Computing (2023) 105:2511–2537
https://doi.org/10.1007/s00607-023-01188-4

REGULAR PAPER

A provable secure and lightweight ECC-based authenticated key agreement scheme for edge computing infrastructure in smart grid

Cong Wang[1] · Peng Huo[1] · Maode Ma[2] · Tong Zhou[1] · Yiying Zhang[1]

Received: 7 May 2022 / Accepted: 28 May 2023 / Published online: 27 June 2023

Abstract

With the gradual maturity of Smart Grid (SG), security challenges become the most important issues that need to be addressed urgently. In recent years, many schemes adopt mutual authentication and key agreement to ensure secure communication in SG. However, most existing methods have their own shortcomings in either security or efficiency which make them difficult to satisfy the security requirements of SG. In this paper, we propose a provable secure and lightweight authenticated key agreement scheme based on the Elliptic Curve Cryptosystem (ECC) for the cloud-edge-end collaboration SG communication network. The proposed scheme can guarantee the security of the communication and provide anonymity for both the edge server and the smart meters. The anonymity of the smart meters can be provided by a pseudonym mechanism. By rigorous security analysis, the proposed scheme can resist to the typical attacks including replay attacks and identity forgery attacks. The security properties are also evaluated by using the Canetti and Krawczyk (CK) adversary model. The performance simulation results denotes that the proposed scheme not only owns stronger security functions but also improves computation efficiency by 24.5% on average than other four schemes. By simulation results, the proposed scheme has been shown to hold high efficiency.

Keywords Authentication · ECC · Key Management · Smart Grid

✉ Maode Ma
acadmmd@gmail.com

[1] Tianjin University of Science and Technology, Tianjin, China

[2] College of Engineering, Qatar University, Doha, Qatar

1 Introduction

The power system is one of the most important physical infrastructures in human society [1]. The Smart Grid (SG) is defined as the electric power network characterized by an efficient and reliable infrastructure with the use of sophisticated and modern control [2], which is built on the base of an integrated, high-speed two-way communication network. By the application of various system technologies such as advanced control methods, the SG becomes much more economical, efficient, and safe to use. Edge computing is an effective way to provide cloud services to the edge devices and further improving the quality of service to end-users by reducing the computing intensity of the end-users. The SG applies edge computing for data processing for the power equipment to migrate computing resources to the equipment side and reduces the overhead and delay caused by the data transmission.

The communication networks in the SG face various kinds of security risks including malicious attacks from the outside and data loading due to excessive amounts of information [3]. Mutual identity authentication can effectively prevent two types of attacks from the outside and verify the identities of the participants in the communication to secure the communication [4] in the SG. It is very important to guarantee secure communication between an Edge Server (ES) and a Smart Meter (SM) because there could be many risks to impair the communication. Therefore, some security measures should be taken to protect communications in the SG.

Secure mutual authentication is the first critical step in deterring attackers. In the past decade, many solutions for authentication and key agreement schemes for the communications in the SG have been proposed. Some schemes in [5–7] introduced a trusted third party (TTP) or a trust anchor to participate in the mutual authentication process. However, two of them don't provide a pseudonym mechanism for SMs and ESs, which is likely to cause ID leakage risk. Some schemes suffer from malicious attacks such as Man-In-The-Middle (MITM) attacks in [8], time synchronization attack in [9], and lack of providing session key security in [3]. Schemes in [10, 11] require high storage and computation costs, it will reduce the robustness of the SG. Schemes in [12, 13] proposed a lightweight authentication process, but these schemes can't keep session key security under the Canetti and Krawczyk (CK) adversary model.

Due to the requirements for high reliability and security for the communications in the SG, each SM needs to be authenticated by an ES before entering the communication network in the SG. Each ES will also negotiate a session key with all SMs. If a mutual authentication is not performed between the SM and the ES, there will be a risk of paralysis of the communications in the SG. Most existing authentication schemes for the communications in the SG have some drawbacks in terms of efficiency and security, which are not feasible for frequent interactions between the SM and the ES. Therefore, a secure mutual authentication process with a shared session key negotiated in advance to provide secure communications between the SM and the ES will be particularly critical.

So the contribution of the paper is as follows:

(1) We propose a Provable Secure and Lightweight Authenticated key agreement scheme called PSLA which is based on the Elliptic Curve Cryptosystem (ECC) for the cloud-edge-end collaboration SG communication network. The proposed PSLA scheme can guarantee the security of the secret key by the elliptical encryption and the anonymity at the ESs and the SMs by introducing a pseudonym mechanism. Our unique contributions are that our PSLA scheme has better performance without sacrificing any safety features.
(2) The CK adversary model is used to analyze the security of the PSLA scheme which is secure under the random oracle model. And by informal security analysis. the PSLA scheme can resist to the typical attacks including replay attacks, MITM attacks, and identity forgery attacks.
(3) Performance evaluation shows that the computation authentication delay and the communication cost of the PSLA scheme are much more lower than other solutions.

The rest of the paper is organized as follows. In Sect. 2, we introduce the related works. In Sect. 3, we introduce the system model. Then in Sect. 4 we explain the authentication process of the proposed PSLA scheme in details. In Sect. 5, we conduct security analysis to prove that our PSLA scheme can resist various attacks under an ideal circumstance. The performance evaluation on the PSLA scheme is presented in Sect. 6. Finally, we summarize our research work in Sect. 7.

2 Related work

Due to limited resources of the devices in the communication networks in the SG, traditional public key infrastructure-based schemes are obviously not applicable for the communications in the SG. Mutual authentication is important to prevent attackers getting effective resources, and ensure the security of SG systems [14] [15]. Recently, numerous interesting authentication schemes for secure communications in the SG have been presented. Yang et al. have proposed a lightweight anonymous mobile user authentication scheme for the SG in [16]. However, Yang's solution needs too many redundant steps in the registration phase causing a lot of time consumption and data loading. Vanga Odelu et al. Have proposed a provably secure authenticated key agreement scheme for the SG in [17], which can greatly reduce the complexity of the registration, while the time consumption at the SM is still huge which is closed to 500 ms. Kuljeet Kaur et al. have proposed a secure lightweight and privacy preserving authentication scheme for V2G communications in the SG in [18], which separates the mutual authentication process from the key agreement process. If an attacker launches a distributed attack on the server, it can cause other devices failure to establish a connection with the server. Abbasinezhad Mood et al. have designed a scheme for isolated SMs in [19] that does not require the intervention of a service provider in the SG in the key agreement process. But, this scheme uses many scalar multiplication operations to lead its computation delay very high. Chen et al. have proposed a self-authenticated key allocation scheme for the SG in [20]. However, Chen's scheme could be subject to a DoS attack with a failure to secure the session key under the CK adversary model [21].

To overcome the security vulnerabilities in the SG's communication, Fouda et al. have proposed a lightweight message authentication scheme for the SG communication in [22], by which communication parties can reduce the total communication delay by avoiding using high communication cryptographic operations and reducing unnecessary signaling messages. But, the scheme could not resist to the DoS attacks. Abdallah has proposed a scheme to utilize the lattice-based homomorphic cryptosystem in [23], while the lattice-based encryption system can incur immense computing overload and communication cost. By the Gope's scheme in [24], before receiving messages, the ES has to check the validity of the SM, which increases the computational complexity linearly with the number of SMs [25]. Kumar et al. have proposed a lightweight authentication and key agreement scheme in [9] that enables trust, anonymity, integrity and adequate security in the domain of smart energy networks. But, it cannot resist ephemeral secret leakage and suffers from synchronization attacks. Tsai et al. have proposed a secure anonymous key distribution scheme for the SG by combining the advantages of identity-based signature and identity encryption in [26], but it could not resist to the MITM attacks without the ability to provide the session-key security and credentials' privacy. Braeken et al. have proposed a provably secure key agreement model for the communications in the SG in [27] which can hardly work against malicious insider attackers [28]. Physical uncloneable functions (PUFs) have gained popularity as a primitive against physical attacks [29, 30]. But they are susceptible to the modeling attacks. PUF is a non-destructible method for protecting the security of integrated circuit chips, but it is vulnerable to modeling attacks based on machine learning [31, 32].

Compared with other cryptographic operations, the ECC can reduce the computational load effectively, which can also be used in the design of the security schemes for the SG. Sureshkumar et al. improved mutual authentication and key agreement mechanism based on ECC [33], but introduced a third party into the key agreement phase may still be threatened by tracking attacks. In order to solve the security risks caused by tracking attacks, Baghestani et al. [34] and Chaudhry et al. [35] don't let the third party participate in the key agreement process. However, both schemes can not resist impersonation attack. Mahmood's scheme in [36] can achieve the mutual authentication with a low computation and communication cost with its weakness against both perfect forward insecurity and private keys leakage [30]. Mohammadali et al. have proposed an identity-based key establishment protocol in [13], which employs elliptic curves for the Advanced Metering Infrastructure in the SG. But, it cannot provide users' anonymity. Wazid et al. have proposed a three factor user authentication scheme in [37] for a renewable energy based SG environment, by employing lightweight cryptographic computations such as one-way hash functions, XOR operations and ECC. Biometrics is used for the identity authentication elements, which leads to a low adaptability and scalability. Badra et al. have proposed a privacy preserving data aggregation scheme in [38], which employs a blind factor to resist the attacks from the internal members. To improve efficiency, it has adopted a very simple authentication process, but it often leads to be less secure in fact, vulnerable to replay attacks. Jo et al. have proposed an anonymous signature-based authentication with a key exchange for an IoT-enabled SG environment in [39]. However, it is unable to defend the MITM and the impersonation attacks. In addition, it does not render the anonymity feature of the SMs. Khan et al. have designed a password-based lightweight key agreement framework (PALK) by using the ECC for the SG in [6]. Chaudhry has analyzed the PALK scheme to disclose that the PALK protocol is incorrect at the login and authentication phase due to the ECC operations in [35]. Chaudhry has proposed an improved scheme in the same domain. But, the new scheme is not free from the PKI challenges with a high computation cost. Khan et al. have proposed a key agreement and lightweight authentication mechanism for the communications in the SG in [40]. However, by the scheme, the communication goes among the user, the trusted third party and the server, that could incur security concerns due to the third party application. To mitigate the issues, a modified mutual authentication and key agreement mechanism by using the ECC has been proposed for the SG in [33], by which the system is secure under the CK adversary model with an additional security scheme required for the control center to organize and initialize the real-time information.

