2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

EPASC-SH: Efficient Privacy-Preserving Authentication and Secure Communication Protocol for Smart Homes

Shuo Wang[1], Yifan Liu[1,2(✉)], Wenlei Chai[3], Fan Feng[3], Yi Liu[3], and Zhenpeng Liu[1]

[1] School of Cyberspace Security and Computer, Hebei University, Baoding, China

[2] School of Management, Hebei University, Baoding, China

[3] Information Technology Center, Hebei University, Baoding, China

lyf@hbu.edu,cn

Abstract. Existing authentication schemes in smart home environments often suffer from centralized reliance, high computational overhead, and vulnerability to single-point failures. Therefore, this paper proposes a distributed aggregated signature-based authentication scheme. The scheme employs an anonymous mechanism to collaboratively generate anonymous identity signatures, through multiple authorization centers to reduce the risk of single-point failure and enhance privacy protection. The signature accumulator aggregates and uniformly verifies the signatures of multiple smart devices to reduce the computational overhead and improve the authentication efficiency. Experimental results show that the scheme can effectively improve the efficiency of signature generation in smart home environments and outperforms existing authentication schemes in terms of computation, communication, and energy overhead, thus providing an identity authentication method with strong security, high authentication efficiency, and privacy protection for smart home systems.

Keywords: Multiple Authorization Center, Anonymous Authentication, Aggregated signatures, Smart Home

1 Introduction

With the advancement of Internet of Things (IoT) technology, the widespread adoption of smart homes is driving the transformation of residential living toward greater intelligence. Family members can remotely monitor and manage home devices, such as surveillance cameras, light control sensors, temperature sensors, and smart TVs, through mobile terminals like smartphones [1]. IoT technology facilitates efficient data transmission by enabling the exchange of commands between devices and users over a network connection [2]. However, the large-scale integration of smart devices imposes higher demands on security mechanisms, making the secure transmission of data a critical concern [3].

In smart home systems, different types of devices have different needs for authentication methods. Core devices, such as smart gateways need to support high concurrency authentication to avoid excessive computational burden; mobile devices, such as smartphones need to be equipped with secure, low-latency remote authentication capabilities [4] to ensure convenient control; low-power devices, such as smart bulbs should adopt lightweight authentication to reduce computational and communication overhead [5] ; temporary devices, such as visitor mobile phones should adopt dynamic authentication mechanisms to ensure that authorization can be controlled to avoid security risks. Temporary devices, such as visitor's mobile phone should adopt dynamic authentication mechanism to ensure that the authorization is controllable and avoid security risks. Despite the many conveniences brought about by the widespread use of smart devices, information exchange remains vulnerable to malicious attacks. Therefore, there is an urgent need for a secure and efficient communication protocol that not only ensures the security of data sharing and information transmission but also supports the dynamic access and authentication of large-scale IoT devices.

Traditional smart home authentication schemes typically rely on centralized servers or third-party authorities [6,7].When multiple devices simultaneously send messages and depend solely on a single authorization entity for key management and distribution, the authorization center may become overloaded, leading to single points of failure and increased vulnerability to attacks. This centralized approach is unsuitable for the dynamic nature of smart home environments. To address these challenges, distributed encryption technology based on multiple authorization centers has been introduced [8]. This distributed architecture generates cryptographic keys and provides authentication and access control in a decentralized manner, effectively mitigating the risks associated with single points of failure and centralized management. However, in scenarios where multiple devices engage in concurrent communication, the communication overhead, computational burden, and storage requirements increase significantly, leading to reduced system efficiency. To overcome these limitations, aggregate signature technology has been proposed [9]. By compressing multiple signatures into a single aggregate signature through an aggregator, this technique effectively reduces the computational and communication overhead of smart home devices [10]. Additionally, [11] identified several vulnerabilities in smart home devices, making them susceptible to various attacks, such as phishing and distributed denial-of-service (DDoS) attacks [12]. Beyond security threats, device privacy is a critical concern in smart home environments. Data transmission between mobile and smart devices often involves sensitive user information, which may lead to unauthorized access and control of smart devices [13]. To enhance user identity privacy, [14] introduced a fully aggregated encryption technique combined with a pseudonym mechanism for message transmission, effectively strengthening privacy protection in smart home systems.

Although existing authentication mechanisms and signature schemes have made significant progress in terms of security, efficiency, and privacy protection, they still have certain limitations. This paper proposes an efficient privacy-preserving authentication and communication protocol tailored for smart home environments. The proposed scheme enhances system security through a multi-authority distributed architecture and improves authentication efficiency and privacy protection by integrating an anonymous authentication mechanism and signature aggregation technology. The main contributions of this paper are as follows:

1. Distributed Architecture: The proposed scheme distributes key issuance and identity authentication across multiple authorization centers, reducing reliance on a single authority, mitigating single points of failure, and enhancing the security of smart home systems.
2. Accumulator-Assisted Authentication: By leveraging an accumulator, the scheme efficiently aggregates signatures with minimal storage and computational overhead, supports batch authentication, reduces communication costs, and improves system flexibility.
3. Anonymous Authentication Mechanism: The EPASC-SH scheme employs an anonymous generation algorithm to create pseudonymous identifiers for smart home devices in message interactions, effectively safeguarding device privacy while ensuring unlinkability.
4. Performance Evaluation: This paper conducts a comparative analysis of eight signature schemes. Experimental results demonstrate that the EPASC-SH scheme eliminates the need for bilinear pairing operations, significantly reducing computational costs and meeting the efficiency requirements of smart home authentication.

