IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2023, accepted 30 June 2023, date of publication 7 July 2023, date of current version 12 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3293401

RESEARCH ARTICLE

Consortium Blockchain Based Lightweight Message Authentication and Auditing in Smart Home

BAI LIU, XUEYAN YAO, KUIKUI GUO, AND PENGDA ZHU
College of Computer Science, Hubei University of Technology, Wuhan 430068, China

Corresponding author: Bai Liu (liubai@hbut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62002105, Grant 62072134, Grant U2001205, and Grant 61902116; and in part by the Key Research and Development Program of Hubei under Grant 2021BEA163.

ABSTRACT Currently, smart homes rely heavily on wireless sensor networks (WSNs), which typically consist of wireless sensor nodes with limited resources and are scattered throughout the network. This topology makes them vulnerable to packet sniffing, spoofing, and other malicious attacks, which can result in the leakage of private data collected by devices. Additionally, managing the device key for the entire system becomes difficult as more devices are added. Moreover, the centralization of current cloud service management in smart home systems poses a serious single-point-of-failure problem, and the private data of cloud outsourcing cannot receive strict privacy supervision, ultimately relying entirely on the trust of enterprises. To address these issues, this paper proposes using a consortium blockchain and InterPlanetary File System (IPFS) instead of the existing centralized structure. An improved pairing-free certificateless aggregated signature(CLAS) scheme ensures the security of message authentication and solves the device key management problem. Our scheme reduces the computational and communication overheads at the device side by 50% and 25%, respectively, compared with existing schemes in WSNs. The overall computational overhead is also reduced by 28.6%, making it more suitable for smart home scenarios. Additionally, we use an auditing method based on Merkle root hash verification to ensure the reliability of data storage in IPFS.

INDEX TERMS Smart home, message authentication, CLAS, consortium blockchain.

I. INTRODUCTION

The rapid development of the Internet of Things (IoT) has brought traditional industries and manufacturing into an intelligent stage. Various IoT technologies can be widely used in digital healthcare, intelligent transportation, and smart home [1]. Smart home, as one of the important applications of Internet of Things technology, uses computer technology, network technology, cloud computing, intelligent control, and other technologies to connect smart home devices. Then, the whole system can be connected to the Internet through the family smart gateway. Fig 1 shows the complete architecture diagram of the smart home. Users can control smart home devices remotely through mobile applications and other methods with Internet access. Users can also upload data collected by smart meters, indoor environmental sensors, and other devices to the data management platform. Through big data analysis, the indoor environment can be intelligently adjusted. As a result, the smart home can provide us with a safe, comfortable, and intelligent living environment [2]. However, the wireless connections between wireless sensors and data collection nodes in smart homes are usually unreliable, and devices in the smart home are less resistant to malicious attacks. Once a node involving user privacy is caught, the user's private data can easily be leaked [3]. Therefore, guaranteeing data authenticity and integrity is crucial. In order to improve the security of private data in smart homes, we proposed a message authentication scheme to secure smart home network communication.

The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mueen Uddin.

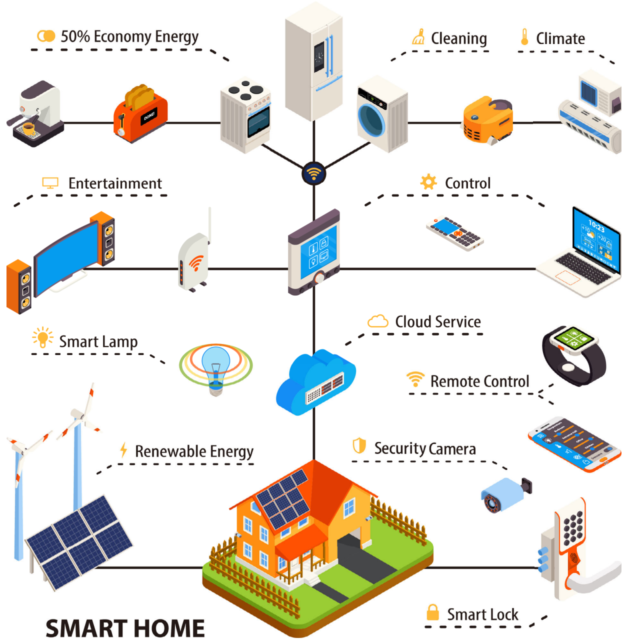


FIGURE 1. **Smart home architecture diagram.**

Simultaneously, it should also note that the storage capacity of smart home wireless sensors is weak, but the amount of collected data is increasing rapidly [4]. For example, home security surveillance is constantly collecting video data, which will require large storage space. But many homes cannot deploy high-cost network-attached storage (NAS). So the traditional smart home solutions usually use the supporting centralized cloud storage service provided by the equipment provider. Users will create a high dependence on the centralized cloud server [5], which will also cause a single point of failure and data security issues. As a semi-honest entity, the enterprise cloud server may delete cold data with low user access to reduce the pressure of storage on the cloud server [6], thus lowering costs and hiding the data corruption, which would compromise the value or security of user data. In order to solve the data storage problem, we decided to use the consortium blockchain to replace the centralized cloud server, and finally store the data on the IPFS system based on distributed hash table technology [7]. However, there are still challenges in applying blockchain to the smart home environment [8]. It is difficult for the device to manage its identity as a node on the chain. For message authentication, miners need to verify the device's identity while guaranteeing its anonymity. The traditional public key cryptography system can ensure the security of the public key through certificates, but it is difficult to use time-consuming certificate management for smart homes. Cleverly, the certificateless cryptosystem can perfectly complement the blockchain and be applied to the smart home environment. The device and the KGC jointly negotiate to generate a public-private key pair, which avoids the key escrow problem in the identity-based public key cryptosystem [9]. And blockchain can also help devices broadcast public keys.

Finally, we improved Li et al. [10]'s cloud data auditing scheme and used it for our message authentication scheme. The scheme implements lightweight auditing based on Merkle tree root hash verification. However, in their scheme, the cloud server can access on-chain data labels when generating proofs, thereby deceiving auditors. Due to the channel isolation mechanism of the Fabric consortium block, we upload the data tags in a separate channel, preventing IPFS nodes from stealing tags when generating audit proofs. The consortium blockchain can also re-audit the audit results according to the smart contract to further improve data integrity in the cloud. By reading the related literature, we find that there exists a large amount of research on IoT data privacy protection in wireless medical sensor networks and vehicle networking, while the smart home domain focuses more on the design of device identity authentication schemes. Meanwhile, the single-point-of-failure problem of devices in the smart home environment relying on third-party centralized servers is always unsolved. Therefore, we design a more suitable data privacy protection scheme for the smart home environment, which not only solves the dependency problem but also significantly reduces the computational overhead on the device side.

Our contributions are as follows:

- A pairing-free certificateless aggregation signature-based message authentication scheme without bilinear pairs is designed to reduce the computational overhead and communication overhead of smart home wireless sensor devices.
- We replaced the centralized server in the traditional smart home with blockchain and stored the data in IPFS based on distributed hash table technology. It solved the problem of users' high dependence on it.
- We used a lightweight auditing method based on Merkle tree root hash verification to guarantee the integrity of user data in IPFS. In addition, smart contracts re-audit audit results to prevent malicious auditors.

The paper organization is as follows. *Section II* presents previous related work. *Section III* describes some background knowledge used in this paper. *Section IV* presents the system model, threat model and problem statement. *Section V* defines our purposed scheme. *Section VI* presents the security analysis and performance evaluation. Conclusions are discussed in *Section VII*.

II. RELATED WORK

In smart home networks, protecting the data integrity and immutability of users' private data during transmission is the key issue. In recent years, a large number of researchers have proposed various identity authentication protocols and message authentication protocols to secure smart home networks.

Shuai et al. [11] proposed an efficient anonymous authentication scheme for smart homes based on elliptic curve cryptography and proved that the scheme can effectively resist replay attacks and clock synchronization problems.