Some other schemes in [10–12] can cause a significant drain on communication and computation resources due to their high computation complexity. Dariush et al. have proposed an solution named as Anonymous ECC-Based Self-certified Key distribution scheme (AESK) for the SG in [10]. However, it relies on tamper-proof security modules to implement certain security functions such as perfect confidentiality. Qi et al. have proposed a scheme named as Two-pass Privacy Preserving Authenticated key agreement (TPPA) for the SG based on the elliptic curve Qu-Vanstone implicit certificates with a trusted third-party participation in the SG in [12]. But, under the CK adversary model [34], the TPPA cannot guarantee the security of the session key. Jia, X. et al. have proposed a solution named as Provably Identity-based Anonymous Authentication scheme (PIAA) for mobile edge computing in [11]. But, the PIAA adopts too many scalar multiplication in the certification process, which cannot meet the current low-time consumption requirements for the SG. Xiang et al. in [7] proposed a Secure Privacy-preservation Authentication Key agreement scheme (SPAK) in SG communications without providing anonymity of SMs. Thus, how to provide a better privacy-preservation and efficient authentication process deserves further study.

3 System model and preliminary

In this section, we discuss the details of the system model and security model under the study with the mathematical background.

3.1 System model

The architecture of the cloud-edge-end collaboration communication network in the SG is a three-layer network as shown in Fig. 1 including a Home Area Network (HAN) as the end layer, an Edge Mesh Network (EMN) as the edge computing layer

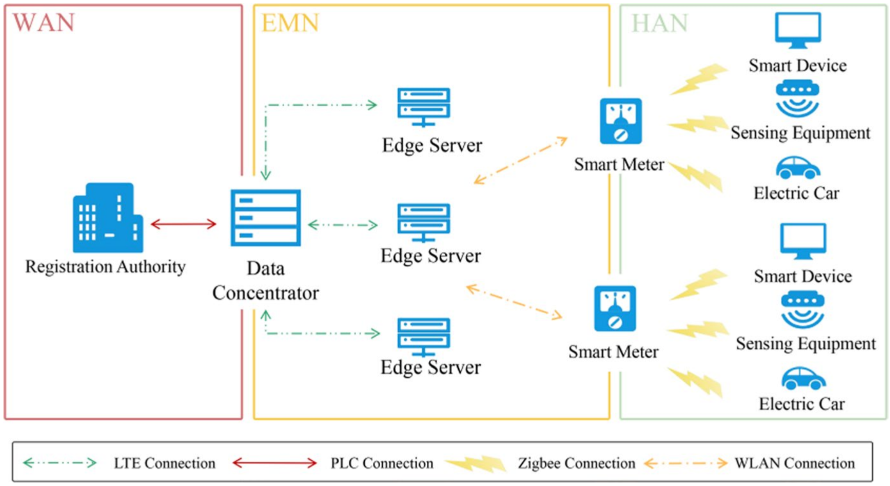


Fig. 1 Architecture of communication system in SG

and a Wide Area Network (WAN) as the cloud layer [9]. Within the HAN, a *SM* acts as a home gateway that collects energy consumption reading from devices such charging station through Zigbee connection, sends collected readings to the edge sever acting as the aggregator or utility controller through WLAN connection. The EMN supports the communication between the *ESs* and the *SMs*. Multiple EMNs can be connected into the WAN via a wireless mesh network. Given the wireless mesh network between metering gateways and/or *SMs*, the *ES* could gather all of the required data in periodic intervals and then send them to the data concentrator (*DC*) through LTE connection. Finally, the DC forwards all the data to the Registration Authority (*RA*) located near the utility service provider in the WAN, which typically communicates over the powerline communication (PLC).

The *ES* and the *SM* should get mutual authenticated for a session key agreement. Before the authentication, the *ES* and the *SM* should register with the *RA*. The communication between the *SM* and devices such as charging station, the *ES* and the *DC* is the connection of Zigbee and LTE respectively, whose communication security has been guaranteed by mature mechanisms. The connection between the *DC* and the *RA* is wired. Since the two-way communication between *ESs* and *SMs* is a type of wireless communication, it is easy to be exposed to public, attackers may launch some malicious attacks, such as MITM attacks, replay attacks, and DoS attacks, etc.

3.2 Security model

A widely known and commonly accepted adversarial model known as the CK threat model [21] has been adopted in this work. The security model is a game played between resister $\mathcal{D}$ and opponent $\mathcal{A}$. The game is modeled by a probabilistic polynomial-time (P. P. T.) turing machine. Let $\mathit{\Pi}\ l\Omega$ represents an instance $\updownarrow$ of participant

Ω, where Ω represents a *SMi* or an *ESj*. In the hypothetical model, $\mathcal{A}$ can perform a series of operations. In order to reduce the attack difficulty of $\mathcal{A}$, $\mathcal{D}$ must reply in accordance with the following regulations.

Hash(m): $\mathcal{A}$ performs a hash query on message *m*. $\mathcal{D}$ checks whether message *m* has been asked before. If yes, the corresponding *h(m)* is returned. If not, a random number *r* is returned. Then this record (*m,r*) is stored in the list.

Execute(*SMi*,*ESj*): Execute query is defined on the basis of an passive eavesdropping attack which returns a copy of the information passed between the participants in the system. $\mathcal{A}$ may get the session key by collecting the information passed by both parties.

Extract(ID): This query models$\mathcal{A}$'s ability of corrupting a legal entity and obtaining the private key of it. When $\mathcal{A}$ queries this oracle with identity ID, the oracle returns the private key corresponding to ID.

Send(P,m): Send query is defined on the basis of a modification attacks, replay attacks, simulation attacks, etc. By Send query,$\mathcal{A}$ sends message m to P and will receive a response message by P.

Test(*SMi*,*ESj*): $\mathcal{A}$ can submit this query for only once. When P receives a Test query, an unbiased coin c is flipped. If $c=1$, the actual session key is returned. Otherwise, a random value with the same length is returned to$\mathcal{A}$.

Definition 1 The scheme introduces an arbitrary polynomial function $P(n)$. If the equation $\mu(n) = O(\frac{1}{P(n)})$ holds, then the function $\mu(n)$ can be ignored.

Definition 2 Calculating Diffie-Hellman (CDH) problem. Suppose that given three elements,P,xP,$yP \in G$, which are used to calculate the value of $(xy)P \in G$. Due to x,$y \in Z$, any P.P.T. attacker cannot know these elements in a special way.

Definition 3 The AKA-Authenticated Key Agreement-Security. If the advantages $\mathrm{Adv}_{\sum}^{\mathcal{A}}(\mathrm{E}_{\mathrm{AKA}})$ provided to any P.P.T. attacker $\mathcal{A}$ are negligible, then it can prove that the scheme is AKA safe in the mobile internet environment.

Definition 4 The Mutual Authenticated-Security. If the advantages $Adv_{\sum}^{\mathcal{A}}(E_{MA})$ provided to any P.P.T. attacker $\mathcal{A}$ are negligible, it can prove that the scheme is MA-Security in the mobile internet environment.

3.3 Elliptic curve cryptosystem and bilinear pairings

A large prime number q will be chosen. And the elliptic curved $Eq(a,b)$ is defined by the equation $E{:}y^2 = x^3 + ax + b(modq)$ where $a,b,x,y \in Z$, $4a^3$+$27b^2$ $(modq) \neq 0$.

Choosing a point $P \in Ep$ as the generator of G. Scalar multiplication has been defined as $nP = P + P + P + \cdots + P(n$ times), where $n \in Z$. Let GT be a multiplicative cyclic group of the same order q.The map e: $G \times G \rightarrow GT$ is proved to be an admissible bilinear map, if the following conditions could be resolved.

(1) Bilinearity: $e(aP,bQ)=e(P,Q)^{ab}$, for all $P,Q \in G$ and for all $a,b \in Z$.
(2) Non-degeneracy: There exists a $P \in G$, such that $e(P,P) \neq 1$.
(3) Computability: For all $P,Q \in G, e(P,Q)$ can be efficiently computed.

4 The proposed PSLA scheme

In this section, we illustrate the proposed PSLA scheme in details. The primitive notations pertaining to the PSLA scheme are demonstrated in Table 1.