The rest of the paper is as follows: Section II presents the preparatory knowledge. Section III discusses the system design and main algorithms. Section IV performs the security analysis of the system. Section V evaluates the scheme and performs performance comparison and experimental analysis with other schemes. Section VI concludes.

2 Preliminaries

2.1 Elliptic Curve Discrete Logarithm Problem (ECDLP)

ECDLP: g is a cyclic group of prime order q and g is a generating element. For an arbitrary $P \in G$ put λ as security parameter. The ECDLP problem is a probabilistic polynomial time (PPT) where choose $r \in Zq^*$ and $P = rQ$ .For PPT Counterparty A, which solves the ECDLP, the advantages is $\Pr[A(Q, rQ) = r] \le negl(\lambda)$, the optimal where $negl(\lambda)$ is an ignorable value for the security parameter.

2.2 Threat Model

The security model of the EPASC-SH considers two types of attacks based on the attacker's capabilities:1) Public Key Replacement Attack (Type I Attacker A1): The attacker has the ability to replace the user's public key but does not possess the system's master secret key.2) Malicious AC Attack (Type II Attacker A2): The attacker holds the system's master secret key but is unable to replace the user's public key.

3 EPASC-SH System Model

3.1 System Overview

In a smart home environment, the EPASC-SH system ensures secure and efficient message transmission through distributed authorization, anonymous authentication, and signature aggregation. When a smart device initiates a service request to the authorization center, the center generates a device key and an anonymous identifier based on the device's attributes, ensuring both identity privacy and untraceability. When transmitting monitored data to users, the smart device must generate a digital signature to guarantee message integrity and security. If multiple smart home devices generate signatures simultaneously, multiple authorization centers collaboratively compute the signatures and accumulator aggregates the previous to assist the user in validation, thereby reducing computational and communication overhead. Upon successful verification, the aggregated signature is forwarded to the sensors, which execute the corresponding smart home control operations based on user commands.

This scheme enhances authentication efficiency, reduces system overhead, and effectively protects user privacy. The system architecture is illustrated as Fig. 1 and relevant symbols in Table 1.

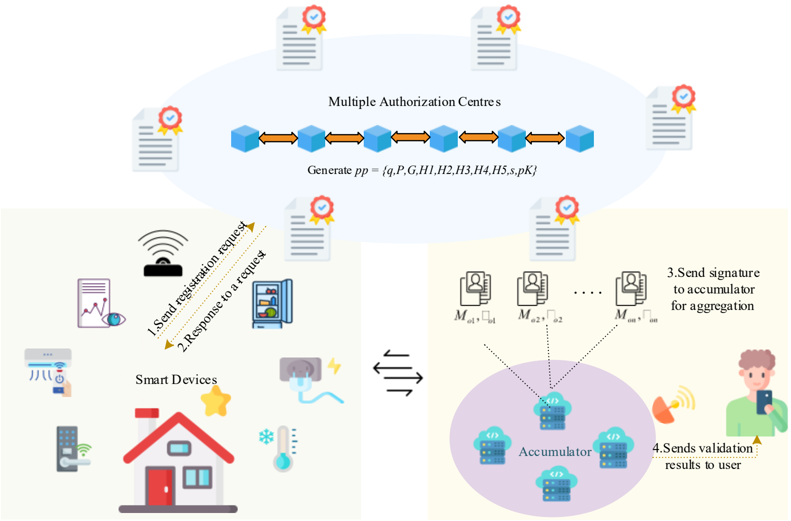

Multiple Authorization Centres
Generate pp = {q,P,G,H1,H2,H3,H4,H5,s,pK}
1.Send registration request
2.Response to a request
Smart Devices
M_{o1}, □$_{o1}$
M_{o2}, □$_{o2}$
....
M_{on}, □$_{on}$
3.Send signature to accumulator for aggregation
Accumulator
4.Sends validation results to user

Fig. 1.System model

3.2 System Detailed Design

3.2.1 System setup Phase

$Setup \rightarrow (pp, msk)$: System first generates a cyclic group of order q based on security parameters λ, with Ci selecting P as generator and generates hash q H_{1-5} functions.

System selects a polynomial of degree n-1, given by $N_i\left(x\right)=a_i0+a_{i1}x+\ldots+a_{i(t-1)}x^{n-1}$. Computes the key components N_{ik} ,where n_{ik} is the i-th coefficient of the polynomial $\left(k=0,1,\ldots,n-1\right)$, N_i is transmitted to other centers. Following this, $A_{ik}=P^{aik}\,(\bmod q)$ is computed and value $f_{ij}=N_i\left(C_j\right)$ is determined. Then broadcasts centers $C_j\,(j\neq i)$. Through the formula $P^{f_{ij}}=\sum_{k=0}^{t-1}(N_{ik})C_j^k$ verify authorization center is trust. Computes $s=\sum_{i=1}^{c}a_{i0}$ and computes $pk=sP$.The s of the system is generated collaboratively by multiple authorization centers, and each authorization center knows only part of the key. Ultimately generates public parameters params $pp=\{q,P,G,H_{1-5},s,pk\}$.

