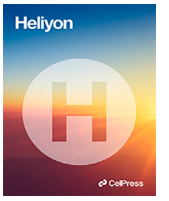
Heliyon 9 (2023) e17240

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A secure and efficient authenticated key exchange scheme for smart grid

Zhe Xia [a], Tao Liu [a], Jingjing Wang [b,*], Shi Chen [b,*]

[a] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
[b] State Grid Hubei Electric Power Co., Ltd., Information Communication Company, Wuhan, China

ARTICLE INFO

Keywords:
Key establishment
Secure data transmission
Smart grid

ABSTRACT

Smart grid provides convenience for power generation, consumption and distribution. Authenticated key exchange (AKE) is a fundamental technique to protect data transmission from interception and tampering in smart grid. However, since the smart meters only have limited resources in computation and communication, most of the existing AKE schemes are inefficient for smart grid. First, many schemes have to use large security parameters to compensate the loose reduction in their security proofs. Second, most of these schemes require at least three-round of communication to negotiate a secret session key with explicit key confirmation. To alleviate these issues, we propose a novel two-round AKE scheme with tight security for smart grid. Our proposed scheme integrates Diffie-Hellman key exchange and a tightly secure digital signature, in which not only mutual authentication can be realized but also the communicating parties can confirm that session keys are negotiated between them explicitly. Compared with the existing AKE schemes, the overheads in both communication and computation are lighter in our proposed scheme, because fewer rounds of communication are required and smaller security parameters can be used to achieve the same security level. Therefore, our scheme contributes to a more practical solution for secure key establishment in smart grid.

1. Introduction

Smart grid is widely considered as the development trend of power grid [1]. It is based on modern communication networks, integrating intelligent sensing technologies and novel decision support systems with the traditional power grid to realize effective power distribution from the power stations to the client terminals. In smart grid, the users can conduct two-way communications with the control center. For example, the users periodically upload their power consumption data to the control center, and it then sends commends to the smart meters based on the operational status. Through data processing and analyses, power generation and electricity distribution can be adjusted dynamically based on users' real-time demands for electricity, and this also helps to improve reliability of the grid system [2]. Nowadays, smart grid has become an important technique in the power industry, and it is crucial for the development of renewable energy in particular (as demonstrated in Fig. 1).

However, as a fundamental infrastructure in our modern society, smart grid is likely to be vulnerable to various security attacks in practice, both from cyberspace and physical worlds [3]. For example, the power consumption data is normally transmitted over public channels, where adversaries may eavesdrop, tamper and forge it. Nowadays, a broad consensus has reached that it is necessary to

* Corresponding authors.
E-mail addresses: xiazhe@whut.edu.cn (Z. Xia), Taoo.L@outlook.com (T. Liu), wangjingjing_china@163.com (J. Wang), chenshi1992m@163.com (S. Chen).

https://doi.org/10.1016/j.heliyon.2023.e17240

Available online 16 June 2023

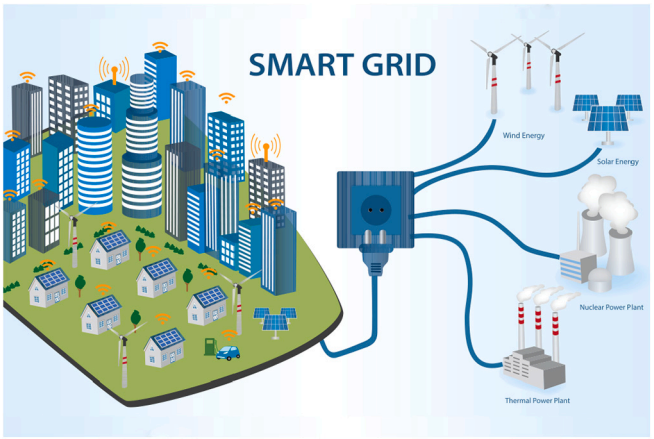


Fig. 1. A demonstration of the smart grid [2].

protection the power usage data transmitted in smart grid. To fulfill this requirement, the user and the control center should establish a secure channel between them, which further requires that a session key is negotiated between them with mutual authentication. They should also be able to confirm that this key is received by the other party but no one else, and this property is called *explicit key confirmation*. Afterwards, a secure channel can be constructed using this session key and some standard block cipher and message authentication code. And both privacy and integrity of the transmitted power usage data can be safeguarded by the secure channel. Therefore, secure and efficient key establishment is a crucial demand in smart grid.