4.1 Overview

The proposed PSLA scheme consist of five phases: system initialization, edge sever *ESj* registration, smart meter *SMi* registration, mutual authentication and key agreement between the *ESj* and the *SMi*, as shown in Fig. 2. The system initialization is issued by the *RA* for generation of all the required system parameters. All the users in the system can receive the system parameters. When one new ESj joins the system, the *ESj* firstly registers with the *RA* through the *DC* to get its pseudonym. Similarity, when one new *SMi* joins the system, the *RA* registers with the *RA* through the *ESj* and the data concentrator *DC* to get its pseudonym. At this phase, the *ESj* and the *DC* only forward the message between the *RA* and the *SMi*. After the registration of the *SMi*, the authentication phase between the *ESj* and the *SMi* is triggered. Once the *ESj* and the *SMi* get mutual authentication, at the key agreement phase a

Table 1 Notations

Notations	Description
SM	Smart meter
ES	Edge server
RA	Registration authority
$\mathrm{ID_i}$	Real identity of smart meter
$\mathrm{ID_j}$ ID_j	Real identity of edge server
$\mathrm{SID_i}$	Anonymous identity of smart meter
$\mathrm{SID_j}$	Anonymous identity of edge server
q	Large prime number
G	An additive cyclic group
G_T	A multiplicative cyclic group
P	Generator of G
s	Private key of RA
P_{pub}	Public key of RA
$\mathrm{H_i}$ ($i=0, 1, 2, 3$)	Secure hash function
R_{i}, x, y	Random numbers in Z_q^*
SK	Session key
T_{sm}	Timestamp of smart meter (64-bits)
T_{s}	Timestamp of edge server (64-bits)

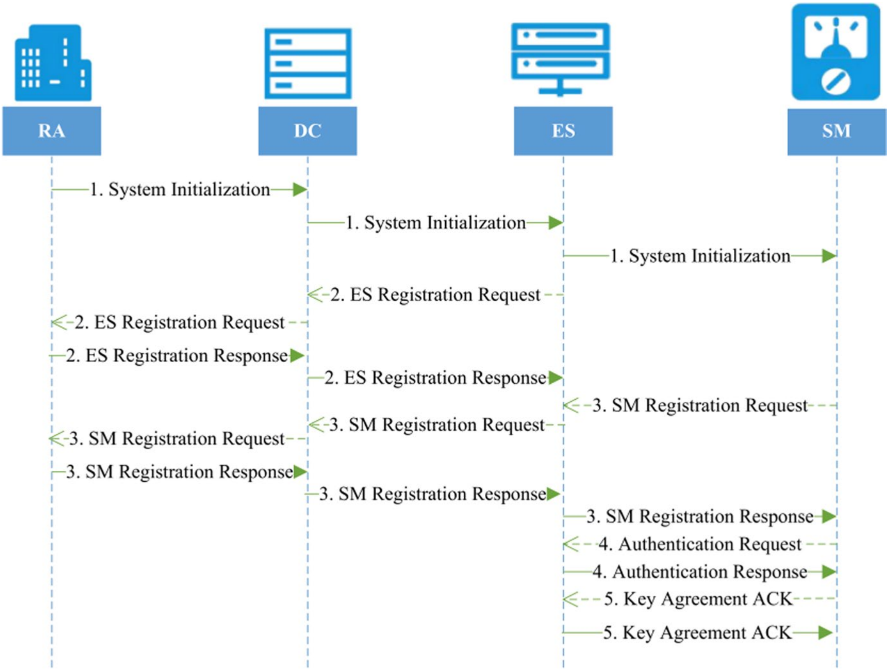

RA
DC
ES
SM
1. System Initialization
1. System Initialization
1. System Initialization
2. ES Registration Request
2. ES Registration Request
2. ES Registration Response
2. ES Registration Response
3. SM Registration Request
3. SM Registration Request
3. SM Registration Request
3. SM Registration Response
3. SM Registration Response
3. SM Registration Response
4. Authentication Request
4. Authentication Response
5. Key Agreement ACK
5. Key Agreement ACK

Fig. 2 The flow chart of the proposed PSLA scheme

session key can be generated for further secure communication between the *ESj* and the *SMi*.

4.2 System initialization phase

In the initialization phase, the *RA* initializes all the required system parameters for the proposed PSLA scheme as follows:

(1) On the non-singular elliptic curve, the *RA* chooses a large prime q, a cyclic additive group G, a multiplicative group $G \times G \rightarrow GT$, a key generator P as the generator of G and GT.The *RA* calculates a bilinear mapping e: $G \times G \rightarrow GT, g = e(P,P)$, which is bilinear, non-degenerate and computable.

(2) *RA* selects a random number $s \in Z^*q$ as its private key, and then calculates its own public key $Ppub = sP$.

(3) *RA* selects four secure hash functions $H0$: $\{0, 1\}^* \times G \rightarrow Z^*q, H1$: $\{0, 1\}^* \times G \rightarrow Z^*q, H2$: $\{0, 1\}^* \times G \rightarrow Z^*q, H3$: $\{0, 1\}^* \times G \rightarrow Z^*q$.

(4) *RA* issues a set of system parameters (*G*,*GT*,*q*,*P*,*Ppub*,*H0*,*H1*,*H2*,*H3*). *H0* and *H1* are used to calculate the pseudonym of the *SM* and the *ES*, *H2* is used to encrypt the information in the mutual authentication process. And *H3* is used to calculate a session key.

4.3 Edge server registration phase

(1) Edge Server *ESj* selects its own real user *IDj* and sends a registration request to the *RA* through the *DC*.

(2) To ensure that the mutual anonymous identity authentication between *SMi* and *ESj* needs the *RA* to generate a pseudonym before the authentication. And the *RA* calculates pseudonym $SIDj = \frac{P}{s+H_1(ID_j)}$ for the *ESj*.

(3) *RA* sends *SIDj* to *ESj*. The process of *ESj* registration is shown in Fig. 3 and Algorithm 1.

```
Algorithm 1 $ES_j$ registration
Input: $H_1, P, s, ID_j$;
Output: $SID_j$;
$SID_j = \frac{P}{s+H_1(ID_j)}$;
return $ES_j \leftarrow (KID_j, K_j)$;
```

4.4 Smart meter registration phase

1) Smart meter *SMi* selects its own real user ID and sends the *IDi* with a registration request to the *RA* through the *ESj* and the *DC*.

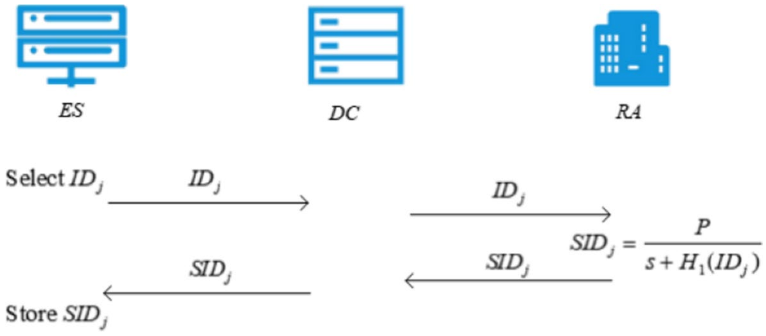


Fig. 3 The process of *ESj* registration

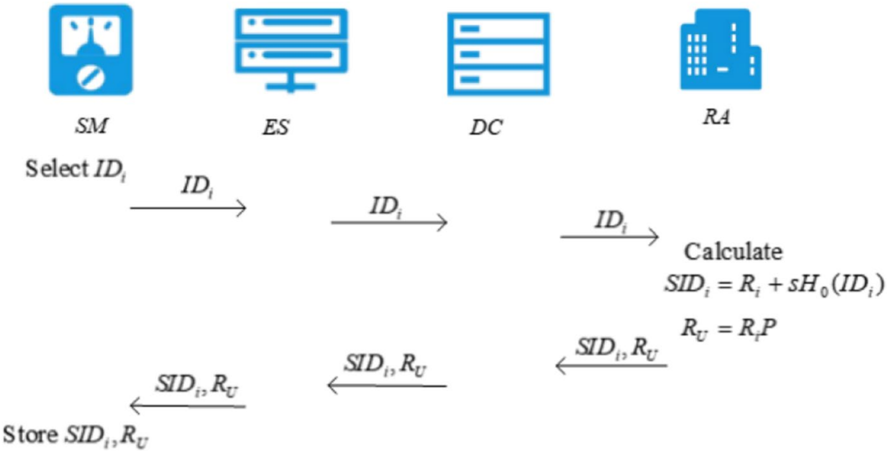
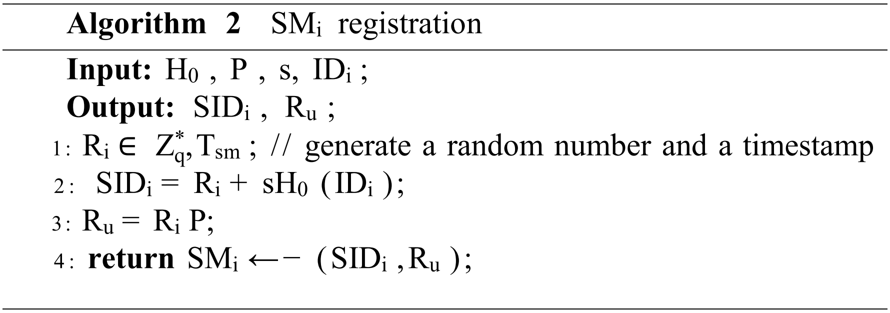


Fig. 4 The process of *SMi* registration

2) After receiving the *SMi*'s registration request, the *RA* generates a random number $Ri \in Z^*q$, and generates a pseudonym $SIDi = Ri + sH0(IDi)$,$Ru = RiP$ for *SMi*. And *Ru* will be used for subsequent mutual authentication with the *ESj*.

3)*RA* sends back (*SIDi*,*Ru*) to the *SMi* through the *ESj* and the *DC*. The process of *ESj* registration is shown in Fig. 4 and Algorithm 2.



```
Algorithm 2   SMi registration
Input: $H_0$, $P$, $s$, $ID_i$;
Output: $SID_i$, $R_u$;
$R_i \in Z_q^*, T_{sm}$; // generate a random number and a timestamp
$SID_i = R_i + sH_0(ID_i)$;
$R_u = R_i P$;
return $SM_i \longleftarrow (SID_i, R_u)$;
```

4.5 Authentication phase

1) *SMi* chooses a random number $x \in Z$ to calculate $gx = g^x$, $X = xP$,$M = x(Ppub + H1(IDj)P)$, $N = H2(gx) \oplus (IDi \| Ru \| X \| Tsm)$, $\sigma = SIDi + xH2(IDi \| Ru \| X \| Tsm)$,where the random number *Ru* is sent by the *RA* in the registration phase.And *Tsm* is the current time. *SMi* sends (M, N, σ, *Tsm*) to *ESj* on a public channel.