Table 1. Description of symbols

Notation	Definitions
G, GT	Two groups that support bilinear maps
pp	System public parameter
P	Generator
Ci	Authorization Center
ID	Device's identity
FID	Device's pseudonym
SKD, PKD	Device private key and public key
H1-H2	$H_{1-2}:\{0,1\}^*\times G\times G\rightarrow Zq^*$
H3-H4	$H_{3-4}:G\times G\times G\times Zq^*\rightarrow Zq^*$
H5	$H5:Zq^*\rightarrow\{0,1\}^*$
c	Number of authorization centers
s, psk	System master key and public key
Moi	Message
δ_{oi}	Digital signature

3.2.2 Registration Phase

$Registration\rightarrow(CertID_i,PKD_i,SKD_i,FID_i)$: Device (D_i) provides its identifier ID_i to Ci, which choose $z_i\in Zq^*$ and derives $W_i=ID_i\oplus H_1(z_i)$ .This pair (ID_i,W_i) is then forwarded to Ci, where select random number $k_i\in Zq^*$ and compute $K_i=k_iP$.Using these values generate $CertIDi=(K_i,B_i)$, which $B_i=k_i+sh_{1i}$ .The credential's validity is confirmed if the equation, if the equation $B_iP=K_i+h_{1i}pk$ holds . Then Ci selection $y_i\in Zq^*$ and derives the corresponding public key $Y_i=y_iP$.The private key

$SKD_i = y_i$ and public key $PKD_i = (K_i, Y_i)$,both of which are securely transmitted to the device. To enhance privacy, device creates pseudonym $\{(FID_{1i}, z_{1i}),......,(FID_{ni}, z_{ni})\}$, where $(FID_{ji}, z_{ji}) \in \{0,1\}^* \times Zq^*$ and $FID_{ji} \oplus H5(z_{ji}) = ID_i \oplus H5(z_i)$.

3.2.3 Identity authentication phase

$Authentication \to (\delta_{oi} \to (0,1))$: D_i selects pseudonym $(FID_{ji}, z_{ji})_{j \in \{1,2......n\}}$ and creates $CertID_{oi} = (K_{oi}, B_{oi})$ .To verify the validity of this certificate, it must satisfy the equation $B_{oi}P = k_{oi} + h_{1i}pk$ .Then Di selects $y_{oi} \in Zq*$ and computes $Y_{oi} = y_{oi}P$,defining the pseudonym key $PKD_{oi} = (K_{oi}, Y_{oi})$ and $SKD_{oi} = y_{oi}$ .Additionally selects $u_{oi} \in Zq*$, computes $U_{oi} = u_{oi}P$. Set messages $M_{oi} = \{FID_{ji}, z_{ji}, ID_w, U_{oi}, T_{oi}\}$, T_{oi} represents the timestamp and IDw is the identifier of the user. The computation of δ_{oi} follows equation $\delta_{oi} = h_{2i}B_{oi} + h_{3i}u_{oi} + h_{4i}y_{oi}$,where $h_{2i} = H_2(M_{oi}, PKD_{oi}, U_{oi})$, $h_{3i} = H_3(pk, PKD_{oi}, U_{oi},$ $h_{2i})$ and $h_{4i} = H_4(pk, PKD_{oi}, U_{oi}, h_{2i})$ .This computed message pair (M_{oi}, δ_{oi}) is sent to the user for authentication verification. After receives authentication request, first check T_{oi} , if valid then proceed to the following steps; otherwise output "0". By computing $h_{1i} = H(W_i, K_i, pk)$ along with h_{2i} , h_{3i} , h_{4i} .If $\delta_{oi}g = h_{2i}(K_{oi} + h_{1i}pk) + h_{3i}U_{oi} + h_{4i}Y_{oi}$ is valid, returns "1", conversely, if the authentication fails, returns "0".

3.2.4 Accumulator-assisted verification:

When multiple devices send detection results $(M_{oi}, \delta_{oi})_{i=1,...n}$ to the user at the same time and wait for the user to perform the relevant operation, we design an accumulator to assist the user in message verification. First check message's T_{oi} , If legitimate, calculate $\delta_o = \sum_{i=1}^{n} \delta_{oi}$ and correlation $W_i = FID_{ji} \oplus H5(z_{ji})$ .Setting the corresponding identifier W_o and public key set U_o ,then send message $(\delta_o, ID_w, W_o, U_o, T_{SA})$ to accumulator. The accumulator first checks T_{SA} after receiving an aggregation message, if the timestamp is expired, it will be discarded, otherwise, calculate the equation $\delta_o g = \sum_{i=1}^{n} h_{2i}K_{oi} + (\sum_{i=1}^{n} h_{2i}h_{1i})pk + \sum_{i=1}^{n} h_{3i}U_{oi} + \sum_{i=1}^{n} h_{4i}Y_{oi}$ .If the equation holds it means that the aggregated information is validly sent to the user, and the user performs the relevant operation according to the detection situation.