Mezrag et al. [12] proposed an identity-based authentication and key agreement scheme that combines elliptic curve cryptography and identity-based public key cryptography to establish secret session keys over insecure channels. Pirayesh et al. [13] proposed a device authentication and key negotiation scheme in smart home networks that combines physical layer security techniques with hyperelliptic curve cryptosystems and claims to resist man-in-the-middle attacks, replay attacks, and desynchronization attacks. These papers [11], [12], [13] all use public-key cryptosystems to authenticate smart home user devices, but most smart home devices are based on wireless sensor networks, and these devices are resource-constrained, and the computational overhead of asymmetric cryptosystems is too large; deploying symmetric key cryptosystems will be more lightweight than asymmetric cryptosystems. Poh et al. [14] proposed PrivHome, a smart home privacy protection scheme based on symmetric cryptography, which contains a lightweight entity and key exchange protocol and an efficient searchable encryption protocol that can support authentication and query of secure data data in smart home systems. However, symmetric encryption systems are still computationally overloaded for resource-constrained sensors, so researchers are continuously working on more lightweight authentication protocols. Xiang et al. [15] proposed an efficient device authentication scheme in smart home systems using the situational awareness features of smart home systems, which can select a suitable authentication protocol based on the security risk information assessed by the system. However, Oh et al. [16] proved that the protocol of [15] has the risk of session key leakage and cannot ensure secure mutual authentication, based on [16] proposed a new secure lightweight smart home authentication protocol and proved that the protocol can effectively resist various attacks such as session key leakage, replay, and MITM. Banerjee et al. [17] proposed an efficient, anonymous, and robust smart home authentication scheme based on hash functions, heterogeneous operations, and fuzzy extractors, and proves that the protocol can effectively resist risks such as replay attacks, man-in-the-middle attacks, and impersonation attacks. However, AL-Turjman et al. [18] pointed out that the protocol of [17] cannot provide identity protection, authentication traceability, and interactive session key agreement. Kaur et al. [19] proposed a two-factor smart home based anonymous authentication protocol, however, Yu et al. [20] analyzed the scheme of [19] and finds that there is also a risk of session key leakage, and proposes a lightweight three-factor privacy-preserving authentication scheme for smart homes based on this. Nimmy et al. [21] also proposed a lightweight smart home remote user authentication protocol based on optical response nonuniformity, which is suitable for deployment in heterogeneous and resource-constrained smart home networks. In addition to authentication schemes, Liu et al. [22] proposed an SM9-based smart home message authentication scheme, Kar et al. [23] proposed an identity-based message authentication scheme in wireless sensor networks, and Kar et al. [24] proposed a certificate-free aggregated signature message authentication scheme in wireless sensor networks to reduce the overhead of data transmission. However, the bilinear pairing computation overhead in [22] and [24] is difficult to adapt for smart home devices, and the identity-based scheme in [23] inevitably suffers from key management problems. Thus Zhou et al. [25] proposed a certificate-free aggregated signature message authentication scheme based on bilinear pairs in wireless medical sensor networks. Other researchers have focused on the secure storage of private smart home data, and Ren et al. [26] proposed the use of a blockchain-based multi-cloud storage mechanism to securely store smart home data. For efficiency, the scheme uses identity-based agent aggregation signatures. However, the key hosting issues of the IBE scheme remain unresolved. And the scheme cannot achieve balanced data storage among multiple cloud providers or guarantee the integrity of data in the cloud. Li et al. [27] A smart contract-based framework for secure access control of smart home data is proposed. Li et al. [10] designed a new public audit solution for cloud data using blockchain technology, which generates lightweight tags for data before it is uploaded and stored to the blockchain. These tags are used to construct Merkle tree root hashes during auditing and generate proofs to verify the integrity of the data in the servers in the cloud server. However, we found that this scheme has the risk of tag leakage, and the cloud server generating the proof can generate the correct proof by accessing the tags stored on the chain, thus masking data loss. After analyzing various literature on smart home privacy protection, we find that almost all current work does not focus on the current over-centralized system architecture of smart homes, so we propose a unified design to secure smart home data and improve cloud data security based on [10].

III. BACKGROUND

A. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM(ECDLP)

Given a prime order finite field $\mathbb{F}_q(q > 3)$, an elliptic curve E over $\mathbb{F}_q$ is defined as $E/\mathbb{F}_q$, where x and y are the solution of the equation $E : y^2 = x^3 + ax + b(\mathit{mod}\ q\ )$, $a, b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0(\mathit{mod}\ q\ )$. The point $P = (x, y)$ on elliptic curve $E/\mathbb{F}_q$ together with an extra point O called the point at finfinity from a group

$$G = \{(x, y) : x, y \in \mathbb{F}_q, E(x, y) = 0\} \cup \{O\}.$$

For the elliptic curve already given, existence of the given $P, Q \in G$, the elliptic curve discrete logarithm problem (ECDLP) is to find an integer $z \in Z_q^*$, such that $Q = zP$.

B. MERKLE HASH TREE

Fig 2 shows a binary Merkle hash tree containing four leaf nodes, the leaf nodes are usually data objects to be stored, the internal nodes are the hashes connected to their leaf nodes, and the root node of the tree is called the root hash. In the field of information security, Merkle hash trees are often used

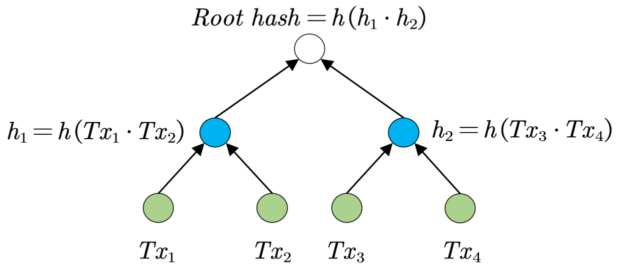

$Root\ hash = h(h_1 \cdot h_2)$
$h_1 = h(Tx_1 \cdot Tx_2)$
$h_2 = h(Tx_3 \cdot Tx_4)$
Tx_1
Tx_2
Tx_3
Tx_4

FIGURE 2. A binary Merkle hash tree.

to store large amounts of data in untrusted memory using small-capacity trusted memory. The data in the block is also organized through a Merkle hash tree. Based on its overall structure and characteristics, it is known that any change in the underlying values will cause a large number of value changes in the tree, so tampering of transactions can be easily detected, and we also use its characteristics to verify the integrity of remote data.

C. CONSORTIUM BLOCKCHAIN

Blockchain technology was first proposed in the Bitcoin system and aimed to achieve a decentralized system with consistently distributed data without relying on trusted third parties. Blockchains are divided into three major categories according to the degree of decentralization: public blockchain, consortium blockchain, and private blockchain. Consortium blockchain, as semi-open blockchains, provides a certain level of controlled access, which can only be accessed by authorized nodes and is more suitable for real-life application scenarios. Our solution is designed based on Fabric Consortium Blockchain, using the channel in Fabric to achieve the isolation between the two businesses of storage and auditing, while Fabric Consortium Blockchain can also deploy chain codes to implement application business logic.

D. IPFS

IPFS (InterPlanetary File System) is a distributed file management system that uses decentralized sharded encrypted storage technology to split files into multiple segments that are stored in various nodes on the network, with the files generating unique hash values as addressing addresses. Since IPFS is based on content addressing and there is a redundant backup mechanism in the distributed system, it can effectively resist a certain level of security attacks, and the system can normally work, even if individual nodes have problems.

E. CERTIFICATELESS AGGREGATION SIGNATURE SCHEME

Generally, a CLAS scheme consists of the following eight probability polynomial-time (PPT) algorithms.

1) **MasterKeyGen:** According to the system security parameters κ, the key generation center (KGC) generates the system parameters *parameter* that need to be disclosed and the system master private key *msk* that needs to be kept by itself.
2) **PseudoIDGen:** Input the real identity MAC_i of a smart device SD_i. This algorithm outputs a pseudo identity ID_i for SD_i. Using pseudo identity can prevent the real identity of the device from being leaked.
3) **PartialKeyGen:** KGC uses system parameters *params*, system master private key *msk*, and the pseudo identity ID_i of a smart device SD_i to generate a partial private key β_i of the device.
4) **UserKeyGen:** For the identity ID_i of a Smart Device SD_i. This algorithm generates a public/secret key pair (pk_i, sk_i) for SD_i.
5) **SignGen:** According to the pseudo identity ID_i, user's private key sk_i, partial private key β_i of a smart device SD_i and the data m_i to be signed, the user generates a signature σ_i on the data.
6) **SignVerify:** For a single signature σ_i that has been generated, this algorithm uses the user's public key pk_i of ID_i to verify that the signature is correct.
7) **AggregateSign:** Input the set of single signatures $\{\sigma_i, i = 1, \ldots, n\}$, and the set of data $\{m_i, i = 1, \ldots, n\}$ for n users, the algorithm generates an aggregate signature σ.
8) **AggregateVerify:** Input the public keys $\{pk_i, i = 1, \ldots, n\}$ of n users $\{ID_i, i = 1, \ldots, n\}$ and the aggregated signature σ_i of the data sets $\{m_i, i = 1, \ldots, n\}$ in the same state, this algorithm verifies that the aggregated signature is correct.

IV. SYSTEM MODEL AND PROBLEM STATEMENT

A. SYSTEM MODEL

Our blockchain-based model for smart home data privacy protection is shown in Fig 3 and contains two functional modules, i.e., smart home privacy data security storage and data integrity audit, in which the following entities exist.