Recently, many AKE schemes were introduced in the literature [4–7]. However, most of these schemes are either insecure or inefficient to be implemented directly in smart grid, since the smart meters only have limited resources in computation and communication. First, some schemes only achieve implicit key confirmation, where the user and the control center can only confirm that if anyone else obtains the session key, it must be the other party. Note that this property is weaker than its explicit counterpart, and the applications based on it may be vulnerable to denial of service (DoS) attacks.[1] Recall that smart grid has very strict demands on security, AKE schemes with explicit key confirmation is more desirable. Second, in many existing AKE schemes, their security proofs are conducted with loose reduction, e.g. via the Folking Lemma [25]. The consequence is that larger security parameters have to be used to compensate the security loss in the security reduction. However, cryptographic schemes initiated with larger security parameters will become less efficient in computation. Third, most of these schemes require at least three-round of communication to negotiate a secret session key with explicit key confirmation. Obviously, if an AKE scheme can be designed with fewer rounds of message exchange, it will become more efficient in communication.

1.1. Related works

We now briefly outline some existing works on authentication and key agreement for smart grid. In 2011, Zhou et al. [9] proposed a key exchange scheme based on Diffie-Hellman paradigm using Elliptic Curve Cryptography (ECC), and they suggested that their scheme can be deployed in smart grid. Zhou's scheme relies on public key infrastructure (PKI) and a trusted third party (TTP) is required. Afterwards, the work in [10] pointed out that the authentication phase in Zhou's scheme is vulnerable to the man-in-the-middle (MITM) attack, i.e. an adversary may impersonate to be a legitimate participant in the protocol. In 2012, Sule et al. [11] proposed an authenticated secure communication scheme in smart grid. This scheme not only provides user anonymity, but also resists insider attack and Denial-of-Service (DoS) attack. Xia et al. [10] proposed a lightweight authentication scheme with access control on directory. This scheme eliminates the high costs in PKI implementation and maintenance, but it requires a TTP to be involved. In 2015, Tasi et al. [12] proposed a key distribution scheme that achieves anonymous authentication for the smart meters and it reduces computational overheads compared with the related works. But their scheme is vulnerable to many security attacks, such as impersonation attack, MITM attack and DoS attack. Moreover, Odelu et al. [13] shown that the session key can be leaked in Tasi's protocol and the smart meters' anonymity could be violated in the Canetti-Krawczyk (CK) model [21]. To solve these problems, an improved protocol is proposed based on identity-based encryption (IBE) and ECC.

Afterwards, Mahmood et al. [14] designed an ECC-based authentication scheme. Their scheme reduces overheads in both computation and communication, and it can provide anonymity and privacy-preserving without employing a TTP. However, it was later pointed out [15] that Mahmood's scheme may suffer the key leakage attack. i.e. both the ephemeral key and the main key pair can be leaked in the CK model. Besides, it cannot achieve perfect forward secrecy, i.e. if the adversary can obtain user's long-term secret key, she can use it to decrypt the exchanged messages in the previous sessions. Although an improved protocol was proposed

[1] Note that in schemes with implicit key confirmation, when there are active attacks, one party maybe unaware that its counterpart has not received the key and still continue executing the following procedures. In comparison, such an attack can be discovered immediately in schemes with explicit key confirmation, and the scheme can be re-run promptly.

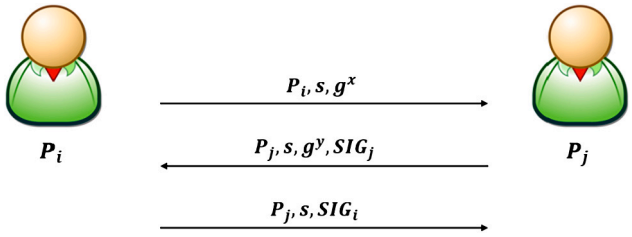


Fig. 2. SIG-DH protocol.

in [15], Chen et al. [16] illustrate that this modified protocol is vulnerable to the replay attacks, and they designed a pairing-based authentication scheme, simplifying the registration process and providing richer security features.