3.2.5 Traceability and revocation:

When a device is found to have committed a malicious act (forging an identity, sending a malicious message, etc.), the authorization center needs to trace the real identity .If the malicious device's authentication request is $(M_{oi}, \delta_{oi}, U_{oi}, T_{oi})$, where $M_{oi} = \{FID_{ji}, z_{ji}, ID_w, U_{oi}, T_{oi}\}$. C_i is necessary to verify whether the signature of the malicious device is valid to prevent the attacker from forging the identity to trace the innocent device. Firstly, according to the equation $\delta_{oi} = h_{2i}B_{oi} + h_{3i}u_{oi} + h_{4i}y_{oi}$ verify whether the certificate of the device is valid, if the equation is valid then the authentication request is from the malicious device. Trace the real identity ID_i of the malicious device according to $ID_i = W_i \oplus H_5(z_i)$,where $W_i = FID_{ji} \oplus H5(z_{ji})$, and recover its , meanwhile find the corresponding $CertID_{oi} = (K_{oi}, B_{oi})$. Add $CertID_{oi}$ to the revocation list CRL , $CRL \leftarrow CRL \cup \{CertID_{Oi}\}$ .After the device certificate is revoked, the device certificate can not continue the authentication request to ensure its security.

4 Security Proof

To ensure that the EPASC-CH scheme is resistant to forgery attacks, we present a security proof demonstrating its ability to withstand A1 and A2 attacks under the given threat model. The proof establishes its existential unforgeability under chosen-message attacks (EUF-CMA) using Theorems 1 and 2.

Theorem 1: If the adversary A1 can successfully forge a signature with non-negligible ability $\varepsilon_1{}' \geq (1-[1/e])([\varepsilon_1]/[e(q1+q2+1)\ q_{H3}])$, then the challenger C1 solves the ECDLP problem.

Proof: Given (P, sP) as input to ECDLP. The interaction between C1 and adversary A1 proceeds as follows:

Initialization Phase: Challenger C1 initializes $pk = sP \in G$, where s is unknown to C1. C1 generates parameters q, P, G, H_{1-4}, pk and maintains four random oracle lists LH_{1-4} corresponding to different hash queries. Additionally, Lu is used to record user-related information obtained during the device creation.

Query: A1 the implementation details are as follows:

1) Device Creation: C1 maintains an initially empty list. When A1 queries with (FID_{ji}, z_{ji}) ,C1 computes $FID_{ji} \oplus H5(z_{ji}) = ID_i \oplus H5(z_i)$ .If $FIDij = FID^*$,C1 randomly selects $k_{oi}{}^*, y_{oi}{}^*, h_{1i}{}^* \in Zq^*$, computes $K_{oi}{}^* = k_{oi}{}^*P$, $Y_{oi} = y_{oi}{}^*P$ .Add the tuple $(FID^*, \perp, y_{oi}{}^*, Y_{oi}{}^*, K_{oi}{}^*)$ and $(FID*, K_{oi}*, pk, h_{1i}*)$ to Lu and LH1. C1 then outputs $(FID_{ij}, y_{oi}, B_{oi}, Y_{oi}, K_{oi})$. If not, C1 selects random values $k_{oi}{}^*, y_{oi}{}^*, h_{1i}{}^* \in Zq^*$, computes $K_{oi} = B_{oi}P - h_{1i}pk$ and $Y_{oi} = y_{oi}{}^*P$ then outputs $PKD_{oi} = (K_{oi}, Y_{oi})$ to A1, storing the tuple $(FID_{ij}, y_{oi}, B_{oi}, Y_{oi}, K_{oi})$ and $(FID_{ij}, K_{oi}, pk, h_{1i})$ in Lu and LH1.

2) H-Query: When receiving A1's $H_{1\sim4}$ query, if $(FID_{ij}, K_{oi}, pk, h_{1i}) \in L_{H1}$, returns h_{1i} ,if $(M_{oi}, PKD_{oi}, U_{oi}, h_{2i}) \in L_{H2}$,returns h_{2i} ; if $(pk, PKD_{oi}, U_{oi}, h_{2i}, h_{3i}) \in L_{H3}$, returns h_{3i} ; if $(pk, PKD_{oi}, U_{oi}, h_{2i}, h_{4i}) \in L_{H4}$,returns h_{4i} .Otherwise, C1 randomly selects $h_{2i}, h_{3i}, h_{4i} \in Zq^*$ and adds the relevant information to the list L_{H1-4} .

3) Certificate Creation: When A1 submits FID_{ji} for a certificate signing, C1 checks if $(FID_{ij}, y_i, B_i, Y_i, K_i) \in L_u$, C1 returns $CertID_{oi} = (K_{oi}, B_{oi})$ to A1.Otherwise, C1 runs Device Creation procedure send $CertID_{oi} = (K_{oi}, B_{oi})$ to A1 and then terminates.

4) PrivateKey: When $FID_{ji} \oplus H5(z_{ji}) = ID_i \oplus H5(z_i)$ with A1 input (FID_{ji}, z_{ji}). If $FID_{ij} = FID^*$, C1 continues and returns $\perp$. If $(FID_{ij}, y_i, C_i, Y_i, K_i) \in L_u$, then C1 returns $SKD_{oi} = y_i$.Otherwise, C1 runs Create Device, returns $SKD_{oi} = yi$.C1 produces $\perp$, if A1 takes the place of the matching public key.

5) Replace PublicKey: Public key query request from A1, where $PKD'_{oi} = (K'_{oi}, Y'_{oi})$, Computes $FID_{ji} \oplus H5(z_{ji}) = ID_i \oplus H5(z_i)$.If $FID_{ij} = FID*$, C1 ignored. If not, C1 modifies tuple $(FID_{ij}, y_i, B_i, Y_i', K_i')$.