1) Smart Devices: In our proposed scheme, smart home devices are the main body of data collection, and they are mostly sensor devices embedded in smart homes, responsible for collecting data such as video surveillance, indoor air quality, temperature, and humidity.
2) Family Smart Gateway: The Home Smart Gateway is an edge network node, at least one of which exists in every home, through which users can control and manage all their devices. The Home Smart Gateway is mainly responsible for collecting data from uploaded sensor devices and assumes the aggregation and verification of certificateless signatures to reduce the computational burden of the devices.
3) IPFS: InterPlanetary File System is a new hypermedia text transfer protocol, IPFS network storage files, using decentralized fragmented encrypted storage technology, which splits the files into multiple pieces and stores them on various nodes of the network, We use it to replace the centralized trusted server as the outsourced storage for smart home data, solving the single

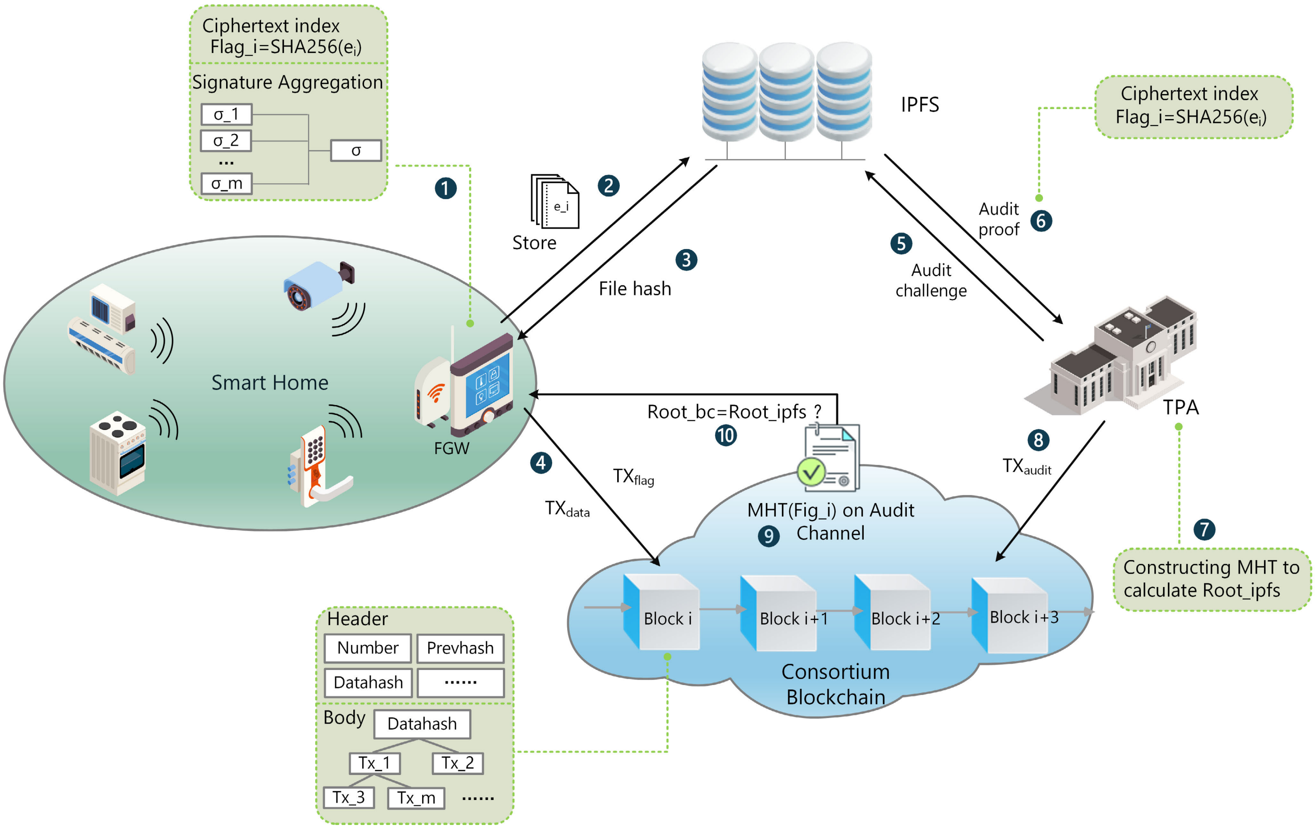


FIGURE 3. System model architecture diagram.

point of failure problem of the traditional centralized trusted server.

4) TPA: Third-party auditing, as a necessary entity in data integrity auditing, is mainly responsible for auditing challenges of data blocks in IPFS according to audit requirements in our system, and uploading the obtained audit results to smart contracts for secondary checking.

B. NOTATION

Throughout this article, some basic notations and their descriptions are shown in Table 1.

C. THREAT MODEL

In the threat model, we consider the potential threats faced by the message authentication scheme and the data auditing scheme in their implementation, respectively.

In the message authentication scheme, we use a pairing-free certificateless signature aggregate scheme to implement, according to the security model described in the literature [28] and [29], the signature should be able to resist both types of adversaries in the certificateless signature scheme.

Type I: Malicious device masquerader adversary $\mathcal{A}_I$, as an external adversary, can replace the public key of the device at will, but cannot obtain the master key of the system or the private key of the device part.

Type II: Malicious but passive semi-trusted key generation center $\mathcal{A}_{II}$, as an internal adversary, has access to the system's master key and the user's partial private key, but cannot replace the device's public key.

Also, the aggregated signature should be able to resist fully selective key attacks, where an attacker cannot generate a valid aggregated signature using an invalid individual signature aggregate, even if he has the private keys of all devices.

In the data auditing scheme, since we use the Merkle hash tree root hash verification scheme under the provable data possession model to achieve integrity auditing of lightweight cloud data, we need to ensure that the original data labels for constructing Merkle trees are not stolen by any verifier.

D. PROBLEM STATEMENT

Our design goal is to achieve data privacy protection for smart home environments, which should satisfy the following design objectives.

- **Lightweight devices:** For the network environment of smart home devices with low computing power, the computing involvement of the devices should be as lightweight as possible.
- **Anonymity:** In the proposed scheme, the true identity of the device is not available to other entities except the device itself.

TABLE 1. Basic notation.

Symbols	Description
κ	Security parameter
q	Prime
$\mathbb{Z}_q$	Finite field
$\mathbb{Z}_q^*$	$\mathbb{Z}_q/\{0\}$
E	Elliptic curve
G	Cyclic group of order q
$params$	System parameters
H_0, H_1, H_2, H_3	Cryptographic hash functions
SD $_i$	Smart Devices sensor node
RID $_i$	Real identity of SD$_i$
ID $_i$	Pseudo identity of SD $_i$
(α_i, β_i)	Private key of SD_i
(A_i, B_i)	Public key of SD_i
FGW	Family Smart Gateway
IPFS	InterPlanetary File System
TPA	Third-party auditor
CLAS	Certificateless aggregation signature
KGC	Key generation center
(Λ_E, λ_E)	IPFS node public-private key pairs
(F_E, T_E)	Public keys for FGW and TPA
$\mathcal{A}_I, \mathcal{A}_{II}$	Adversary I and Adversary II respectively
$\mathcal{C}_I, \mathcal{C}_{II}$	Challenger I and Challenger II respectively
σ	Signature
(P_{pub}, s)	Public and private key pair of KGC

- **Efficiency:** A large amount of device privacy data needs to be uploaded online in real-time, so efficient signature verification is also very important.
- **Accountability:** All data storage records and audit records are tamper-proof and traceable to accountability.

V. PROPOSED SCHEME

This section describes our Smart Home Data Privacy Protection Act solution combining blockchain and IPFS, which has two phases: the data storage phase and the data audit phase. The data storage phase uses the CLAS algorithm to generate and aggregate signatures when storing data to address the lack of scalability of the blockchain, and the use of CLAS can overcome the certificate management problem of multiple devices in the IoT environment. In the data auditing phase, we divide the auditing task between smart contracts and third-party auditing to solve the reliance on third-party trusted auditing in traditional auditing schemes.

A. SETUP

Performed by KGC to complete system initialization.

1) Input the security parameter κ, selects an elliptic curve additive group G of large prime number $q > 2^\kappa$ and a generator P of the group G.
2) Randomly select a value $s \in \mathbb{Z}_q^*$ as the system master key, and sets $P_{\mathrm{pub}} = sP$.
3) Define four hash functions H_0, H_1, H_2, H_3, where $H_0 : G \times \{0,1\}^* \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \times G \times G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G \times G \times G \rightarrow Z_q^*$, and $H_3 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \rightarrow Z_q^*$.
4) Publish the system parameter and keep s in secret, $params = \{G, q, P, P_{\mathrm{pub}}, H, H_1, H_2, H_3\}$.