Moreover, Zhang et al. [17] designed an efficient authentication scheme with user anonymity. They also considered the limitation of storage and computing resources in smart meters, and replaced the expensive exponential operations by XOR operations to reduce computational overheads. Li et al. [18] proposed an anonymous authentication scheme in smart grid where smart meters' identities can be kept private from the control center. This protocol only employs some lightweight cryptographic primitives. However, Wu et al. [19] demonstrated later that Li's protocol is vulnerable to the DoS attack and it fails to provide mutual authentication. Furthermore, Kumar et al. [20] presented an authentication scheme for smart grid in which demand response is considered. This scheme also allows dynamic join or leaves of communication devices, which is more versatile in real-world applications. In [26], Srinivas et al. described an anonymous authentication and key agreement scheme for smart grid, where users' identities can be protected. The anonymity requirement is desirable in some circumstances, but it also brings additional overheads in computation. In [27], Gope et al. proposed a novel AKE scheme using reconfigurable PUFs to withstand the modeling attacks. In [28], Baghestani et al. pointed out that an existing ECC-based AKE scheme is not secure, and then proposed a modified one based on ECC. In [29], Deng et al. proposed a certificateless AKE scheme to avoid the certificate management problem. However, bilinear paring operation is required in the scheme and its computation costs are high.

1.2. Our contributions

To alleviate the above issues, we propose a novel authentication and key agreement scheme for smart grid. The main contributions are as follows:

- **Communication efficiency:** Smart gird always demands for low communication overheads to ensure the higher service quality. Our proposed scheme only requires two-round of communication to negotiate a secret session key with explicit key confirmation. Compared with the existing AKE schemes that require at least three-round of communication, our scheme has lower overheads in communication.
- **Computation efficiency:** Our scheme is analyzed in the CK model with tight security reduction. To achieve the same security level, one can use smaller security parameters to initialize the scheme. Therefore, it also enjoys computational advantages over the existing AKE schemes.
- **Resist to various attacks:** No TTP is employed in our scheme, and it can resist various security attacks, such as MITM attack, replay attack, impersonation attack, etc. Moreover, it provides forward secrecy so that even if the users' long-term secret key was leaked, the adversary still cannot decrypt the exchanged messages in the previous sessions.

1.3. Paper organization

This paper is organized as follows. In section 2, some cryptographic building blocks are outlined. In Section 3, the models and definitions are presented. A novel AKE scheme for smart grid is described in Section 4. Its security and performance analyses are given in Section 5 and 6, respectively. Finally, we conclude in Section 7.

2. Preliminaries

2.1. Signed Diffie-Hellman protocol

The signed Diffie-Hellman (SIG-DH) protocol is the integration of an authenticator and the unauthenticated Diffie-Hellman protocol. The SIG-DH protocol provides mutual authentication and key exchange for the communicating parties. Suppose a signed Diffie-Hellman protocol is executed between users A and B. A first sends $\upsilon = (g^a, \sigma_A)$, where σ_A is a signature generated using A's secret key, then B returns $w = (g^b, \sigma_B)$, where σ_B is a signature of g^b generated using B's secret key. Finally, the established session key is $k = g^{ab}$. In detail, the signed Diffie-Hellman protocol works as follows (also see Fig. 2).

- **Initialization:** Suppose p and q are large primes, satisfying $q|p-1$. g is a generator of $\mathbb{G}$ which a subgroup of $\mathbb{Z}_p^*$, and the order of $\mathbb{G}$ is q. Each participant generates a key pair and all the public keys are publicly known.

- **Step 1:** The initiator P_i randomly selects a value $x \in \mathbb{Z}_q$ and sends (P_i, s, g^x) to P_j, where s is the session-id.
- **Step 2:** Once receiving (P_i, s, g^x), the responder P_j randomly selects a value $y \in \mathbb{Z}_q$ and sends (P_j, s, g^y) to P_i. Moreover, the message is accompanied by a digital signature $SIG_j(P_j, s, g^y, g^x, P_i)$. Then, P_j computes the session key $k = g^{xy}$.
- **Step 3:** Once receiving (P_j, s, g^y) and the signature, P_i first validates the signature. If the verification is successful, P_i sends $(P_i, s, SIG_i(P_i, s, g^x, g^y, P_j))$ to P_j and computes the session key $k = g^{xy}$.
- **Step 4:** Once receiving (P_i, s, SIG_i), P_j validates the massage as well as the corresponding signature. If the verification is successful, P_j accepts the session key.

2.2. Tightly-secure digital signature

In security proofs by reduction, one can turn an adversary A who breaks the proposed cryptosystem into another adversary B who breaks some mathematical assumptions. If B's running time and success probability is similar as A's ones, the reduction is tight. Otherwise, if B's running time is much more than A's one or if B's success probability is much less than A's one, the reduction is loose.