2) When *ESj* receives a session request from *SMi*, *ESj* first checks the freshness of the timestamp *Tsm*. Then it calculates$\left(g_x^n\right) \oplus N$. $g_x^n = g_x$ proved as (1). Then, ES_j calculates $L = (R_u + P_{pub}H_0(ID_i)) + X'H_2\left(ID_i'||R_u'||X'|T_{sm}\right)$ and verifies whether L is equal to σP as shown in (2). If it is wrong, *ESj* will reject the authentication request and terminate the session. Otherwise, ESj will choose a random number $y \in Z$ and calculate $Y = yP$. *ESj* sets up session keys $SKS-SM=H3(yX \parallel Ts)$, $S=H1(SKS-SM)$, uses current time as *TS* and sends (*S*, *Y*, *Ts*) back to *SMi*.

$$\begin{aligned} g_x' &= e\left(M, SID_j\right) = e\left(x\left(P_{pub} + H_1\left(ID_j\right)P\right), \frac{P}{s + H_1\left(ID_j\right)}\right) \\ &= e\left(x\left(sP + H_1\left(ID_j\right)P\right), \frac{P}{s + H_1\left(ID_j\right)}\right) \\ &= e\left(x\left(s + H_1\left(ID_j\right)\right)P, \frac{P}{s + H_1\left(ID_j\right)}\right) \\ &= e(P,P)^{x\left(s+H_1\left(ID_j\right)\right)\cdot\frac{1}{s+H_1\left(ID_j\right)}} \\ &= e(P,P)^x = g^x \end{aligned} \tag{1}$$

$$\begin{aligned} \sigma P &= \left(SID_i + xH_2\left(ID_i||R_u||X||T_{sm}\right)\right)P \\ &= \left(\left(R_i + sH_0(ID_i\right) + xH_2\left(ID_i||R_u||X||T_{sm}\right)\right)P \\ &= \left(R_iP + sPH_0\left(ID_i\right) + xH_2\left(ID_i||R_u||X||T_{sm}\right)\right)P \\ &= \left(R_u + P_{pub}H_0\left(ID_i\right) + XH_2\left(ID_i||R_u||X||T_{sm}\right)\right) \end{aligned} \tag{2}$$

3) After *SMi* receives the response from *ESj*, *SMi* will first check the freshness of the timestamp *TS*. It then sets the session key $SKSM-S=H3\ (xY \parallel T_s)$. *SMi* will check whether $H1(SKSM-S)$ is equal to *S*. If they are equal, the authentication with *ESj* is completed, and the agreement of the session key is ensured. In this phase, *ESj* and *SMi* will perform mutual authentication and negotiate a public session key *SK* for subsequent communications as shown in Fig. 5. The authentication details are shown in Algorithm 3 and 4.

4.6 Session key agreement phase

1) In order to ensure the agreement of the negotiated session key, if *SMi* authenticates *ESj* successfully, *SMi* needs to send back a verification message encrypted by the session key to *ESj*. *SMi* sends the session key agreement information $A=E(Tnow, SKu-s)$, where *Tnow* is the current time, back to *ES*.

2) After *ESj* receives the session key agreement information *A* sent by *SMi*, *ESj* needs to use the session key $SKs-u$ to decrypt *A*, and then determines whether the content in *A* is a timestamp *Tnow* and checks the freshness of it. If so, *ESj* can guarantee that *SMi* holds the same session key. Until now, the agreement of session key is over.

Algorithm 3 Encrypt(R_u, ID_i) //*SMi* encrypts authentication information

```
Input: $H_1, H_2, P_{pub}, P, s, R_u, ID_i$;
Output: $g_x, X, M, N, \sigma, T_{sm}$;
$x \in Z_q^*$; //generate a random number
$g_x = g^x$;
$X = xP$;
$M = x(P_{pub} + H_1(ID_j)P)$;
$N = H_2(g_x) \oplus (ID_i \parallel R_u \parallel X \parallel T_{sm})$;
$\sigma = SID_i + xH_2(ID_i \parallel R_u \parallel X \parallel T_{sm})$;
return $(g_x, X, M, N, \sigma, T_{sm})$;
```

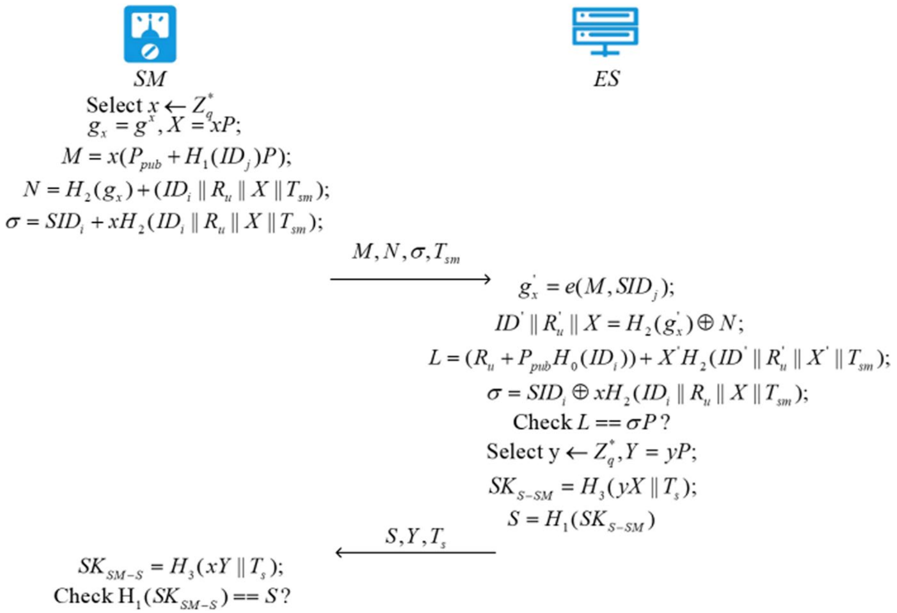

SM
ES
Select $x \leftarrow Z_q^*$
$g_x = g^x, X = xP$;
$M = x(P_{pub} + H_1(ID_j)P)$;
$N = H_2(g_x) + (ID_i \parallel R_u \parallel X \parallel T_{sm})$;
$\sigma = SID_i + xH_2(ID_i \parallel R_u \parallel X \parallel T_{sm})$;
M, N, σ, T_{sm}
$g_x^{'} = e(M, SID_j)$;
$ID^{'} \parallel R_u^{'} \parallel X = H_2(g_x^{'}) \oplus N$;
$L = (R_u + P_{pub}H_0(ID_i)) + X^{'}H_2(ID^{'} \parallel R_u^{'} \parallel X^{'} \parallel T_{sm})$;
$\sigma = SID_i \oplus xH_2(ID_i \parallel R_u \parallel X \parallel T_{sm})$;
Check $L == \sigma P$?
Select y $\leftarrow Z_q^*, Y = yP$;
$SK_{S-SM} = H_3(yX \parallel T_s)$;
$S = H_1(SK_{S-SM})$
S, Y, T_s
$SK_{SM-S} = H_3(xY \parallel T_s)$;
Check $H_1(SK_{SM-S}) == S$?

Fig. 5 Mutual authentication process of PSLA

Algorithm 4 *ESj* authenticates *SMi*

```
Input: $g_x, X, M, N, \sigma, T_{sm}, SID_j$;
Output: $(S, Y, T_s)$ or terminate;
$SM_i$.Encrypt($R_u, ID_i$)
Check the freshness of the $T_{sm}$;
$g'_x = e(M, SID_j)$;
$ID'_i \parallel R'_u \parallel X' \parallel T_{sm} = H_2(g'_x) \oplus N$;
$L = (R_u + P_{pub}H_0(ID_i)) + X'H_2(ID'_i \parallel R'_u \parallel X' \parallel T_{sm})$;
if($L = \sigma$P) then
    return accept;
    $y \in Z_q^*, T_s$;  //generate a random number and a timestamp
    $Y = yP$;
    $SK_{S-SM} = H_3(yX \parallel T_s)$;
    $S = H_1(SK_{S-SM})$;
```

Algorithm 5 *SMi* authenticates *ESj*

```
Input: $S, Y, T_s, x$;
Output: complete or terminate;
Check the freshness of the $T_s$;
$SK_{SM-S} = H_3(xY \parallel T_s)$;
if($H_1(SK_{SM-S}) = S$) then
set session key $SK_{SM-S}$;
return complete;
else return terminate;
```

5 Security analysis

In Sect. 5.1, we provide probabilistic analysis to prove that the proposed scheme is MA-Security and AKA-Security under the condition of providing advantages for attackers by the CK adversary model. In Sect. 5.2, we prove that our scheme has sufficient security by theoretical analysis.