B. KeyGen

Performed by KGC and smart devices to generate public-private key pairs for the device.

1) SD_i randomly chooses value $\alpha_i \in Z_q^*$ as it's secret value, calculates $A_i = \alpha_i P$.
2) SD_i uses its real identity RID_i to generate a pseudo-identity $ID_i = H_0\left(\alpha_i P_{\mathrm{pub}}, T_i\right) \oplus RID_i$, where T_i denotes the corresponding pseudo-identity validity time period, and sends (ID_i, A_i, T_i) to the KGC.
3) KGC selects a random value $r_i \in Z_q^*$ and calculates $B_i = r_i P$, $h_i = H_1(ID_i, A_i, B_i, P_{\mathrm{pub}})$, and $\beta_i = r_i + s \cdot h_i \bmod q$.
4) KGC sends B_i as a partial public key and β_i as a partial private key to SD_i over a secure channel.
5) SD_i verifies the valid of the partial key by checking whether
$$\begin{aligned} \beta_i P &= r_i P + h_i s P \\ &= B_i + h_i P_{\mathrm{pub}} \end{aligned}$$
holds.
6) The secret key of the SD_i with ID_i is set as $SK_i = (\alpha_i, \beta_i)$, and the corresponding public key is set as $PK_i = (A_i, B_i)$.

C. SignGen

Performed by smart devices to generate signatures for a given data m_i.

1) Collects the data m_i and encrypts the data into ciphertext e_i using the public key PK_i.
2) Randomly choose value $v_i \in Z_q^*$, and calculate $V_i = v_i P$.
3) Calculate $x_i = H_2(ID_i, V_i, PK_i, P_{\mathrm{pub}})$, $y_i = H_3(ID_i, e_i, V_i, PK_i, P_{\mathrm{pub}})$ and $\varphi_i = x_i v_i + y_i(\alpha_i + \beta_i)$.
4) Output a signature $\sigma_i = (V_i, \varphi_i)$ on the encrypted data e_i.

D. SignVerify

The FGW verifies a signature σ_i on the encrypted data e_i with ID_i, $PK_i = (A_i, B_i)$.

1) Calculate $h_i = H_1(ID_i, R_i, P_{\mathrm{pub}})$, $x_i = H_2(ID_i, V_i, PK_i, P_{\mathrm{pub}})$ and $y_i = H_3(ID_i, e_i, V_i, PK_i, P_{\mathrm{pub}})$.
2) Verifies the verification equation:
$$\varphi_i P - x_i V_i = y_i(A_i + B_i + h_i P_{\mathrm{pub}})$$
3) Accept the signature if the equation holds, otherwise reject this signature.

4) Correctness:

$$\begin{aligned}\varphi_i P &= x_i v_i P + y_i(\alpha_i P + \beta_i P)\\ &= x_i v_i P + y_i(\alpha_i P + r_i P + sh_i P)\\ &= x_i V_i + y_i(A_i + B_i + h_i P_{\text{pub}})\end{aligned}$$

E. AggregateStore

Given ciphertext signature tuple $\{ID_i, PK_i, e_i, \sigma_i\}(1 < i < n)$ on data m_i and *IPFS* node public key Λ_E, *FGW* generate certificateless aggregation signature σ and upload them to the *IPFS*.

1) Generate Ciphertext hash marks $Flag_i = SHA256(e_i)$ for each ciphertext e_i.
2) Calculate $x_i = H_2(ID_i, V_i, PK_i, P_{\text{pub}})$.

$$\tau = \sum_{i=1}^{n} x_i V_i, \phi = \sum_{i=1}^{n} \varphi_i, \Phi = \phi \Lambda_E.$$

3) Output the aggregate signature

$$\sigma = (\tau, \Phi, \{V_1, V_2, \ldots, V_n\})$$

4) Upload certificateless aggregation signature σ and ciphertext set $\{e_1, e_2, \ldots, e_n\}$ to the *IPFS*.

F. AggregateVerify

Given an aggregate signature σ on e_i, *IPFS* node performs the following operations to verify it.

1) Calculate $h_i = H_1(ID_i, R_i, P_{\text{pub}})$ and $y_i = H_2(ID_i, e_i, V_i, PK_i, P_{\text{pub}})$.
2) Verifies the verification equation:

$$\Phi - \tau = \lambda_E \sum_{i=1}^{n} (y_i(A_i + B_i + h_i P_{\text{pub}}))$$

3) Accept the signature if the equation holds, otherwise reject this signature and ciphertext.
4) Correctness:

$$\begin{aligned}\Phi - \tau &= \phi \Lambda_E - \sum_{i=1}^{n} x_i V_i\\ &= \lambda_E \phi P - \sum_{i=1}^{n} x_i V_i\\ &= \lambda_E \sum_{i=1}^{n} \varphi_i P - \sum_{i=1}^{n} x_i V_i\\ &= \lambda_E \sum_{i=1}^{n} (x_i V_i + y_i(A_i + B_i + h_i P_{\text{pub}}))\\ &\quad - \sum_{i=1}^{n} x_i V_i\\ &= \lambda_E \sum_{i=1}^{n} (y_i(A_i + B_i + h_i P_{\text{pub}}))\end{aligned}$$

5) Return the corresponding ciphertext e_i storage address Δ_i to *FGW*.

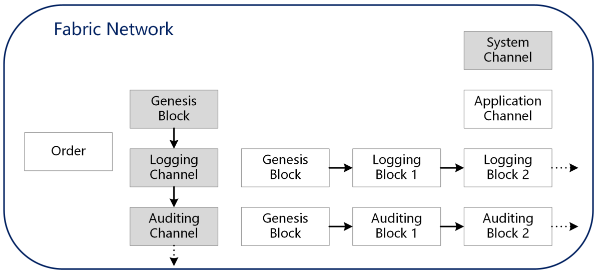


FIGURE 4. Fabric network channel.

G. StoreTransGen

After the aggregated signature is verified, the *FGW* generates data storage transactions $TX_{data} = \{ID_{FGW}, F_E, \sigma, \Delta_i\}$ and tag storage transactions $TX_{tag} = \{ID_{FGW}, F_E, Flag_i\}$ by using its public key F_E. After that, *FGW* uses the private key to generate transaction signatures σ_{data} and σ_{flag}, and then the data storage transactions and tag storage transactions are published to the logging channel and the auditing channel respectively after the signature verification is passed. The channel to isolate the storage and audit operations is shown in Fig 4.

H. AuditTransGen

When the user needs to verify the integrity of the data in the cloud, the *TPA* will obtain the hash value $HASH_t$ of the latest block on the blockchain at the current time t to construct a pseudo-random number generator $\Upsilon = Rand(HASH_t)$ and randomly select the ciphertext message $\{i, e_i\}_{i \in \Upsilon}$ for the challenge. Then *TPA* requests the identification $Tag_i = SHA256(e_i)$ of the corresponding cipher block from the IPFS node, and the IPFS node calculates $Tag = \sum Tag_i$ and responds the TPA with the proof $\{\{Tag_i\}, Tag, Sig_{\lambda_E} Tag\}$, *TPA* verifies the proof and constructs a Merkle hash tree to obtain $Root_{ipfs}$ using Tag_i, the process is shown in Fig 5. Finally the *TPA* generates the data audit transaction $TX_{audit} = \{ID_{TPA}, T_E, Root_{ipfs}, \{i\}_{i \in \Upsilon}\}$ and uses the private key to generate the transaction signature σ_{audit} and publishes it to the logging channel.

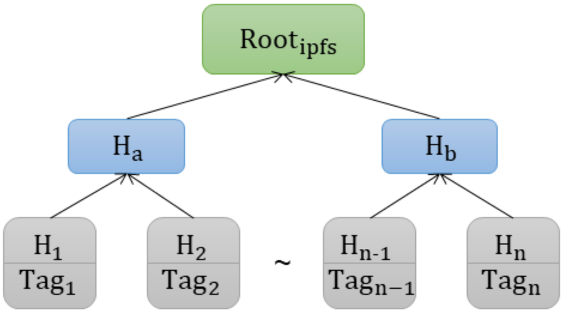


FIGURE 5. Construct $Root_{ipfs}$.

I. ReAudit

After the *TPA* submits a data audit transaction, the chain code deployed on the audit channel will look up the corresponding

TABLE 2. Description of curve parameters.

Curve	Pairing	Group	p_1, p	Length of elements of the group
$\hat{E} : y^2 = x^3 + x \ mod \ p_1$	$\hat{e} : G_1 \times G_1 \rightarrow G_T$	$G_1(P)$	p_1 =512 bits	$\|G_1\|$ =1024 bit
$E : y^2 = x^3 + ax + b \ mod \ p$	Without Pairing	$G(P)$	p =160 bits	$\|G\|$ =320 bit

TABLE 3. Description and run time of operation.