In the literature, most signature schemes are not tight in security reduction. The consequence is that the security loss is linear or exponential to the number of signatures generated previously or the number of users in the multi-user setting. Although a few signature schemes can achieve tight security, many require expensive pairing operations. The first practical and tightly secure digital signature was introduced by Kristian et al. [22]. In their security reduction, the loss factor is reduced to a constant, and it is pairing free. Denote H_1 as a cryptographic hash function from the domain R to the range G. Kristian's scheme consists the following steps:

- **Key generation:** The prover randomly selects a value $a \in \mathbb{Z}_p$. It then computes $x = g^a$. Now, $sk = a$ is the signing key and $vk = x$ is the verification key.
- **Signing:** To sign the message m, the prover randomly samples $r \leftarrow R$ and computes $z = y^a$, where $y = H_1(r, m)$. Then, the prover generates a non-interactive zero knowledge (NIZK) proof $\pi_{eq} \leftarrow \mathsf{ZKPrv}_{eq}(a, x, y, z)$ to prove that $log_g x = log_y z$. The signature is $\sigma = (t, z, \pi_{eq})$.
- **Verification:** To validate the signature $\sigma = (r, z, \pi_{eq})$, the verifier first computes $y = \mathsf{H}_1(r, m)$ and verifies that π_{eq} holds for $log_g x = log_y z$ by checking whether $\mathsf{ZKVfy}_{eq}(\pi_{eq}, x, y, z) = 1$.

In practice, this scheme is very efficient compared with the previous digital signature schemes for two reasons. First, since B can simulate the zero-knowledge proof, there is no need to run the protocol multiple times to obtain a simulation, and this implies that the scheme is tightly secure and smaller security parameters can be used without downgrading its security. Second, it is pairing free and its computational costs are dominated by the exponential operations.

2.3. Tightly-secure AKE

Employing the above tightly secure digital signature, a two-pass AKE with explicit authentication has been proposed by Liu et al. [23] in 2020. Liu's scheme introduces a novel way to achieve explicit authentication without key confirmation. The main idea is to detect active attacks during the authentication phase. In comparison, implicit authentication can only detect active attacks after the authentication phase. Moreover, its security features can be justified in the Random Oracle (RO) model and the reduction is tight. Therefore, it is suitable for key establishment in the multi-user setting.

This scheme consists of a Diffie-Hellman key exchange and a tightly secure signature scheme. While two parties need to maintain a static counter for the states to prevent replay attacks. The status $st_i = \{sctr_{i,0}[j], sctr_{i,1}[j]\}$ of participant P_i is used to record the sequence number for the sessions, where both counters $sctr_{i,0}[j]$ and $sctr_{i,1}[j]$ are set to 0 initially. The counter $sctr_{i,0}[j]$ implies that the initiator is P_i and the responder is P_j, while $sctr_{i,1}[j]$ implies that the initiator is P_j and the responder is P_i. Liu's scheme works as shown in Fig. 3.

3. Models and definitions

3.1. System model

The system model, as shown in Fig. 4, consists of three types of entities: a control center (CC), some smart meters (SMs) and a TTP.

- **Smart Meter:** The SMs are intelligent devices installed at users' homes, collecting users' electricity consumption data. However, their computation and storage resources are normally limited. After the SM and the CC complete the authenticated key exchange protocol, a session key K is negotiated between them. Afterwards, the SM can send the power usage data to the CC securely at a fixed time interval.
- **Control Center:** The CC is responsible for collecting and analyzing users' electricity consumption data, controlling power generation and distribution. The CC is assumed to have sufficient computation and storage resources. After completing mutual authentication with the SM and obtaining the session key, the SM is allowed to upload the electricity consumption data, and the CC can send the feedback to the SM.

AKE.Setup(1^λ)

– The public parameters are generated as $pp_{SIG} \leftarrow SIG.Setup(1^\lambda)$, $pp_{KEM} \leftarrow KEM.Setup(1^\lambda)$. Return $pp_{AKE} := (pp_{SIG}, pp_{KEM})$.

AKE.Gen(pp_{AKE}, P_i)

– The participant P_i generates the key pair for the digital signature $(vk_i, sk_i) \leftarrow SIG.Gen(pp_{SIG})$, initializes the counter array $sctr_{i,0}[u] = 0$, $sctr_{i,1}[u] = 0$ and $st_i = \{sctr_{i,0}[u], sctr_{i,1}[u]\}$. Return $((vk_i, sk_i), st_i)$.