5.1 Provable security

Theorem 1: If the attacker A forges the pseudonym of *SMi*, *SIDi* or the pseudonym of *ESj*, *SIDj* and completes the mutual authentication process, it means that the attacker *A* can initiate an impersonation attack at the both sides to break the security of MA. Assuming that *A* knows the authentication process of the proposed scheme given a certain probability δ. After *A* obtains the ID of *ES* or *SM*, *A* will use this identity to join the network, thereby obtaining the authentication information to break the MA. The advantage of *A* breaking scheme $\sum$ is defined as $Adv_{\sum}^{A}(E_{MA}) = P_r[SID_j' = SID_j \cup SID_i' = SID_i]$. As long as *A* breaks the two-way authentication at either side, it can be regarded as breaking the MA-secure to the protocol. Under the premise of $Adv_{\sum}^{A}(E_{MA})$, the probability of *A* successfully destroying the MA-secure is calculated as (3):

$$\begin{aligned}\Pr[Ema] &= 1-\left(1-p_1\right)\left(1-p_2\right)\\ &= 1-\left\{1-\left[\left(\left(\frac{1}{q^2}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_a}\right)\left(\frac{1}{q_m}\right)\right)^{q_b}\right]^{q_{Hash}}\delta\right\}\\ &\left\{1-\left[\left(\frac{1}{q}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_m}\right)\left(\frac{1}{q_a}\right)\right]^{q_{Send}}\delta\right\}\end{aligned} \tag{3}$$

where $p1$, $p2$ represent the probability of destroying the MA-secure on the *ESj* side and the *SMi* side respectively. Where *qHash*, *qSend*, *q*, *qe*, *qh*, *qm*, *qa*, *qb* respectively represents the limit of Hash(m) queries, Send(P,m) queries, random number generation, modular exponentiation, hash function (Hash), multiplication (Multiplication), addition (Add) and bilinear pairing (BPA) operations of *A*.

Proof: We define the following events:

(1) E*Extract*: *A* obtains the correct pseudonym of an instance.
(2) E*am*: *A* cracks the encrypted information required in the two-way authentication process.
(3) E*sm-es*: *A* successfully destroys the authentication of the smart meter to the edge server.
(4) Ees-sm: *A* successfully destroys the authentication of the edge server to the smart meter.
(5) E*am*: *A* destroys the MA-secure in mutual authentication.

If *A* wants to destroy the MA on *ESj*, *A* needs to crack the encrypted information in the mutual authentication process under the premise of obtaining the pseudonym of *ESj*, and successfully destroys the authentication of on the ES side. The probability $p1$ of *A* destroying the MA on *ESj* is calculated as (4):

$$p1 = \Pr[E_{am} \wedge E_{es-sm} | E_{Extract}] \tag{4}$$

To analyze the process of destroying MA on *ESj*, define the following event:
1) E*ps-es*: *A* obtains the correct pseudonym of *ESj* successfully.

2) E*gx*: *A* uses the generated pseudonym to obtain encrypted information *gx* by a bilinear pairing operation.

3) E*Send*: When *gx* generated by *A* goes through the Send(*gx*) query, the resister sends back Y. The probability *p*1 of *A* destroys the MA at the ES side is also calculated as (5):

$$p1 \;=\; Pr[\mathrm{E}am \wedge \mathrm{E}es - sm|\mathrm{E}Extract] \;=\; Pr[\mathrm{E}ps - es \wedge \mathrm{E}gx|\mathrm{E}Send] \tag{5}$$

(1) A obtains the real *IDj* of *ESj* in the network, and disguises his identity as *ESj*"*Sf*'. *SMi* registered at *RA* tries to perform two-way authentication with the impersonation"*ES*".

(2) When *SMi* first generates the random number $x \in Z$, and produces the authentication request message as follows:$gx = g^x$, $X = xP$, $M = xPpub(H1(IDj)^{(-1)})$,$\mathrm{N} = \mathrm{H}_2\left(\mathrm{g}_\mathrm{x}\right) \oplus \left(\mathrm{ID}_\mathrm{i} \parallel \mathrm{R}_\mathrm{u} \parallel \mathrm{X} \parallel \mathrm{T}_\mathrm{sm}\right), \sigma = \mathrm{SID}_\mathrm{i} + \mathrm{xH}_2(\mathrm{ID}_\mathrm{i} \parallel \mathrm{R}_\mathrm{u} \parallel \mathrm{X} \parallel \mathrm{T}_\mathrm{sm})$, where *Ru* is sent back by the *RA* in the registration phase. After the calculation, *SMi* will send (*M*, *N*, *σ*) to *ESj*"*Sf*' to negotiate a session key.

(3) After *A* intercepts the authentication information sent by *SMi*, *A* needs to crack it. The most effective way to crack it is to calculate $gn_x = e(M, SID_j), H_2\left(gn_x\right) \oplus N$, and get $IDn_i \parallel R_u^n \parallel Xn \parallel T_{sm} = H_2\left(gn_x\right) \oplus N$ by forging the pseudonym *SIDj* of *ESj*. In order to improve the advantage of the attacker, we default that *A* only needs to crack $H_2\left(gn_x\right)$ launching an impersonation attack. *p*1 is calculated as (6):

$$p_1 = \left\{\left(\left(\frac{1}{q^2}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_a}\right)\left(\frac{1}{q_m}\right)\right)^{q_b}\right\}^{q_{Hash}}\delta \tag{6}$$

a. First, *A* has to challenge the private key s of *RA* and random number *Ri*. Because of *s*, *Ri*∈*Z*q*, the probability of *A* generates random number *s* and *Ri* is $\frac{1}{q^2}$. *SIDj* consists of three operations: hash operation, point addition and linear multiplication twice. So the probability of cracking *SIDj* is $\left(\frac{1}{q^2}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_a}\right)\left(\frac{1}{q_m}\right)$ .b. *A* needs to use the generated *SIDj* to calculate the bilinear pairing operation, which needs to be performed with the total number of times *qb* power.c. Then *A* needs to launch Hash(*gx*) query to determine whether the generated *gx* is correct. The probability of initiating the Hash operation successfully and passing the test correctly is *qHash* power.

If A wants to destroy the MA at *SMi*, *A* needs to crack the random number *Ri* under the premise of obtaining the *SIDi*. So, the probability *p*2 of *A* destroying the MA at *SMi* is calculated as (7):

$$p_2 = Pr[E_{am} \bigwedge E_{sm-es}|E_{Extract}] \tag{7}$$

We further define the following event:

(1) E*ps-sm*: *A* successfully obtains the correct *SIDi*.
(2) E*r*: *A* cracks the random number *Ri* on *SMi* generated by *RA*.
(3) E*Send*: *A* launches Send(*SMi*, Start) query with $SMi = SM_i^*$. The probability of *p*2 can be calculated as (8):

$$p_2 = Pr[E_{am} \bigwedge E_{sm-es} | E_{ps}] = Pr[E_{ps-sm} \bigwedge E_r \bigwedge E_{Send}] \tag{8}$$

1) *A* can launch an impersonation attack on *SMi* with the two conditions of obtaining *SIDi* and the random number *Ri*.

2) After *A* holds the conditions to forge *SIDi*, *A* launches Erode(*SMi*, *Ri*) query. When the resister *D* sends back the registered information, *A* has all the conditions to be able to initiate the impersonation attack on *SMi*.

3) *A* only needs to send the correct pseudonym *SIDi* and *Ri* to the resister *D*. So p2 is calculated as (9):

$$p_2 = \left\{ \left[(\frac{1}{q})(\frac{1}{q_h})(\frac{1}{q_m})(\frac{1}{q_a}) \right] \right\}^{q_{Send}} \delta \tag{9}$$

a. *SIDi* consists of three operations: hash operation, linear multiplication, and linear addition. In addition, *A* wants to generate a random number $Ri \in Z^*q$. So the probability of forging *SIDi* and $Ri \in Z^*q$ by *A* is is $(\frac{1}{q})(\frac{1}{q_h})(\frac{1}{q_m})(\frac{1}{q_a})$b. After *A* completes the generation of *SIDi*, there is a probability of $\frac{1}{q_s}$ to send the information successfully during the postback process.c. *A* needs to check the correctness of generating *SIDi* and *Ri*, *A* needs to launch Send(*SMi*, *Start*) query to resister *D*. The probability of initiating Send operation with $SMi = SM_i^*$ is *qSend* power.

So we can conclude the Eq. (3). Since the operations in (3) are large enough, even if the attacker *A* has such an advantage $\mathrm{Adv}_{\sum}^{A}(\mathrm{E_{MA}})$, the probability of *A* successfully destroys the MA is still negligible.

Theorem 2: If *A* completes the generation of σ or *S*, *A* can destroy the AKA secure. Let E*AKA* represent the event *A* successfully wins the game and $\sum$ represent an identity authentication scheme. *A* has an probability ξ that could directly destroy the MA on the public channel. The advantage of *A* breaking scheme $\sum$ is defined as $Adv_{\sum}^{A}(E_{AKA}) = P_r[L' = \sigma P \cup S' = S]$. Under the premise of $\mathrm{Adv}_{\sum}^{A}(\mathrm{E_{MA}})$, the probability of *A* successfully destroys the AKA secure is calculated as (10):

$$\begin{aligned} \Pr[Eaka] &= 1 - \left(1 - p_3\right)\left(1 - p_4\right) \\ &= 1 - \left\{ 1 - \left[\left(\frac{1}{q^2}\right)\left(\frac{1}{q_a}\right)\left(\frac{1}{q_m}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_a}\right)\left(\frac{1}{q}\right)\left(\frac{1}{q_m}\right)\left(\frac{1}{q_h}\right) \right]^{q_{Send}} \xi \right\} \\ &\quad \left\{ 1 - \left[\left(\frac{1}{q^2}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_h}\right) \right]^{q_{Send}} \xi \right\} \end{aligned} \tag{10}$$

where *p*3,*p*4 respectively represent the probability of *A* destroys AKA secure on *ESj* and *SMi*. Where *qSend*, *q*, *qe*, *qh*, *qm*, *qa*, *qb* respectively represents the limit of Send(P,m) queries, random number generate, modular exponentiation, hash function (Hash), multiplication (Multiplication), addition (Add) and bilinear pairing (BPA) operations of *A*.

Proof: We define the following events:

1) E*ma*:*A* successfully destroys the MA-secure of the protocol. ($Pr[Ema] = \xi \leq 1 - (1 - p1)(1 - p2)$).

2) E*aka*: *A* destroys AKA secure.

3) E*aka-es*: *A* successfully breaks the *SMi* to *ESj* authentication.