6) Sign: Signature query with a (FID_{ij}, M_{oi}) from A1.If $FID_{ij} = FID^*$ and $y_i \neq \perp$, C1 selects $u_{oi} \in Zq^*$ and computes $U_{oi} = u_{oi}P$.Then Ci calculates $\delta_{oi} = h_{2i}(B_{oi} + u_{oi}) + h_{3i}y_i$,where $h_{3i} = H_3(pk, PKD_{oi}, U_{oi}, h_{2i})$.Lastly,C1 returns the signature pair (U_{oi}, δ_{oi}). If $FID_{ij} \neq FID^*$,C1 selects random values $\delta_{oi}{}^*, h_{2i}{}^*, h_{3i}{}^*, h_{4i}{}^* \in Zq^*$ to computes $U_{oi}{}^* = (1/h_{3i}{}^*)(\delta_{oi}{}^*P - h_{4i}{}^*Y_i^* - h_{2i}{}^*(K_{oi}{}^* + h_{1i}{}^*pk))$ and returned signature (U_{oi}, δ_{oi}).Notice that, h_{1i} or $h_{1i}{}^*$ can be retrieved from LH1. During this process, (FID_{ij}, M_{oi}) is inserted to LS, which is initialized to store all signing search information.

Forgery Stage: Once all queries are completed, A1 outputs a forged signature (δ^*, U^*) for pair $(FID_{ij}{}^*, M^*)$,where $U^* = u^*P$. If $FID_{ij} \neq FID*$, C1 outputs ⊥; otherwise C1 replays A1 and obtains forged signature $(\delta_1{}^*, U_1{}^*)$.

Regarding Forking Lemma [15], A1 can provide legitimate signature $(\delta_1{}^*, U_1{}^*)$, if C1 repeats A1 with the exact same integer $u^* \in Zq^*$ but a different answer h_2^1.Now, we can have formal. Then C1 output $\mathrm{a}=(1/h_1^*)([(\delta^* - \delta_1^*)/(h_2^* - h_{_2}^1)] - k^*)$ as a remedy for the ECDLP issue.

$$\begin{cases} \delta^* = h_2^*(k^* + ah_1^*) + h_3^*u^* + h_4^*\mathrm{y}^* \\ \delta_1^* = h_{_2}^1(k^* + ah_1^*) + h_3^*u^* + h_4^*\mathrm{y}^* \end{cases} \tag{1}$$

In order to analyze the advantages of C1 successfully outputting this solution ε_1, the following events are defined.

(1) The probability that M1 occurs, meaning C1 fails in the querying phase, is at least: $Pr(M1) \geq (1 - 1/[q1 + q2 + 1])^{q1+q2}$,where q1 and q2 are the amount of queries made for certificate generation and private key generation

(2) The probability that M2 happens, meaning A1 successfully generates two distinct valid signatures is: $Pr(M1) \geq (1 - 1/[q1 + q2 + 1])^{q1+q2}$.

(3) If A1 has a probability ε_1 of producing two valid signatures $\delta_{i1}{}^*$ and $\delta_{i2}{}^*$ with a probability of at least:, $Pr(M3) \geq (1 - 1/e)(\varepsilon 1/qH_3)$.

Thus, the combined probability of all three events occurring is:

$$Pr(M1 \wedge M2 \wedge M3) \geq (1 - \frac{1}{e}) \frac{\varepsilon_1}{e(q1 + q2 + 1)q_{H3}} \quad (2).$$

Theorem 2: If the adversary A2 succeeds in forging the forged signature with non-negligible ability $\varepsilon_2{}' \geq (1-[1/e])([\varepsilon_2]/[e(q2+1)\ q_{H3}^{1}])$, then the challenger C2 solves the ECDLP problem.

Proof: First, tuple (P, sP) as input to ECDLP. The interaction between C2 and adversary A2 follows a similar structure as in Theorem 1, with the distinction that the responses from H_{2-4} operate in the same manner.

Initialization phase: C2 generates parameters pp, C1 transmits pp and pk to A2, C2 chooses an FID^* in an adaptable manner for a sign of the challenge and records the queried messages using four lists LH_{2-4} and Lu.

Query phase: During this phase, A2 is allowed to issue different queries, which are handled similarly to those in Theorem 1. After obtaining responses, A2 proceeds to the next step.

Forgery Stage: After finish queries, A2 generates faked signature (δ^*, U^*) corresponding to the pair $(FID_{ij}{}^*, M^*)$,which $U^* = u^* P$. If $FID_{ij} \neq FID^*$,C2 outputs ⊥; otherwise, C2 replays A1 and obtains a new forged signature $(\delta_1{}^*, U_1{}^*)$ .C2 can replays A2 with the same random value $u^* \in Zq^*$ and a different response h_4^1, A2 can output another valid signature $(\delta_1{}^*, U_1{}^*)$ .At this point,C1output a=$[(\delta^* - \delta_1{}^*)/(h_4{}^* - h_4^1)]$ as the solution of the ECDLP problem. We have that formal:

$$\begin{cases} \delta^* = h_2{}^*(k^* + sh_1{}^*) + h_3{}^* u^* + h_4{}^* a \\ \delta_1{}^* = h_2{}^*(k^* + sh_1{}^*) + h_3{}^* u^* + h_4^1 a \end{cases} \tag{3}$$

We said C2 uses A2's skill to effectively deal with ECDLP problem's difficulty. M1~M3 and N1~N3 have the same meaning, forged signatures $\delta_{i1}{}^*$ and $\delta_{i2}{}^*$. we have formal as follow, where q_{H3}^1 is the number of matter H4 oracle queries.