Cryptographic operation	Running time (ms)
Biliner pairing T_{bp}	4.16
Biliner pairing scalar multiplication T_{bpsm}	1.56
Biliner pairing point addition T_{bppa}	0.007
ECC scalar multiplication T_{ecsm}	0.47
ECC point addition T_{ecpa}	0.001
Map to G hash T_{htg}	3.66
Map to Z_q^* hash T_{htz}	0.001

ciphertext tag $Flag_i$ in the audit channel based on the audit data ID $\{i\}_{i \in \Upsilon}$ in the data audit transaction and use the same method as TPA to construct a Merkle Hash tree to obtain $Root_bc$.

$$Root_{ipfs} = Root_{bc}$$

Finally, the final audit results are obtained by comparing the consistency of the audit results from IPFS and the chain.

VI. SECURITY ANALYSIS

A. PROVABLE SECURITY

Provable security in Appendix A.

B. MESSAGE AUTHENTICATION & AVAILABILITY

We use the pairing-free certificateless aggregation signature algorithm for message authentication, whose security comprises two components: the unforgeability of certificateless signatures and the unforgeability of aggregated signatures. In *Section VI* on provable security, we proved the unforgeability of the certificateless signature of our scheme under the random prediction machine model, and that the trusted home smart gateway node in our scheme handles the generation and verification of the certificateless aggregation signature. Therefore, our message authentication is secure and available.

C. LIGHTWEIGHT

The purpose of our scheme is to reduce the computing and storage burden of low-power devices in the environment of the Smart Home Internet of things. Therefore, we make the family smart gateway as an edge node to assist device computing in the stage of identity authentication and message authentication. Because the family smart gateway and the device are in the same local LAN, we assume that the communication between them is safe. In the process of message authentication, the device only needs to do one hash calculation and one scalar multiplication calculation on the elliptic curve. The family smart gateway executes most of the operations.

D. ANONYMITY

In the key generation phase, the true identity of the device has been obfuscated by the hash function, i.e., $ID_i = H_0(\alpha_i P_{pub}, T_i) \oplus RID_i$, and the signature aggregation makes it difficult for an attacker to determine the identity of the user by the connection between multiple transactions. Therefore, other nodes and adversaries cannot determine the true identity of a device by analyzing the messages sent by the same device.

VII. PERFORMANCE EVALUATION

In this section, we compare the certificate-free aggregated signature scheme proposed in this paper with the certificate-free aggregated signatures currently applied in various domains of IoT. The computational overhead and communication overhead in our scheme are analyzed by calculating the master operation time and signature size used in different schemes to evaluate the performance of our proposed signature scheme. It is worth mentioning that the main goal of our scheme is to reduce the overall cost on the user's device side, including the time to verify a single signature on the device, as well as the individual signature size. At the same time, our scheme has good aggregation overhead in a real environment with multiple devices. In evaluating the bilinear pairing-based scheme for certificateless aggregation signatures, the curves are chosen to be super-singular elliptic curves $\hat{E} : y^2 = x^3 + x \ mod \ p_1$ on a finite field F_{q_1} with q_1 of 512-bit prime numbers, capable of achieving the security level of the 1024-bit RSA algorithm. To achieve the same security level, the scheme without bilinear pairs chooses the elliptic curve group G on the Koblitz elliptic curve $E : y^2 = x^3 + ax + b \ mod \ p$ over the finite field $\mathbb{Z}_q^*$ with q bit 160 bits of prime, and the relevant curve parameters are given in the Table 2. Based on the above choice of security parameters, we use the C++ based MIRACL cryptographic library to calculate the running time of different operations in the Visual Studio 2022 environment. The relevant computing hardware platform is a personal computer configured with an AMD R5-4600U processor, 16GB of RAM, and Windows 11 64bit

TABLE 4. Computational cost comparison of different certificateless aggregation signature scheme.

Scheme	Single Sign.Cost	Single Verify.Cost	Aggregate.Cost	Aggregate Verify.Cost	Pairing
Liu *et al.* [31]	$2T_{ecsm} + 3T_{htz}$	$4T_{ecsm} + 3T_{htz} + 2T_{ecpa}$	$nT_{htz} + (2n - 2)T_{ecpa}$ $+nT_{ecsm}$	$3nT_{htz} + 3nT_{ecsm} + 3nT_{ecpa}$	No
Kumar *et al.* [32]	$2T_{bpsm} + T_{htz} + T_{htg}$	$3T_{bp} + T_{bpsm}$	$(n - 1)T_{bppa}$	$(n + 1)T_{htg} + nT_{htz} + 3T_{bp} + T_{bpsm}$ $+(n - 1)T_{bppa}$	Yes
Shen *et al.* [33]	$T_{bpsm} + T_{bppa} + T_{htg}$	$2T_{htg} + 3T_{bp}$	$nT_{bp} + nT_{htz} + T_{bpsm}$ $+(n - 1)T_{bppa}$	$nT_{bp} + T_{htz} + (n - 1)T_{bppa} + T_{bpsm}$	Yes
Gayathri *et al.* [34]	$2T_{ecsm} + 3T_{htz}$	$2T_{htz} + 5T_{ecsm} + 2_{ecpa}$	$2nT_{ecsm} + (2n - 2)T_{ecpa}$	$2nT_{htz} + (2n + 1)T_{ecsm} + (2n + 1)T_{ecpa}$	No
Our Scheme	$T_{ecsm} + 2T_{htz}$	$3T_{htz} + 4T_{ecsm} + 2T_{ecpa}$	$(n - 1)T_{ecpa} + nT_{ecsm}$ $+nT_{htz}$	$2nT_{htz} + (2n + 1)T_{ecsm} + (n + 1)T_{ecpa}$	No

operating system. We indicate the runtime of the relevant operations in the Table 3. We ignored the $10^{-3}ms$ level of time overhead in our performance evaluation because their impact on the results is negligible. The comparison schemes are Liu et al. [31], Kumar et al. [32], Shen et al. [33], and Gayathri et al. [34]'s certificateless aggregation signature scheme.

A. COMPUTATION COST

We can evaluate the computational overhead of the schemes by calculating the running time of different operations in four phases: 1) the message signature generation phase, 2) the message signature verification phase, 3) the signature aggregation phase, and 4) the aggregated signature verification phase. As the Table 4 shows the computational overhead of each phase of the different schemes, we have chosen two bilinear pairing-based schemes and two elliptic curve-based schemes for comparison. We can see that the individual signature overhead of our scheme is lower than all the schemes compared, since our goal is to reduce the computational burden of smart home wireless sensor devices as much as possible. In the signature phase, the device requires only 1 scalar multiplication operation on the elliptic curve and 1 hash operation mapping to Z_q^*. In the signature verification phase, 4 scalar multiplication operations on the elliptic curve and 2 point addition operations on the elliptic curve and 3 hash operations mapping to Z_q^* are required. For multiple signatures of n devices, the aggregation phase requires $(n-1)$ elliptic curve scalar multiplication operations and n elliptic curve scalar multiplication operations and n hash operations mapped to Z_q^*. Finally, in the mapping phase of aggregated signatures, $(2n + 1)$ elliptic curve scalar multiplication operations and $(n - 1)$ elliptic curve point addition operations and $2n$ hash operations mapped to Z_q^* are required. The comparison plots of computational overhead for single signature and verification and the comparison plots of aggregated signature verification overhead are shown in Fig 6 and Fig 7. The evidence shows that for the certificateless aggregation signature scheme using no pairing, there is not much difference in computational overhead for each phase. Our scheme has the same overhead as Liu et al.'s scheme in the phase of single signature verification, and is slightly lower in the aggregation phase and the aggregated signature verification phase. The computational overhead of a single signature is lower, which is more conducive to smart home environment where the computing power of devices is low.

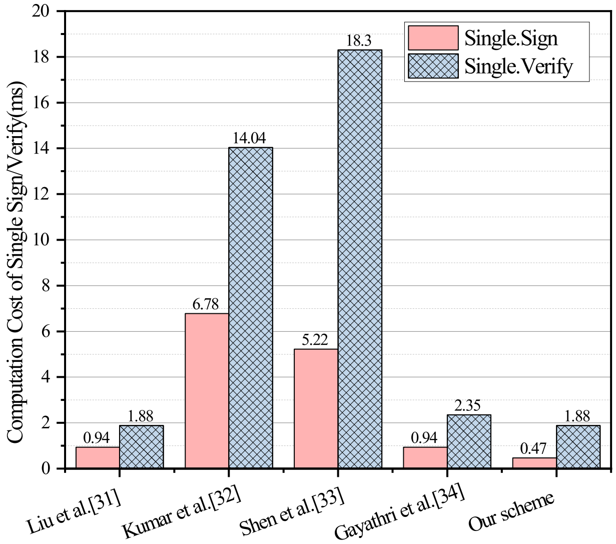


FIGURE 6. Single sign/verify computation cost.