4) E*aka-sm*: *A* successfully breaks the *ESj* to *SMi* authentication.

To analyze the process of destroys AKA security on the edge server side with $\mathrm{Adv}_{\Sigma}^{\mathrm{ES}}(\mathrm{E}_{\mathrm{AKA}})$, we further define the these event:

1) E*Execute*(*L*): *A* obtains the authentication information *L* sent by the smart meter.

2) E*Send*: *A* successfully forges a legal login message *L'*.

The probability *p*3 of *A* destroys AKA security on *ESj* is calculated as (11):

$$p_3 = Pr[E_{aka-es}|E_{ma}] = Pr[E_{Execute(L)} \bigwedge E_{Send}] \tag{11}$$

Because *A* has an advantage $\mathrm{Adv}_{\Sigma}^{\mathrm{ES}}(\mathrm{E}_{\mathrm{AKA}})$, in order to destroy the consistency of the session key, *A* only needs to crack the random number *x* in the session key. *A* verifies the correctness of *x* by verifying the correctness of *Ln* = *σP* so that *A* could destroy the AKA security by forging session key. So *p*3 is calculated as in (12):

$$p_3 = [(\frac{1}{q^2})(\frac{1}{q_a})(\frac{1}{q_m})(\frac{1}{q_h})(\frac{1}{q_a})(\frac{1}{q})(\frac{1}{q_m})(\frac{1}{q_h})]^{q_{Send}}\xi \tag{12}$$

a. *A* generates a random number *x*, *Ru* with probability ($\frac{1}{\mathrm{q}^2}$), and the part of $XH_2(ID_i \parallel R_u \parallel X \parallel T_{sm})$ with probability ($\frac{1}{q_m}$)($\frac{1}{q_h}$).b. *A* needs to connect them by linear addition, and the probability of linear addition is($\frac{1}{q_a}$). *A* needs to launch Send(*ESj*,*L'*) query to resister *D* to check the correctness of *L'*. The probability of initiating Send operation with *L'*= *σP* is *qSend* power.

To analyze the process of *A* destroys AKA security on *SMi* with $\mathrm{Adv}_{\Sigma}^{\mathrm{SM}}(\mathrm{E}_{\mathrm{AKA}})$, we further define the following event:

1) E*Execute*(*S*): *A* obtains the authentication information *S* sent by *ESj*.

2) E*Send*: *A* successfully forges a legal login message.

Due to *A* has an advantage of $Adv_{\Sigma}^{SM}(E_{AKA})$, in order to destroy the AKA secure, *A* only needs to crack the key generation process. But to judge the correctness of the key, *A* needs to forge the correct *S* and verify the correctness of *S*. If *S* is correct, it means that *A* has obtained the session key for communication between the two parties. So *p*4 is calculated as in (13):

$$p_4 = [(\frac{1}{q^2})(\frac{1}{q_h})(\frac{1}{q_h})]^{q_{Send}} \xi \tag{13}$$

a. *A* generates the part of *xY* || *Tms* with probability $(\frac{1}{q^2})$.b. *A* generates the part of SK with probability $(\frac{1}{q^2})$.c. *A* generates the part of *S* with probability $(\frac{1}{q_h})$.

After generating *S'*, A checks the correctness of *S'* = *S* and the probability of this part is *qequ* power. *A* needs to launch Send(*SMi*, *S'*) query to resister *D*. The probability of initiating Send operation with *S'* = *S* is *qSend* power.

So we conclude the Eq. (10). Since these operations in (10) are large enough, even if *A* has such an advantage $\mathrm{Adv}_{\Sigma}^{A}(\mathrm{E_{AKA}})$, the probability of destroying AKA secure is still negligible.

5.2 Informal security analysis

In this subsection, we will conduct informal security analysis. The proposed PSLA scheme holds the following security properties.

1)*Mutual authentication*: The *SM* will authenticate the target *ES* by checking the correctness of *H1*(*SKSM* − *S*) = *S*. *ES* will verify the correctness of *L* = *σP* to confirm the identity of the *SM*. So the both sides of ES and the SM have completed a mutual authentication.

2) *Session key agreement*: *SM* and *ES* can maintain a good consistency with the negotiated session key *SK* = *SKS* − *SM* = *H3*(*yX* ||*TS*) = *SKSM* − *S* = *H3* (*xY* || *TS*) = *H3*(*xyP* || *TS*). Due to the existence of the CDH problem, it can be determined that the session key will not be obtained by any other participant or adversary.

3)*Perfect forward secrecy*: Assuming that if the session key negotiated between *SM* and *ES* has been leaked, the attacker *A* intercepts all the authentication information (*M*, *N*, *σ*, *Tsm*) and (*S*,*Y*,*TS*) on the channel including *SIDi* and *Y*. If the attacker wants to obtain the session key *SK* = *SKS* − *SM* = *H3*(*yX* || *TS*) = *SKSM* − *S* = *H3*(*xY* || *TS*) = *H3*(*xyP* || *TS*), he/she has to know the random number *x* or *y*. Due to the difficulty of cracking in the CDH assumption, the session key cannot be compromised without knowing the random number *x* or *y*. It is easy to conclude that the PSLA scheme can provide the perfect forward secrecy.

4)*Confidentiality of RA*: By the PSLA scheme, the *RA* does not participate the MA between *ES* and *SM*. Its major responsibility is to design a pseudonym for *ES* and *SM* in the registration over a secure channel, so it is impossible for an attacker to intercept the pseudonym information in this process.

5)*Edge server privacy information protection*: After receiving an authentication request, the *ES* extracts the user identity and calculates the session key.

During the authentication process, both parties use a pseudonym generated by the *RA*. The pseudonym cannot be easily disclosed because it is hidden as the encrypted data. At the same time, the process of generating pseudonyms by the *RA* is carried out on a secure channel, which provides an effective protection for the security and privacy of pseudonyms. Furthermore, the proposed scheme can resist against the following attacks.

1)*Identity forgery attacks on SM*: If attacker *A* wants to forge *SMi*, it must generate a legal authentication message (M, N, σ, Tsm). Among them, σ is actually the authentication information signed by the *SM*. Since the attacker cannot forge the pseudonym of the *SM*, the attacker cannot generate the correct encrypted information σ. Even if the attacker sends the encrypted information σ to the *ES*, the subsequent authentication will be interrupted due to $\mathrm{L} \neq \sigma\mathrm{P}$.

2) *Identity forgery attacks on ES*: In the authentication message (M, N, σ, Tsm), *SMi* uses the identity of the *ES* to encrypt (IDi, Ru, X). Without knowing the pseudonym of the *ES*, the attacker *A* cannot extract (IDi, Ru, X) from the authentication message. Since the attacker cannot achieve the forgery of the pseudonym of the *ESj*, the correct authentication information S cannot be generated and sent back to the *SM*. Therefore, the attacker cannot impersonate the *ES*.

3)*MITM attacks*: By the PSLA scheme, an *ES* can verify the true identity of a *SM* by checking the correctness of $\sigma P = L = (Ru + Ppub\ H0(IDi)) + X'H2(\mathrm{ID'i} \,\|\, R'u \,\|\, \mathrm{X'} \,\|\, Tsm)$. The *SM* can authenticate the *ES* by checking correctness of $H1(SKSM\text{-}S) = S$. Without knowing the pseudonym *SIDj* of the *ES* and the pseudonym *SIDi* of the *SM*, the adversary cannot generate a valid message. Therefore, the proposed PSLA scheme can resist MITM attacks.

4)*Replay attacks*: By a replay attack, an attacker sends data that has been received by a destination with the aim to deceive the system. Since a time stamp and random number pseudonym have been introduced in the PSLA scheme, freshness of the information in each data interaction can be ensured. Even if the attacker has tampered with the timestamp and sends the data packet, in the subsequent authentication process, the replay attack can be resisted by the random number used.

6 Performance evaluation

In this section, we comprehensively evaluate the performance of the proposed PSLA scheme and compare it with that of other four related schemes including the TPPA [12], the AESK [10], the PIAA [11] and the SPAK [7] schemes in terms of number of the cryptography operations, computation delay and communication cost. We

Table 2 Comparison of number of cryptography operations

	PSLA		TPPA		AESK		PIAA		SPAK	
	SM	ES	SM	ES	SM	ES	SM	ES	SM	ES
MUL	3	4	4	4	5	5	4	5	2	2
PAD	2	2	3	4	1	1	0	3	1	1
HAS	5	5	4	6	5	5	5	5	5	5
EXP	1	1	2	2	1	1	1	0	1	1
BPA	0	1	0	0	0	0	0	1	1	1

Table 3 Execution time of basic operations (ms)

	Description	Alibaba cloud	Google nexus
T_{BPA}	The execution time of a Bilinear Pairing (ECB mode and 512 bits)	5.275	48.66
T_{MUL}	The execution time of a Curve25519 Point Multiplication (160 bits)	1.97	19.919
T_{PAD}	The execution time of a Curve25519 Point Addition (160 bits)	0.012	0.118
T_{HAS}	The execution time of a SHA256 (160 bits)	0.009	0.089
T_{EXP}	The execution time of a Curve25519 Modular Exponentiation (160 bits)	0.339	3.328

assume that the bit length of EXP and timestamp is 64 bits, the bit length for random number authentication and Hash function is 160 bits, the operation for each point on the elliptic curve is 161 bits, and the operation for each element in the multiplicative group is 512 bits.

6.1 Number of cryptography operations

We divide the encryption operations into five categories. Among them, PAD, HAS, EXP, BPA and MUL represents point addition, hash operation, modular exponentiation, bilinear pairing and scalar multiplication, respectively. Without any attack, the number of specific encryption operations on the SM and the ES by five schemes is shown in Table 2.