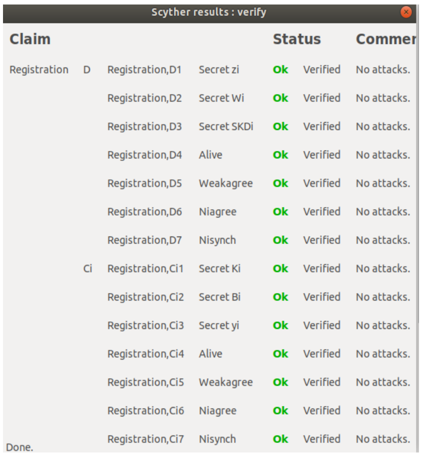
$$\Pr(N1 \wedge N2 \wedge N3) \ \geq \ (1 - \frac{1}{e}) \ \frac{\varepsilon_2}{e(\ q2 \ + \ 1)q_{H3}^1} \tag{4}$$

As a result, our system has robust EUF-CMA safety.

5 Experiment Analysis

5.1 Security analysis via Scyther simulation

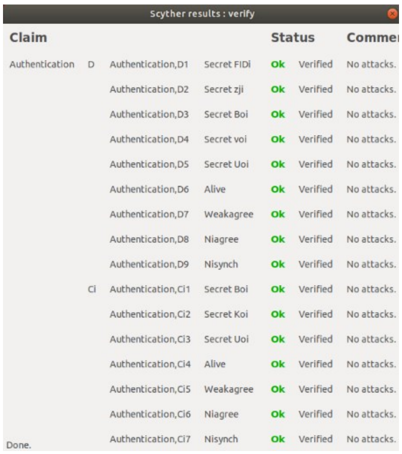
This section employs the mature automated protocol analysis tool Scyther [17] to evaluate the security of the proposed authentication scheme. The authentication negotiation process is modeled using Scyther's security policy definition language, specifying the message exchange between roles. Scyther systematically detects and analyzes security attributes based on predefined claim events, including Secret to ensure confidentiality of parameters, Alive to verify entity activity and Weakagree, Nisynch, and Niagree to assess the protocol's resilience against replay, tampering, and eavesdropping attacks. During the registration and authentication phases, Scyther generates a verification report where each row corresponds to a claim event, detailing the protocol role, claim identifier, and verification results. The results of the analysis are shown in Fig. 2 and Fig. 3, where the confidentiality of key parameters, including pseudo-identities and authentication credentials is guaranteed and the protocol is able to withstand common attacks. Furthermore, no attack paths were found within the bounded query space, indicating that the proposed scheme ensures secure authentication.



```
Scyther results : verify

Claim                                                   Status           Commer

Registration   D    Registration,D1    Secret zi        Ok    Verified   No attacks.
                    Registration,D2    Secret Wi        Ok    Verified   No attacks.
                    Registration,D3    Secret SKDi      Ok    Verified   No attacks.
                    Registration,D4    Alive            Ok    Verified   No attacks.
                    Registration,D5    Weakagree        Ok    Verified   No attacks.
                    Registration,D6    Niagree          Ok    Verified   No attacks.
                    Registration,D7    Nisynch          Ok    Verified   No attacks.
               Ci   Registration,Ci1   Secret Ki        Ok    Verified   No attacks.
                    Registration,Ci2   Secret Bi        Ok    Verified   No attacks.
                    Registration,Ci3   Secret yi        Ok    Verified   No attacks.
                    Registration,Ci4   Alive            Ok    Verified   No attacks.
                    Registration,Ci5   Weakagree        Ok    Verified   No attacks.
                    Registration,Ci6   Niagree          Ok    Verified   No attacks.
Done.               Registration,Ci7   Nisynch          Ok    Verified   No attacks.
```



```
Scyther results : verify

Claim                                                     Status           Comme

Authentication  D    Authentication,D1    Secret FIDi     Ok    Verified   No attacks.
                     Authentication,D2    Secret zji      Ok    Verified   No attacks.
                     Authentication,D3    Secret Boi      Ok    Verified   No attacks.
                     Authentication,D4    Secret voi      Ok    Verified   No attacks.
                     Authentication,D5    Secret Uoi      Ok    Verified   No attacks.
                     Authentication,D6    Alive           Ok    Verified   No attacks.
                     Authentication,D7    Weakagree       Ok    Verified   No attacks.
                     Authentication,D8    Niagree         Ok    Verified   No attacks.
                     Authentication,D9    Nisynch         Ok    Verified   No attacks.
                Ci   Authentication,Ci1   Secret Boi      Ok    Verified   No attacks.
                     Authentication,Ci2   Secret Koi      Ok    Verified   No attacks.
                     Authentication,Ci3   Secret Uoi      Ok    Verified   No attacks.
                     Authentication,Ci4   Alive           Ok    Verified   No attacks.
                     Authentication,Ci5   Weakagree       Ok    Verified   No attacks.
                     Authentication,Ci6   Niagree         Ok    Verified   No attacks.
Done.                Authentication,Ci7   Nisynch         Ok    Verified   No attacks.
```

Fig. 2. Registration phase validation results.