AKE.Protocol(P_i, P_j)

– P_i can access to $res_i = (sk_i, st_i, pp_{AKE}, PKList = \{vk_u\})$ and P_j can access to $res_j = (sk_j, st_j, pp_{AKE}, PKList = \{vk_u\})$. P_i invokes $(pk_{KEM}, sk_{KEM}) \leftarrow KEM.Gen(pp_{KEM})$ and increases its counter $sctr_{i,0}[j] = sctr_{i,0}[j] + 1$. Then, P_i uses sk_i to generate the signature σ_1 of message $m_1 = (P_i, P_j, sctr_{i,0}[j], pk_{KEM})$ and sends (m_1, σ_1) to P_j.
– P_j verifies σ_1 with vk_i, and checks whether its counter $sctr_{j,1}[i]$ is less than $sctr_{i,0}[j]$ that is contained in m_1. If the verification succeeds, P_j accept m_1 as a valid message; otherwise P_j rejects it. If m_1 is valid, P_j encapsulates the session key K via $(K, C) \leftarrow KEM.Encap(pk_{KEM})$ and synchronizes the counter $sctr_{j,1}[i] = sctr_{i,0}[j]$. Then, P_j generates the signature $\sigma_2 \leftarrow SIG.Sign(sk_j, m_1||m_2)$ and sends (m_2, σ_2) to P_i.
– P_i verifies whether σ_2 is a valid signature and checks the synchronization of its counter $sctr_{i,0}[j]$ with the counter $sctr_{j,1}[i]$ that is contained in m_2 (i.e., whether $sctr_{i,0}[j] = sctr_{j,1}[i]$). If the verification succeeds, P_i decapsulates the ciphertext C to obtain session key K.

Fig. 3. A tightly secure AKE scheme.

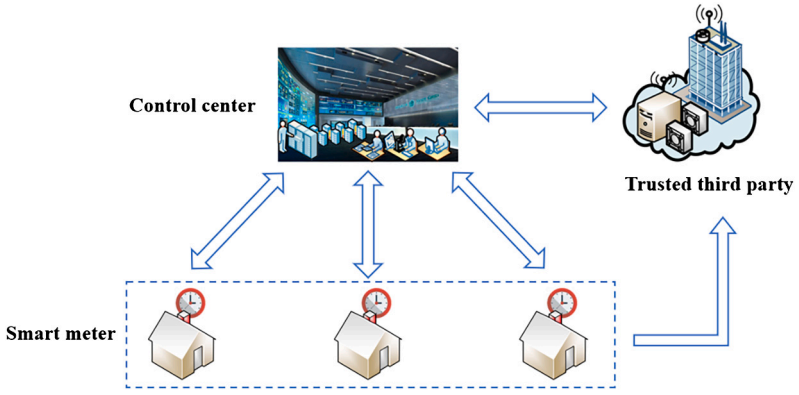


Fig. 4. System model.

– **Trusted Third Party:** The TTP is assumed to be trustworthy. It is responsible for the registration of the CC and SMs, as well as the generation of system parameters. After the system is initialized, the TTP can go offline.

3.2. Communication model

In the registration phase, the TTP communicates with the CC and SMs through a secure channel. During authentication, the SM and the CC communicate via a public channel. We assume that messages exchanged in the secure channel cannot be eavesdropped and tampered by the adversary. However, in public channels, the transmitted messages can be eavesdropped, tampered or fabricated. After the CC and the SM having authenticated each other's identity and negotiate a secret key, a secure channel is established where both privacy and integrity of the transmitted messages can be protected.

3.3. Adversary model

The CK model [21] is slightly modified to capture the adversary's abilities. Suppose P is a protocol and U is some participant within the protocol. There are two types of participants: the SM and the CC. Π_U^i denotes the i-th instance that U participates in. A's abilities are modeled via various oracle queries, which are described as follows.

- **Execute(A, i, B, j):** It models A's ability to passively eavesdrop on the protocol, i.e. the instances Π_A^i and Π_B^i are honestly executed, then the oracle returns the transcripts of this execution to the adversary.
- **Send(U_i, m):** It models A's ability to launch an active attack on the instance Π_U^i, i.e. A sends m to the oracle Π_U^i, then the oracle returns the response of instance Π_U^i to the adversary.
- **Reveal(U_i):** It models the circumstance that A obtains the session key. If the instance Π_U^i is accepted and a session key K_i is generated, the session key K_i is sent to A after this query.
- **Corrupt(U):** It models A's ability to corrupt some participants, i.e. A can obtain the participant U's secret states including her private key if U is corrupted.
- **Test(U_i):** It captures the session key's security. If the instance Π_U^i has accepted, then the challenger C flips a coin $b \in \{0,1\}$. If $b = 1$, C outputs the real session key, otherwise C randomly outputs a value in the key space.