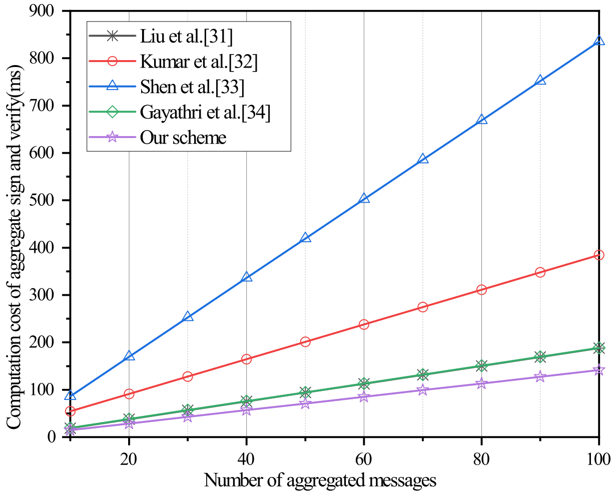


FIGURE 7. Aggregate sign/verify computation cost.

B. COMMUNICATION COST

In the signature scheme, the device sends the signature to the aggregator, and then the aggregator aggregates the

TABLE 5. Communication cost.

Scheme	Single Signature	Aggregate Signature(n message)
Shen *et al.* [33]	$\lvert 2G_1\rvert = 2048bit$	$(n+1)\lvert G_1\rvert = (n+1)1024bit$
Kumar *et al.* [32]	$\lvert 2G_1\rvert = 2048bit$	$(n+1)\lvert G_1\rvert = (n+1)1024bit$
Liu *et al.* [31]	$2\lvert G\rvert = 640bit$	$2\lvert G\rvert = 640bit$
Gayathri *et al.* [34]	$\lvert G\rvert + 2\lvert Z_q^*\rvert = 640bit$	$2\lvert G\rvert + \lvert Z_q^*\rvert = 800bit$
Our scheme	$\lvert G\rvert + \lvert Z_q^*\rvert = 480bit$	$(n+2)\lvert G\rvert = (n+2)320bit$

signature and sends it to the server. We will compare the size of individual signatures and the size of aggregated signatures in different schemes, and evaluate the communication overhead of the schemes by the signature size. The comparison of the communication overheads of different schemes is shown in the Table 5. The evidence shows that our certificateless aggregation signature scheme reduces the computational overhead of generating a single signature by the device.

According to the above experimental results, the computational and communication costs of our scheme for the single signature of wireless sensor devices are lower than those of existing schemes.

VIII. CONCLUSION

This paper proposed a lightweight smart home data privacy protection scheme based on the Consortium Blockchain. The solution includes two modules, message authentication and data integrity audit, to overcome the excessive dependence of users on third-party servers and solve the privacy protection problem of smart home device data. Meanwhile, message authentication used a pairing-free certificateless aggregate signature scheme, which significantly reduced the computing and communication overhead on the device side and ensured data integrity through the Merkle hash tree audit scheme. Moreover, the security analysis showed that the message authentication protocol proposed in this scheme meets the security requirements, and the audit results are automatically uploaded to the chain by the smart contract, which is unforgeable. Finally, the performance evaluation analysis showed that the proposed scheme significantly reduces the computing and communication overhead on the device side and is safe and feasible in the smart home environment. As part of the further research in this paper, we will expand the integrity verification scheme to support dynamic audits of that data while ensuring lightweight authentication to improve the timeliness of audit results. In addition, we will continue to study more efficient message authentication schemes to further reduce the communication overhead after device signature aggregation.

APPENDIX A
PROVABLE SECURITY

A. LEMMA 1:(NON-FAKEABILITY OF ADVERSARY $\mathcal{A}_I$)

Under the random oracle machine model, if a *Type I* adversary $\mathcal{A}_I$ can successfully forge a signature in polynomial time with a non-negligible advantage ε, then there exists an algorithm that can successfully solve the elliptic curve discrete logarithm problem in polynomial time with a non-negligible advantage $\frac{1}{e} \cdot \frac{1}{q_{PPK}} \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{\varepsilon}{q_{H_1}}$.

Proof: Suppose that Algorithm $\mathcal{C}_I$ is an instance of solving the ECDLP problem with input $(P, P_{pub} = sP)$ and the final goal is to find the value of s.

1) *Initialization phase*

$\mathcal{C}_I$ executes the *Setup* algorithm, randomly selects $s \in \mathbb{Z}_q^*$ as the system master key, calculates the corresponding system public key $P_{pub} = sP$, generates the system parameters $params = \left\{G, q, P, P_{\text{pub}}, H, H_1, H_2, H_3\right\}$ and sends them to adversary $\mathcal{A}_I$.

$\mathcal{C}_I$ maintains the initially empty lists $L_{pk} = (ID_i, \alpha_i, \beta_i, A_i, B_i)$, $L_{H1} = (ID_i, A_i, B_i, h_i)$, and L_{ID} to record the queries to the user's public key, the queries to the oracle H_1, and the adversary identity information, respectively.

$\mathcal{C}_I$ does not know $\mathcal{A}_I$'s challenge identity before the challenge, so $\mathcal{C}_I$ adaptively chooses a challenge identity $ID_{\mathcal{A}_I}$ during $\mathcal{A}_I$'s query and uses $\mathcal{A}_I$ as a subroutine to solve the ECDLP problem.

2) *Queries phase*

- $\mathcal{O}_{PK}$(*Public Key Query*) :When $\mathcal{C}_I$ receives $\mathcal{A}_I$'s *Public Key Query* about ID_i, it returns $PK_i = (A_i, B_i)$ to adversary $\mathcal{A}_I$ if the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}; otherwise $\mathcal{C}_I$ randomly selects $\alpha_i, h_i, \beta_i \in Z_q^*$, and computes:

$$A_i = \alpha_i P, B_i = \beta_i P - h_i P_{pub}.$$

  Add tuples to the lists L_{pk} and L_{H_1}, respectively, and return $PK_i = (A_i, B_i)$ to adversary $\mathcal{A}_I$.
- $\mathcal{O}_{H_1}$(H_1 *Query*) :When $\mathcal{C}_I$ receives $\mathcal{A}_I$'s H_1 *Query* about ID_i, it checks whether the corresponding tuple (ID_i, A_i, B_i, h_i) exists in the list L_{H_1}, and returns h_i to $\mathcal{A}_I$ if it exists; otherwise, $\mathcal{C}_I$ executes the $\mathcal{O}_{PK}$ and then returns the corresponding h_i to adversary $\mathcal{A}_I$.
- $\mathcal{O}_{SV}$(*Secreat Value Query*) :When $\mathcal{C}_I$ receives $\mathcal{A}_I$'s *Secreat Value Query* about ID_i, it checks whether the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}, and returns α_i to $\mathcal{A}_I$ if it exists; otherwise, $\mathcal{C}_I$ executes the $\mathcal{O}_{PK}$ and then returns the corresponding α_i to adversary $\mathcal{A}_I$.
- $\mathcal{O}_{PPK}$(*Partial Private Key Query*) :When $\mathcal{C}_I$ receives $\mathcal{A}_I$'s *Partial Private Key Query* about ID_i, if $ID_i = ID_{\mathcal{A}_I}$, then $\mathcal{C}_I$ aborts the query; if $ID_i \neq ID_{\mathcal{A}_I}$, $\mathcal{C}_I$ checks whether the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}, and return β_i to $\mathcal{A}_I$ if it exists; otherwise, $\mathcal{C}_I$ executes the $\mathcal{O}_{PK}$ and then returns the corresponding β_i to adversary $\mathcal{A}_I$.
- $\mathcal{O}_{PKR}$(*Replace Public Key Query*) :When $\mathcal{C}_I$ receives $\mathcal{A}_I$'s *Replace Public Key Query* about $PK_i^* = (A_i^*, B_i^*)$, it checks the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}, then replace A_i with A_i^* and B_i with B_i^*.

- $\mathcal{O}_{SG}$(*Signature Generation Query*) :When $\mathcal{C}_I$ receives a *Signature Generation Query* from $\mathcal{A}_I$ about ID_i, e_i and PK_i, it computes x_i, y_i and φ_i to generate a signature, and then returns the signature σ_i to $\mathcal{A}_I$ and records the identity of the adversary in the list L_{ID}.