It is clear to see in Table 2 that the proposed PSLA scheme uses fewer MULs and more EXPs. It is worth noting that the EXP can effectively cut down computation delay, while the MUL takes much more time than other operations. It is obvious that the proposed PSLA scheme can guarantee the security of the authentication while incurring a low time delay to meet the efficiency requirements.

6.2 Computation delay

Due to the large difference on computing power between the *SM* and the *ES*, time-consuming statistics has been conducted over different platforms for each party. The *ES* is simulated on the cloud platform provided by Alibaba,Intel(R) Xeon(R) CPU E5-26,300 @ 2.30 GHz, 1 GB RAM and Ubuntu 14.04 for 64bit operating system. The SM has been simulated over a 2 GHz ARM CPU Emulated on Google Nexus One smartphone with armeabi-v7a, 300 MB RAM and Android 4.4.2 operating system. We include the statistics time of the cryptography operations in Table 3 [41, 42].

The proposed PSLA scheme is compared with the TPPA, AESK, PIAA and SPAK schemes, those are much time-consuming on certification. According to the information in Tables 2 and 3, the number of specific encryption operations by the TPPA scheme is 29, which needs 95.968 ms. The number of specific encryption operations

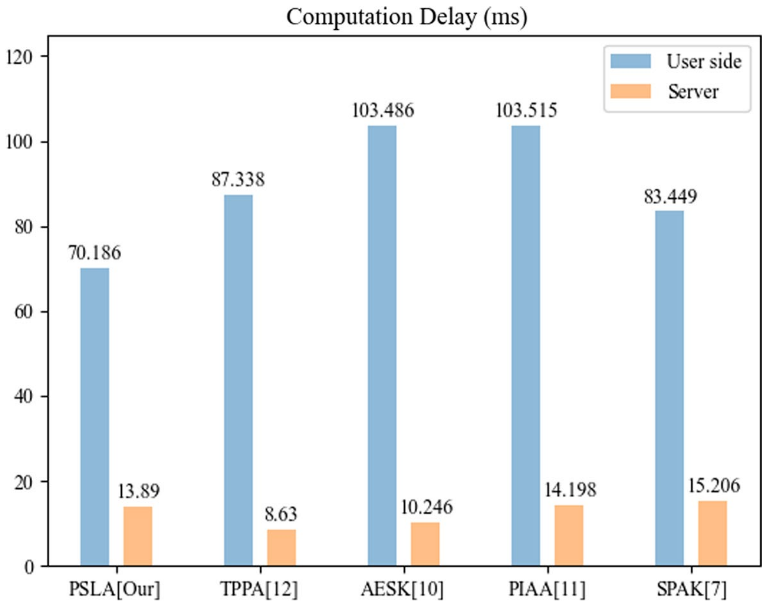


Fig. 6 Computation delay in authentication process

by the AESK scheme is 24, which needs 113.732 ms. The specific number of encryption operations by the PIAA scheme is 25, which needs 98.655 ms. The specific number of encryption operations by the SPAK scheme is 25, which needs 98.655 ms. The specific number of encryption operations by the SPAK scheme is 26, which needs 102.358 ms. The delay by the specific operations can be calculated by Eq. (14–18). The minimum calculation delay of the PSLA scheme is 77.329 ms, which is 19% lower than that of the TPPA scheme, 32% lower than that of the AESK scheme, 22% lower than that of the PIAA scheme, and 25% lower than that of the SPAK scheme.

$$T_{PSLA} = 4T^{ES}_{MUL} + 3T^{SM}_{MUL} + 2T^{ES}_{PAD} + 2T^{SM}_{PAD} + 5T^{ES}_{HAS} + 5T^{SM}_{HAS} + T^{ES}_{EXP} + T^{SM}_{EXP} + T^{ES}_{BPA} = 77.329(ms) \tag{14}$$

$$T_{TPPA} = 4T^{ES}_{MUL} + 4T^{SM}_{MUL} + 4T^{ES}_{PAD} + 3T^{SM}_{PAD} + 6T^{ES}_{HAS} + 4T^{SM}_{HAS} + 2T^{ES}_{EXP} + 2T^{SM}_{EXP} = 95.968(ms) \tag{15}$$

$$T_{AESK} = 5T^{ES}_{MUL} + 5T^{SM}_{MUL} + T^{ES}_{PAD} + T^{SM}_{PAD} + 5T^{ES}_{HAS} + 5T^{SM}_{HAS} + T^{ES}_{EXP} + T^{SM}_{EXP} = 113.732(ms) \tag{16}$$

$$T_{PIAA} = 5T^{ES}_{MUL} + 4T^{SM}_{MUL} + 3T^{ES}_{PAD} + 5T^{ES}_{HAS} + 5T^{SM}_{HAS} + T^{ES}_{EXP} + T^{ES}_{BPA} = 98.655(ms) \tag{17}$$

$$\begin{aligned} T_{SPAK} = {} & 2T^{ES}_{MUL} + 2T^{SM}_{MUL} + 3T^{ES}_{PAD} + 3T^{SM}_{PAD} + 6T^{ES}_{HAS} \\ & + 6T^{SM}_{HAS} + T^{ES}_{EXP} + T^{SM}_{EXP} + T^{ES}_{BPA} + T^{SM}_{BPA} = 102.358(ms) \end{aligned} \tag{18}$$

Figure 6 shows the computation delay comparison of the five schemes in the form of a histogram. The computation delay at the smart metre side by our proposed PSLA, the TPPA, the AESK, the PIAA and the SPAK schemes is 63.766 ms, 87.338 ms, 103.486 ms, 83.449 ms and 102.358 ms respectively. And the computation delay on the server side at the PSLA, the TPPA, the AESK, the PIAA and the SPAK schemes is 13.563 ms, 8.63 ms, 10.246 ms 15.206 ms and 9.664 ms respectively. It is clear that the computation delay by our proposed PSLA scheme is the shortest on the smart metre side.

Since the our proposed PSLA solution can resist some typical attacks, which have been verified by the security analysis and named as known attacks, the mutual authentication could not be interrupted by those known attacks. However, the authentication process can be interrupted by some new types of malicious attacks. Since the emergence of new malicious attacks is unpredictable, these potential attacks are considered as unknown attacks. Assume that during an authentication process, the mutual authentication process can be interrupted by unknown attacks. In general, if an authentication process is attacked by the known attacks, the computation delay to complete a mutual authentication is fixed. But, if the authentication process is attacked by the unknown attacks, the delay to complete the mutual authentication will be uncertain. We simulate the authentication process under the 2 different types of the attacks by using C++ coding. At each stage of an mutual authentication process, the authentication process by five authentication schemes could be interrupted due to unknown attacks.

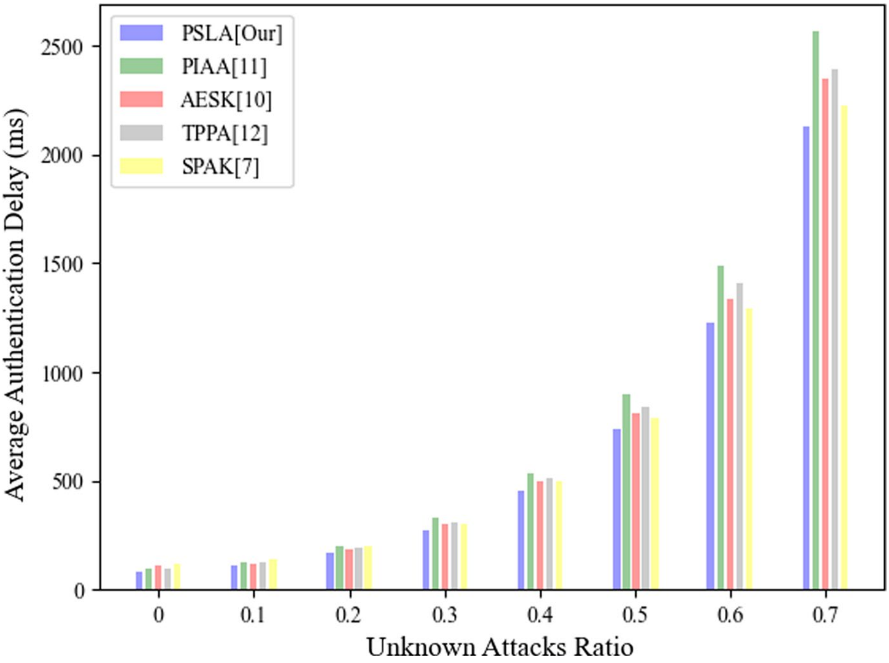


Fig. 7 Average authentication delay under different attack ratio

Table 4 Communication Cost for Different Schemes

Scheme	Communication overhead (bits)	Transmission delay (μs)	Propagatio delay (μs)	Total delay (μs)
PSLA	868	8.68	2	10.68
TPPA	966	9 66	2 5	12 16
AESK	801	8.01	3	11.01
PIAA	1156	11.56	2	12.56
SPAK	962	9.62	4	13.62

In the simulation experiments, we constantly change the ratio of unknown attacks and compare the computational delay by the five different schemes. Each authentication process has been simulated to run a total of number of 10,000 times by the five authentication schemes to analyze their performance in terms of authentication delay by constantly changing the ratio of attack types. In the simulation, when the ratio of unknown attacks to known attacks changes, the computation delay are measured for each scheme. The parameter that measures performance is the average successful authentication delay defined as in (19):

$$\mathrm{delay}_{\mathrm{AVG}} = \frac{\mathrm{delay}_{\mathrm{unknown}} * \mathrm{times}_{\mathrm{unknown}} + \mathrm{delay}_{\mathrm{known}} * \mathrm{times}_{\mathrm{known}}}{\mathrm{times}_{\mathrm{known}}} \tag{19}$$

where *timesunknown* = *ratiounknown*·timesALL-ATTACK

The specific simulation results are shown in Fig. 7, in which the abscissa represents the ratio of unknown attacks, and the ordinate represents the average authentication incurred by the five schemes. According to the results in Fig. 7, the proposed PSLA scheme can have the lowest delay when the ratio of unknown attacks increase from 0 to 0.7. The average authentication delay during the authentication process can also remain relatively stable as the ratio of the unknown attacks increases. Therefore, the proposed PSLA scheme has its efficiency advantages even under different unknown attacks.