Fig. 3 Authentication phase validation results.

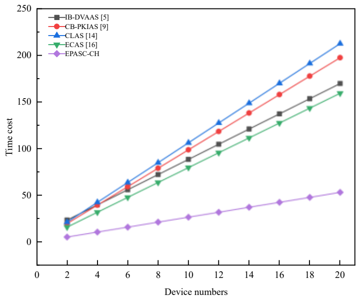
5.2 Evaluation of overheads: Computational and Communication

The experiments were conducted on VMware with Ubuntu 18.04.06. The pairing parameters are based on E-type elliptic curves with a 160-bits and a 256-bit basic field. We examine p-order curves on E(Fp), where |p| = 512 b. The cryptographic operations bilinear pairing operation, scalar multiplication of Zq^* and G-groups, add operation, and exponential. A total of 100 random experiments were performed, and the average computation time was used to determine the computational cost, denoted by TP, TM and TA as shown in Table 2.

Table 2 Cryptographic operations and running time

Cryptographic operation	Running time (ms)
Bilinear pairing (TP)	2.1343
Scalar multiplication (TM)	2.6571
Point addition operation (TA)	0.0012

In Table 3, we analyze the computational cost and communication overhead of these schemes in the signing and verification process. EPASC-CH does not need to perform pairing operations, and the total execution time for generating signatures and verifying signatures is 6TM+5TA, which is significantly better than the other schemes. The public and private keys of this scheme are 2G and $Zq*$, respectively, which are slightly larger than other schemes because its public key consists of two parts to enhance security. To ensure real-time messages, we use multi-authority centers distributed mechanism and an accumulator to assist users in multiple device signature generation and verification, which improves system efficiency and reduces computational overhead. We calculated the total time for n=20 aggregated signatures to be 253.323ms. Fig. 4 .We tested n from 2 to 20 the computational overhead of signatures.



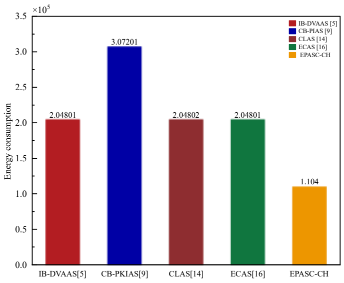


Fig. 4 Total time for aggregation of signatures. **Fig. 5** Total energy consumption comparison.

Table 3 Computational and communication overhead

Scheme	[5]	[9]	[14]	[16]	EPASC-CH
SSign	3TM+TA	3TM+2TA	4TM+2TA	3TM+TA	TM+2TA
SVerify	4TM+2TP	4TM+2TA	3TM+4TP	2TM+3TP+TA	5TM+3TA
Prkey	|Zq*|	|Zq*|	|Zq*|	|Zq*|	|Zq*|
Pukey	|G|	|G|	|G|	|G|	|2G|
SSLength	|2G|	|3G|	|2G|	|2G|	|Zq*|+|G|
ASlength	(n+1) |G|	(2n+1) |G|+|Zq*|	(n+1) |G|	(n+1) |G|	n|G|+|Zq*|

*SSign is a single signature generation, SVerify is a single signature verification, Prkey is the private key of the device, Pukey is the public key of the device, SSLength is the length of the single signature, and ASlength is the length of the aggregated signature.

5.3 Energy consumption and functional analysis

The energy consumption model of EPASC-CH focuses on the energy consumption of smart devices during computation and communication. Based on the specifications outlined in reference[18], we assume that the power of the MICA2DOT sensor is $P = 0.131mW$ and the energy consumption per byte of data sent or received is 400 nJ. P is the power of the sensor, E_{en} is the energy consumed to send/receive 1 byte of data.

Energy Consumption: (1) Computational Energy Consumption: The computational cost of the smart device is Tcp and the computational energy consumption of the smart device is

$E_{Tcp} = Tcp \cdot P$.(2) Communication energy consumption: the communication cost of a smart device is CM and the communication energy consumption of a smart device is $E_{CM} = CM \cdot E_{en}$. A comparison of the energy consumption of each scenario is shown in the figure. The total energy consumption of the device is $E_T = E_{Tcp} + E_{CM}$. Comparing the total energy consumption of this scenario with the other scenarios, the visual results are shown in Fig. 5. the device energy consumption of the proposed scheme is lower than the other four schemes. This is mainly due to the fact that EPASC-CH requires less communication than the other schemes to achieve message authentication and privacy protection, which effectively reduces the communication energy consumption. In addition, Table 3 shows that EPASC-CH has the lowest communication overhead and thus its communication energy consumption is reduced accordingly. From Fig. 4 total time for aggregation of signatures., it can be seen that the computational overhead of EPASC-CH is lower than the other four schemes, and thus the computational energy consumption is also lower.

Combining the analysis results of computation overhead and communication overhead, it can be concluded that this scheme has significant advantages in both computation and communication, while the computation energy consumption and communication energy consumption are both at a lower level. Therefore, this scheme can effectively reduce the total energy consumption of the device and is particularly suitable for communication devices in smart home environments.