3) *Forgery phase*

$\mathcal{A}_I$ outputs a valid forged signature σ_i' on message e', ID_i' and PK_i', and $\mathcal{C}_I$ aborts the forgery phase if $ID_i' \neq ID_{\mathcal{A}_I}$; if $ID_i' = ID_{\mathcal{A}_I}$, ID_i' is not in the adversary identity list L_{ID} and $\mathcal{A}_I$ has not queried the signature of ID_i' on e', then $\mathcal{A}_I$ generates a valid forged signature $\sigma_i' = (V_i, \varphi_i')$ of ID_i' on e'. Then, according to the Forking Lemma [30], $\mathcal{A}_I$ generates another valid forged signature $\sigma_i'' = (V_i, \varphi_i'')$ with different hash value, then we have the following equation:

$$\varphi_i' P = x_i V_i + y_i(A_i + B_i + h_i' P_{\text{pub}})$$
$$\varphi_i'' P = x_i V_i + y_i(A_i + B_i + h_i'' P_{\text{pub}})$$

According to the above equations we can calculate

$$s = \frac{(\varphi_i' - \varphi_i'')}{y_i(h_i' - h_i'')},$$

Thus $\mathcal{C}_I$ is an example of an ECDLP problem that was successfully solved using $\mathcal{A}_I$.The probability that $\mathcal{C}_I$ successfully obtains s is analyzed here. Next, we define three events.

E1: There is no interruption during the query phase.
E2: There is no interruption during the forgery phase.
E3: $\mathcal{A}_I$ successfully generates two valid forged signatures.

Assume that the probability of $ID_i \neq ID_{\mathcal{A}_I}$ is ξ, so the probability of $ID_i = ID_{\mathcal{A}_I}$ is $1-\xi$, then the probability that $\mathcal{A}_I$ is not suspended during the query phase and the forgery phase are $Pr\,[\text{E1}] \geq \xi^{q_{PPK}}$ and $Pr\,[\text{E2}] = 1-\xi$, respectively, where q_{PPK} is the query number of $\mathcal{O}_{PPK}$. According to the forking lemma, the probability that $\mathcal{A}_I$ successfully generates two valid forged signatures is $Pr\,[\text{E3}] = \left(1 - \frac{1}{e}\right)\frac{\varepsilon}{q_{H_1}}$, where q_{H_1} is the query number of $\mathcal{O}_{H_1}$.

Therefore, the probability of $\mathcal{C}_I$ successfully solving the ECDLP problem based on $\mathcal{A}_I$ is

$$Pr\,[\text{E1} \wedge \text{E2} \wedge \text{E3}] \geq \frac{1}{e} \cdot \frac{1}{q_{PPK}} \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{\varepsilon}{q_{H_1}}.$$

Hence, $\mathcal{C}_I$ can solve the ECDLP in polynomial time with a non-negligible probability.

B. LEMMA 2:(NON-FAKEABILITY OF ADVERSARY $\mathcal{A}_{II}$)

Under the random oracle machine model, if a *Type I* adversary $\mathcal{A}_{II}$ can successfully forge a signature in polynomial time with a non-negligible advantage ε, then there exists an algorithm that can successfully solve the elliptic curve discrete logarithm problem in polynomial time with a non-negligible advantage $\frac{1}{e} \cdot \frac{1}{q_{SV}} \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{\varepsilon}{q_{H_3}}$.

Proof: Suppose that Algorithm $\mathcal{C}_{II}$ is an instance of solving the ECDLP problem with input $(P, P_pub = sP)$ and the final goal is to find the value of s.

1) *Initialization phase*

$\mathcal{C}_{II}$ executes the *Setup* algorithm, randomly selects $s \in \mathbb{Z}_q^*$ as the system master key, calculates the corresponding system public key $P_{pub} = sP$, generates the system parameters $params = \{G, q, P, P_{\text{pub}}, H, H_1, H_2, H_3\}$ and sends them to adversary $\mathcal{A}_{II}$.

$\mathcal{C}_{II}$ maintains the initially empty lists $L_{pk} = (ID_i, \alpha_i, \beta_i, A_i, B_i)$, $L_{H3} = (ID_i, m_i, V_i, PK_i, P_{\text{pub}}, y_i)$, and L_{ID} to record the queries to the user's public key, the queries to the oracle H_3, and the adversary identity information, respectively.

$\mathcal{C}_{II}$ does not know $\mathcal{A}_{II}$'s challenge identity before the challenge, so $\mathcal{C}_{II}$ adaptively chooses a challenge identity $ID_{\mathcal{A}_{II}}$ during $\mathcal{A}_{II}$'s query and uses $\mathcal{A}_{II}$ as a subroutine to solve the ECDLP problem.

2) *Queries phase*

- $\mathcal{O}_{PK}$(*Public Key Query*) :When $\mathcal{C}_{II}$ receives $\mathcal{A}_{II}$'s *Public Key Query* about ID_i, it returns $PK_i = (A_i, B_i)$ to adversary $\mathcal{A}_I$ if the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}; otherwise, if $ID_i \neq ID_{\mathcal{A}_{II}}$, $\mathcal{C}_{II}$ randomly selects $x_i, r_i \in \mathbb{Z}_q^*$, calculates $A_i = x_i P, B_i = r_i P$ and $\beta_i = r_i + s \cdot H_1(ID_i, A_i, B_i)$, then the $PK_i = (A_i, B_i)$ is returned to the adversary $\mathcal{A}_{II}$ after adding the tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ to the list L_{pk}.if $ID_i = ID_{\mathcal{A}_{II}}, \mathcal{C}_{II}$ randomly selects $r_i, A_i \in \mathbb{Z}_q^*$, calculates $B_i = r_i P$, then the $PK_i = (A_i, B_i)$ is returned to the adversary $\mathcal{A}_{II}$ after adding the tuple $(ID_i, \perp, \perp, A_i, B_i)$ to the list L_{pk}.
- $\mathcal{O}_{H_3}(H_3\ Query)$:When $\mathcal{C}_{II}$ receives $\mathcal{A}_{II}$'s H_3 *Query* about ID_i, it checks whether the corresponding tuple $(ID_i, e_i, V_i, PK_i, P_{\text{pub}}, y_i)$ exists in the list L_{H_3}, and returns y_i to $\mathcal{A}_{II}$ if it exists; otherwise, $\mathcal{C}_{II}$ randomly selects $y_i \in \mathbb{Z}_q^*$ and then returns the corresponding y_i to adversary $\mathcal{A}_{II}$ after adding the y_i to the list L_{H_3}.
- $\mathcal{O}_{SV}$(*Secreat Value Query*) :When $\mathcal{C}_{II}$ receives $\mathcal{A}_{II}$'s *Secreat Value Query* about ID_i, if $ID_i = ID_{\mathcal{A}_{II}}, \mathcal{C}_{II}$ aborts the query; otherwise, $\mathcal{C}_{II}$ checks whether the corresponding tuple $(ID_i, \alpha_i, \beta_i, A_i, B_i)$ exists in the list L_{pk}, if it exists then return α_i to $\mathcal{A}_{II}$, if not then executes the $\mathcal{O}_{PK}$ and return α_i to $\mathcal{A}_{II}$.
- $\mathcal{O}_{SG}$(*Signature Generation Query*) :When $\mathcal{C}_{II}$ receives a *Signature Generation Query* from $\mathcal{A}_{II}$ about ID_i, e_i and PK_i, it computes x_i, y_i and φ_i to generate a signature, and then returns the signature σ_i to $\mathcal{A}_{II}$ and records the identity of the adversary in the list L_{ID}.

3) *Forgery phase*

$\mathcal{A}_{II}$ outputs a valid forged signature σ_i' on message e', ID_i' and PK_i', and C aborts the forgery phase if $ID_i' \neq ID_{\mathcal{A}_{II}}$; if $ID_i' = ID_{\mathcal{A}_{II}}$, ID_i' is not in the adversary identity list L_{ID} and $\mathcal{A}_{II}$ has not queried the signature of ID_i' on e', then $\mathcal{A}_{II}$ generates a valid forged signature $\sigma_i' = (V_i, \varphi_i')$ of ID_i' on m'. Then, according to the Forking Lemma, $\mathcal{A}_{II}$ generates another valid forged signature $\sigma_i'' = (V_i, \varphi_i'')$ with different hash value, then we have the following equation:

$$\varphi_i'P = x_iV_i + y_i'(A_i + B_i + h_iP_{\text{pub}})$$
$$\varphi_i''P = x_iV_i + y_i''(A_i + B_i + h_iP_{\text{pub}})$$

According to the above equations we can calculate

$$s = \frac{\varphi_i' - \varphi_i''}{h_i(y_i' - y_i'')} - \frac{x_i + r_i}{h_i},$$

Thus $\mathcal{C}_{II}$ is an example of an ECDLP problem that was successfully solved using $\mathcal{A}_{II}$.The probability that $\mathcal{C}_{II}$ successfully obtains s is analyzed here. First, we define the following events.