6.3 Communication cost

There are two parts for the communication cost including transmission delay and propagation delay. For the transmission delay *Ttran*, as the communication between *SM* and *ES* uses WLAN connection, the experienced data rate of downlink and uplink in the urban area is 100 Mbps. For the propagation delay *Tprop*, the wave propagation speed is approximately equal to $3{\cdot}10^8$ m/s in wireless communication. It is assumed that the radius of a cell is 150 m, and the signal sent by the SMi will travel 150 m at the speed of $3{\cdot}10^8$ m/s to arrive at a *ESj*. So, the propagation delay from the *SMi* to the *ESj* is 0.5 μs. And the propagation delay from the *ESj* to the *SMi* is also 0.5 μs. The theoretical communication overhead of the proposed PSLA, the TPPA, the AESK, the PIAA and the SPAK schemes are respectively 868 bits, 966 bits, 801 bits, 1092 bits, 1156 bits and 962 bits. The theoretical communication cost is compared in Table 4.

From the Table 4, we can conclude that the total communication cost of the proposed PSLA scheme is 12.2% lower than that of the TPPA scheme, 3.0% lower than that of the AESK scheme, 15.0% lower than that of the PIAA scheme, and 21.6% lower than that of the SPAK scheme. In summary, the proposed PSLA scheme still has a good advantage in terms of communication cost.

7 Conclusion

In this paper, we have proposed a mutual anonymous authentication scheme for the communications in the SG with a key management function based on the ECC. The proposed PSLA scheme applies a pseudonym to ensure the greatest degree of protection against impersonation attacks in the authentication process for secure communications. The security of the PSLA scheme has been qualitatively analyzed to show its advantages. The proposed PSLA scheme can provide session key agreement, confidentiality of the *RA*, perfect forward secrecy and the *ES* privacy protection. The performance of the proposed PSLA scheme has been evaluated and compared with other existing solutions to conclude that the proposed PSLA scheme cannot only incur a low computation delay but also can realize all the security functions provided by other schemes. In future research, we will introduce a weighted evaluation pseudonym based on trust to realize the secure storage control of distributed trust data, and design a comprehensive trust model.

Funding Open Access funding provided by the Qatar National Library.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Lingjuan L, Karthik N et al (2018) PPEA: privacy preserving fog-enabled aggregation in smart grid. IEEE Trans Industr Inf 14(8):3733–3744
2. GaneshKumar P, Durgadevi V et al (2017) Fuzzy-based trusted routing to mitigate packet dropping attack between data aggregation points in smart grid communication network. Computing 99(1):81–106
3. Mohammadali A, Haghighi MS (2021) A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid. IEEE Trans Smart Grid 12(6):5212–5220
4. Chen G, He M, Gao J, et al (2021) Blockchain-based cyber security and advanced distribution in smart grid. In: 2021 IEEE 4th international conference on electronics technology (ICET), pp 1077–1080
5. Kumar P, Gurtov A, Sain M et al (2019) Lightweight authentication and key agreement for smart metering in smart energy networks. IEEE Trans Smart Grid 10(4):4349–4359
6. Khan A, Kumar V, Ahmad M, et al (2020) PALK: password-based anonymous lightweight key agreement framework for smart grid. Int J Electr Power Energy Syst 121
7. Xiang X, Cao J (2022) An efficient authenticated key agreement scheme supporting privacy-preservation for smart grid communication. Electric Power Syst Res 203:107–630
8. Deng L, Gao R (2021) Certificateless two-party authenticated key agreement scheme for smart gridsciencedirect. Inf Sci 543:143–156
9. Kumar P, Lin Y, Bai G et al (2019) Smart grid metering networks: a survey on security, privacy and open research issues. IEEE Commun Surv Tutor 21(3):2886–2927
10. Dariush A-M, Morteza NN et al (2018) An anonymous ecc-based self-certified key distribution scheme for the smart grid. IEEE Trans Ind Electron 65(10):7996–8004
11. Jia X, He D, Kumar N et al (2020) A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. IEEE Syst J 14(1):560–571
12. Qi M, Chen J (2021) Two-pass privacy preserving authenticated key agreement scheme for smart grid. IEEE Syst J 15(3):3201–3207
13. Mohammadali A, Haghighi MS, Tadayon MH et al (2018) A novel identity-based key establishment method for advanced metering infrastructure in smart grid. IEEE Trans Smart Grid 9(14):2834–2842
14. Barreto R, Faria P, Vale Z (2022) Electric mobility: an overview of the main aspects related to the smart grid. Electronics 11
15. Je SM, Woo H, Choi J, et al (2022) A research trend on anonymous signature and authentication methods for privacy invasion preventability on smart grid and power plant environments. Energies 15
16. Yang B, Xu G, Zeng X, et al (2018) A lightweight anonymous mobile user authentication scheme for smart grid. In: 2018 IEEE SmartWorld, ubiquitous intelligence and computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp 821–827
17. Odelu V, Das A, Wazid M et al (2018) Provably secure authenticated key agreement scheme for smart grid. IEEE Trans Smart Grid 9(3):1900–1910
18. Kaur K, Garg S, Kaddoum G, et al (2019) A secure, lightweight, and privacy-preserving authentication scheme for v2g connections in smart grid. In: international workshop on hot topics in social and mobile connected smart objects (HotSALSA'19) in conjunction with IEEE INFOCOM 2019 April 29–May 2, 2019, pp 541–546
19. Abbasinezhad-Mood D, Ostad-Sharif A, Nikooghadam M (2019) Novel anonymous key establishment protocol for isolated smart meters. IEEE Trans Ind Electron 4(67):2844–2851
20. Chen Y, Mart´ınez JF et al (2017) An anonymous authentication and key establish scheme for smart grid: fauth. Energies 10(9):1354
21. Ran C, Krawczyk H (2001) Analysis of key-exchange protocols and their use for building secure channels. Springer, Berlin, pp 453–474
22. Fouda MM, Kato N, Lu R et al (2017) a light-weight message authentication scheme for smart grid communications. IEEE Trans Smart Grid 2(4):675–685

23. Abdallah A, Shen XS (2018) A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. IEEE Trans Smart Grid 9(1):396–405
24. Gope P, Sikdar B (2019) An efficient data aggregation scheme for privacy -friendly dynamic pricing-based billing and demand-response management in smart grids. IEEE Internet Things J 14(6):1554–1566
25. Zhang H, Wang J, Ding Y (2019) Blockchain-based decentralized and secure keyless signature scheme for smart grid. Energy 108:955–967
26. Tsai JL, Lo NW (2016) Secure anonymous key distribution scheme for smart grid. IEEE Trans Smart Grid 7(2):906–914
27. Braeken A, Kumar P, Martin A (2018) Efficient and provably secure key agreement for modern smart metering communications. Energies 11(10):26–62
28. Xu G, Li X, Jiao L et al (2020) BAGKD: a batch authentication and group key distribution protocol for vanets. IEEE Commun Mag 58(7):35–41
29. Tahavori M, Moazami F (2020) Lightweight and secure puf-based authenticated key agreement scheme for smart grid. Peer-to-Peer Network Appl 13(5)
30. Prosanta G, Biplab S (2019) Privacy-aware authenticated key agreement scheme for secure smart grid communication. IEEE Trans Smart Grid 10(4):3953–3962
31. Kaveh M, MartiN D, Mosavi MR (2020) A lightweight authentication scheme for v2g communications: a PUF-based approach ensuring cyber/physical security and identity/location privacy. Electronics 9(9):1479
32. Mall P, Amin R, Das AK et al (2022) Puf-based authentication and key agreement protocols for iot, wsns, and smart grids: a comprehensive survey. IEEE Internet Things J 9(11):8205–8228
33. Sureshkumar V, Anandhi S, Amin R et al (2021) Design of robust mutual authentication and key establishment security protocol for cloud-enabled smart grid communication. IEEE Syst J 15(3):3565–3572
34. Baghestani SH, Moazami F, Tahavori M (2022) Lightweight authenticated key agreement for smart metering in smart grid. IEEE Syst J 16(3):4983–4991
35. Chaudhry SA (2021) PALK: password-based anonymous lightweight key agreement framework for smart grid. Int J Electr Power Energy Syst 125
36. Mahmood K, Chaudhry SA, Naqvi H et al (2017) An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. Fut Gener Comput Syst 81:557–565
37. Wazid M, Das A, Kumar JN, Rodrigues JJPC (2017) Secure three-factor user authentication scheme for renewable-energy-based smart grid environment. IEEE Trans Ind Inf 13(6):3144–3153
38. Badra M, Zeadally S (2017) Lightweight and efficient privacy-preserving data aggregation approach for the smart grid. Ad hoc Netw 64:32–40
39. Jo M, Jangirala S, Das AK et al (2021) Designing anonymous signature- based authenticated key exchange scheme for iot-enabled smart grid systems. IEEE Trans Ind Inf 17(7):4425–4436
40. Khan AA, Kumar V, Ahmad M, et al (2021) LAKAF: lightweight authentication and key agreement framework for smart grid network. J Syst Archit 116
41. OpenSSL (2017) Openssl, cryptography and ssl/tls toolkit. http://www.openssl.org
42. Arduino (2022) Arduinolibs: cryptographic library. http://rweather.github.io/arduinolibs/crypto.html

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.