin, "Multi-dimensional resource management system based on blockchain and cybertwin," in *Proceedings of the The 2nd International Conference on Computing and Data Science(CONF-CDS 2021)*, ACM, Stanford CA, USA, June 2021.

[16] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," *Information Processing & Management*, vol. 58, no. 1, Article ID 102397, 2021.

[17] A. Joux, "A one-round protocol for tripartite Diffie-Hellman," in *Proceedings of the 4th International Algorithmic Number Theory Symposium(ANTS-IV)*, pp. 385–394, Springer-Verlag, London, UK, July 2000.

[18] Q. Y. Huang, Z. Y. Li, and W. T. Xie, "Edge computing in smart homes," *Journal of Computer Research and Development*, vol. 57, no. 9, pp. 1800–1809, 2020.