E1: There is no interruption during the query phase.
E2: There is no interruption during the forgery phase.
E3: $\mathcal{A}_{II}$ successfully generates two valid forged signatures.

Assume that the probability of $ID_i \neq ID_{\mathcal{A}_{II}}$ is ξ, so the probability of $ID_i = ID_{\mathcal{A}_{II}}$ is $1-\xi$, then the probability that $\mathcal{A}_{II}$ is not suspended during the query phase and the forgery phase are $Pr\,[\text{E1}] \geq \xi^{q_{SV}}$ and $Pr\,[\text{E2}] = 1 - \xi$, respectively, where q_{SV} is the query number of $\mathcal{O}_{SV}$. According to the forking lemma, the probability that $\mathcal{A}_I$ successfully generates two valid forged signatures is $Pr\,[\text{E3}] = \left(1 - \frac{1}{e}\right)\frac{\varepsilon}{q_{H_3}}$, where q_{H_3} is the query number of $\mathcal{O}_{H_3}$.
Therefore, the probability of $\mathcal{C}_{II}$ successfully solving the ECDLP problem based on $\mathcal{A}_{II}$ is

$$Pr\,[\text{E1} \wedge \text{E2} \wedge \text{E3}] \geq \frac{1}{e} \cdot \frac{1}{q_{SV}} \cdot \left(1 - \frac{1}{e}\right) \cdot \frac{\varepsilon}{q_{H_3}}.$$

Hence, $\mathcal{C}_{II}$ can solve the ECDLP in polynomial time with a non-negligible probability.

REFERENCES

[1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020.

[2] A. H. Sodhro, A. Gurtov, N. Zahid, S. Pirbhulal, L. Wang, M. M. U. Rahman, M. A. Imran, and Q. H. Abbasi, "Toward convergence of AI and IoT for energy-efficient communication in smart homes," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9664–9671, Jun. 2021.

[3] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang, and F. Zhou, "Access control and authorization in smart homes: A survey," *Tsinghua Sci. Technol.*, vol. 26, no. 6, pp. 906–917, Dec. 2021.

[4] Y. Li, Z. Zhang, X. Wang, E. Lu, D. Zhang, and L. Zhang, "A secure sign-on protocol for smart homes over named data networking, *IEEE Commun. Mag.*, vol. 57, no. 7, pp. 62–68, Feb. 2019.

[5] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale Internet of Things data storage and protection," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 762–771, Sep. 2019.

[6] Y. Lin, J. Li, S. Kimura, Y. Yang, Y. Ji, and Y. Cao, "Consortium blockchain-based public integrity verification in cloud storage for IoT," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3978–3987, Mar. 2022.

[7] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[8] M. AbuNaser and A. A. A. Alkhatib, "Advanced survey of blockchain for the Internet of Things smart home," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 58–62.

[9] C. Xie, J. Weng, and D. Zhou, "Revocable identity-based fully homomorphic signature scheme with signing key exposure resistance," *Inf. Sci.*, vol. 594, pp. 249–263, May 2022.

[10] J. Li, J. Wu, G. Jiang, and T. Srikanthan, "Blockchain-based public auditing for big data in cloud storage," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102382.

[11] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Comput. Secur.*, vol. 86, pp. 132–146, Sep. 2019.

[12] F. Mezrag, S. Bitam, and A. Mellouk, "An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 200, Apr. 2022, Art. no. 103282.

[13] J. Pirayesh, A. Giaretta, M. Conti, and P. Keshavarzi, "A PLS-HECC-based device authentication and key agreement scheme for smart home networks," *Comput. Netw.*, vol. 216, Oct. 2022, Art. no. 109077.

[14] G. S. Poh, P. Gope, and J. Ning, "PrivHome: Privacy-preserving authenticated communication in smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1095–1107, May 2021.

[15] A. Xiang and J. Zheng, "A situation-aware scheme for efficient device authentication in smart grid-enabled home area networks," *Electronics*, vol. 9, no. 6, p. 989, Jun. 2020.

[16] J. Oh, S. Yu, J. Lee, S. Son, M. Kim, and Y. Park, "A secure and lightweight authentication protocol for IoT-based smart homes," *Sensors*, vol. 21, no. 4, p. 1488, Feb. 2021.

[17] 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, p. 1215, Feb. 2020.

[18] F. Al-Turjman and B. D. Deebak, "Seamless authentication: For IoT-big data technologies in smart industrial application systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2919–2927, Apr. 2021.

[19] D. Kaur and D. Kumar, "Cryptanalysis and improvement of a two-factoruser authentication scheme for smart home," *J. Inf. Secur. Appl.*, vol. 58, pp. 102787–102798, Mar. 2021.

[20] S. Yu, N. Jho, and Y. Park, "Lightweight three-factor-based privacy-preserving authentication scheme for IoT-enabled smart homes," *IEEE Access*, vol. 9, pp. 126186–126197, 2021.

[21] K. Nimmy, S. Sankaran, K. Achuthan, and P. Calyam, "Lightweight and privacy-preserving remote user authentication for smart homes," *IEEE Access*, vol. 10, pp. 176–190, 2022.

[22] S. G. Liu, R. Liu, and S. Y. Rao, "Secure and efficient two-party collaborative SM9 signature scheme suitable for smart home," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4022–4030, Jul. 2022.

[23] J. Kar, K. Naik, and T. Abdelkader, "A secure and lightweight protocol for message authentication in wireless sensor networks," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3808–3819, Sep. 2021.

[24] J. Kar, X. Liu, and F. Li, "CL-ASS: An efficient and low-cost certificateless aggregate signature scheme for wireless sensor networks," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102905.

[25] L. Zhou and X. Yin, "An improved pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *PLoS ONE*, vol. 17, no. 7, Jul. 2022, Art. no. e0268484.

[26] Y. Ren, Y. Leng, J. Qi, P. K. Sharma, J. Wang, Z. Almakhadmeh, and A. Tolba, "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Gener. Comput. Syst.*, vol. 115, pp. 304–313, Feb. 2021.

[27] H. Li, D. Han, and C. Chang, "DAC4SH: A novel data access control scheme for smart home using smart contracts," *IEEE Sensors J.*, vol. 23, no. 6, pp. 6178–6191, Mar. 2023.

[28] S. S. Al Riyami and K. G. Patersony, "Certificateless public key cryptography," in *Proc. 9th Int. Conf. Theory Appl. Cryptol.*, 2003, pp. 452–473.

[29] G. Wu, F. Zhang, L. Shen, F. Guo, and W. Susilo, "Certificateless aggregate signature scheme secure against fully chosen-key attacks," *Inf. Sci.*, vol. 514, pp. 288–301, Apr. 2020.

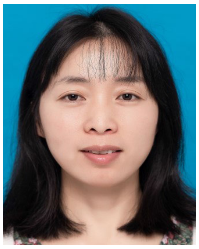
[30] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, Jun. 2000.

[31] J. Liu, L. Wang, and Y. Yu, "Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5256–5266, Jun. 2020.

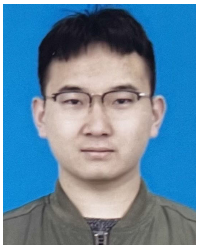
[32] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustain. Comput., Informat. Syst.*, vol. 18, pp. 80–89, Jun. 2018.

[33] L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao, "A secure and efficient ID-based aggregate signature scheme for wireless sensor networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 546–554, Apr. 2017.

[34] N. B. Gayathri, G. Thumbur, P. R. Kumar, M. Z. U. Rahman, P. V. Reddy, and A. Lay-Ekuakille, "Efficient and secure pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9064–9075, Oct. 2019.



BAI LIU received the Ph.D. degree from the University of Chinese Academy of Sciences, in 2016. She is currently an Associate Professor with the School of Computers, Hubei University of Technology, Wuhan. Her research interests include the technology of privacy preserving, information security, secure cloud computing, and quantum security.



XUEYAN YAO received the B.S. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China. He is currently pursuing the master's degree with the College of Computer Science, Hubei University of Technology, Wuhan, China. His research interest includes the IoT data privacy protection.

KUIKUI GUO received the B.S. degree from the Zhengzhou Shengda School of Economics and Management, Zhengzhou, China. He is currently pursuing the master's degree with the College of Computer Science, Hubei University of Technology, Wuhan, China. His research interest includes quantum secure multiparty computing.

PENGDA ZHU received the B.S. degree from Xinxiang University, Xinxiang, China. He is currently pursuing the master's degree with the College of Computer Science, Hubei University of Technology, Wuhan, China. His research interest includes quantum digital signature.