Functional Analysis: Table 4 we first compare the functional differences between the proposed scheme. Our scheme focuses on improving authentication efficiency while protecting device privacy during information authentication. It adopts a distributed architecture based on multiple authorization centers, where each authorization center collaborates to generate the system key and manages it in a decentralized manner, which improves the trustworthiness of the system, effectively prevents a certain center from unilaterally forging and tampering with the data, increases the cost and difficulty of the attack, and avoids a single point of failure. When more than one device sends a signature authentication request at the same time, multiple authorization center work together to generate a message authentication signature, using an accumulator to assist in user verification, while avoiding pairing operations, further improving the efficiency of the system. Using the anonymity mechanism, attackers cannot associate the same device, effectively protecting the privacy of the device and ensuring unlinkability. Therefore, EPASC-CH not only has stronger security, but also better anonymity.

Table 4 Function Comparison

Scheme	[5]	[9]	[14]	[16]	EPASC-CH
Anonymity	√	×	√	×	√
Unlinkability	×	×	×	×	√
Multi-authority	×	×	×	×	√
Bilinear	√	√	√	√	×
Replay Attack	√	√	√	√	√

6 Conclusion

In this paper, an efficient privacy-preserving authentication and secure communication protocol based on smart home environment is proposed. The scheme adopts a distributed architecture based on multiple authorization centers and incorporates accumulator technology to assist user authentication, thus significantly improving the authentication efficiency while ensuring the overall security of the system. To further enhance the device privacy protection, an anonymization mechanism is designed to effectively avoid the leakage of device information. By comprehensively analyzing the computational overhead, communication overhead and energy consumption, the experimental results show that the scheme can significantly improve the computational efficiency and effectively reduce the communication overhead, which has better practicality and performance advantages. In the future, we plan to optimize the scheme by introducing lightweight cryptographic algorithms combined with advanced privacy-preserving technologies to improve authentication efficiency and reduce computational costs.

Acknowledgments. This work was Funded by Science Research project of Hebei Education Department (QN2025014).

References

1. Nyangaresi, V.O., Ogundoyin, S.O.: Certificate based authentication scheme for smart homes. In: 2021 3rd Global Power, Energy and Communication Conference (GPECOM), pp. 202–207. IEEE (2021).
2. Philip, S.J., Luu, T.J., Carte, T.: There's no place like home: Understanding users' intentions toward securing internet-of-things (IoT) smart home networks. Computers in Human Behavior 139, 107551 (2023).
3. Meshram, V.R., Pocchi, R.: A review on wireless smart home automation using IoT. International Journal of Scientific Research in Science and Technology (2021).
4. Chen, C.M., Liu, S., Li, X., Islam, S.K.H., Das, A.K.: A provably-secure authenticated key agreement protocol for remote patient monitoring IoMT. Journal of Systems Architecture 136, 102831 (2023).
5. Deng, L., Wang, T., Feng, S., Qu, Y., Li, S.: Secure identity-based designated verifier anonymous aggregate signature scheme suitable for smart grids. IEEE Internet of Things Journal **10**(1), 57–65 (2022).
6. Yang, H., Guo, Y., Guo, Y.: A PUF-based three-party authentication key establishment scheme for fog-enabled smart home. Pervasive and Mobile Computing 95, 101843 (2023).
7. Shao, X., Guo, Y., Guo, Y.: A PUF-based anonymous authentication protocol for wireless medical sensor networks. Wireless Networks **28**(8), 3753–3770 (2022).
8. Yu, X., Zhao, W., Tang, D.: Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing. Journal of Systems Architecture 126, 102457 (2022).
9. Hou, Y., Xiong, H., Huang, X., Kumari, S.: Certificate-based parallel key-insulated aggregate signature against fully chosen key attacks for industrial Internet of Things. IEEE Internet of Things Journal **8**(11), 8935–8948 (2021).
10. Tang, F., Liu, W., Feng, Z., Ling, G.: Improved Scheme of Practical Byzantine Fault tolerance based on Multi-authority Aggregated signature. Journal of Chongqing University of

Posts & Telecommunications (Natural Science Edition) **34**(4), 705–711 (2022).

11. Fernandes, E., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 636–654. IEEE (2016).

12. Debroy, S., Calyam, P., Nguyen, M., Neupane, R.L., Mukherjee, B., Eeralla, A.K., Salah, K.: Frequency-minimal utility-maximal moving target defense against DDoS in SDN-based systems. IEEE Transactions on Network and Service Management **17**(2), 890–903 (2020).

13. Khan, M.A., Salah, K.: IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems 82, 395–411 (2018).

14. Kumar, P., Kumari, S., Sharma, V., Li, X., Sangaiah, A.K., Islam, S.K.H.: Secure CLS and CL-AS schemes designed for VANETs. The Journal of Supercomputing 75, 3076–3098 (2019).

15. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13, 361–396 (2000).

16. Xu, Z., He, D., Kumar, N., Choo, K.K.R.: Efficient certificateless aggregate signature scheme for performing secure routing in VANETs. Security and Communication Networks 2020, 5276813 (2020).

17. Cremers,C.:The Scyther Tool. CISPA Helmholtz Center for Information Security, Saarbrücken, Germany (2014). [Online]. Available.

18. Kavitha, A., Koppala Guravaiah, R., Leela Velusamy, S., Suseela, S., Kumar, D.: DR-NAP: Data reduction strategy using neural adaptation phenomenon in wireless sensor networks. International Journal of Communication Systems **36**(8), e5467 (2023).