Note that the **Test** query does not relate to the adversary's ability. Instead, it reflects whether A has learnt information of the session key. Moreover, one can launch the **Test** query only once, A needs to guess a bit b' for the coin b. If $b' = b$, A wins, and A's advantage is denoted as $Adv(A)$.

3.4. Security definitions

Definition 1 *(Freshness).* If the participant's instance Π_U^i is fresh, then the following conditions hold:

1. Π_U^i is accepted;
2. A has not initiated **Reveal** query to instance Π_U^i or its partner instance;
3. A has not initiated **Corrupt** query to participant U or its partner instance.

Definition 2 *(Partner).* The CK model allows the users to implement multiple sessions concurrently, capturing security in the parallel environment. If the participant's instances Π_A^i and Π_B^i are partners, and both of them accept the session-id sid, sid' and partner-id pid, pid', then the following conditions hold:

1. Both instances Π_A^i and Π_B^i are accepted, and the same session-id is output. It means that their transcripts in message exchange are matching.
2. Instance Π_A^i outputs $pid = B$ and instance Π_B^i outputs $pid' = A$. This means that they mutual accept after the authentication is completed.

Definition 3 *(SK-Secure).* When A issues a **Test** query on some fresh instances and guesses a value b'. A's advantage in the protocol P is denoted as:

$$Adv(A) = \left| Pr[b' = b] - 1/2 \right| \tag{1}$$

The protocol P is said to be SK-Secure (session key secure), if A's advantage $Adv(A)$ is negligible for any probabilistic polynomial time (PPT) adversary A.

Definition 4 *(Resist various other attacks).* The scheme can also resist some other security attacks, such as MITM attack, replay attack, impersonation attack and it provides forward secrecy.

4. The proposed scheme

Our scheme contains three algorithms: *system setup*, *registration* and *authenticated key agreement*. In the first phase, the system parameters are generated and broadcast. In the registration phase, both the SMs and the CC register with the TTP, and each will be assigned with a unique identity. Besides, the SMs and the CC generate their key pairs respectively. In the authenticated key agreement phase, the CC uses its private key to complete mutual authentication with each SM, and then generates a shared session key. The detailed processes are shown in Fig. 5.

4.1. System setup

Once the security parameter κ is determined, the TTP implements the following procedures:

- First, a cyclic group $G = \langle g \rangle$ with prime order q is generated;
- It selects two hash functions $H_0 : G \times G \to K$ and $H_1 : R \times \{0,1\}^\star \to G$, where K is the session key space and R is the space for random numbers;
- The TTP randomly selects a key pair, where the private key is $sk_T = x_T$ and the public key is $pk_T = g^{x_T}$. Finally, the system parameters $pp = (G, q, g, H_0, H_1)$ are made public.

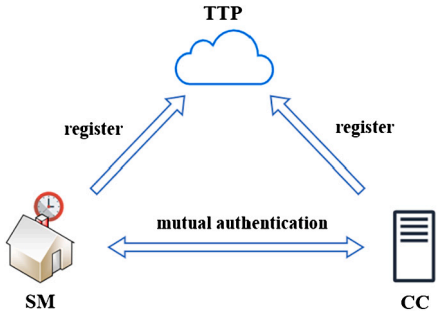


Fig. 5. Flow chart of system.

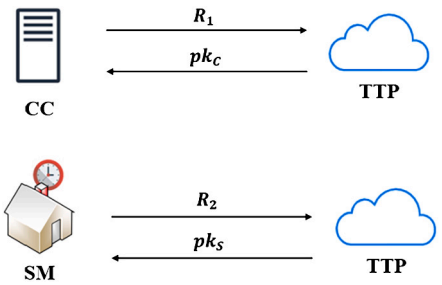


Fig. 6. The registration phase.

4.2. Registration phase

It is assumed that the registration phase is performed in a trustworthy environment, and exchanged messages are transmitted through a secure channel. The registration procedures for the SMs and the CC are similar. In this phase, the SMs and the CC send registration requests to the TTP, and then the TTP completes the registration phase. A demonstration is shown as in Fig. 6.

1. CC registration

- The CC selects its identifier ID_C. Then, it randomly chooses $x_C \in Z_q$ and calculates $R_1 = pk_t^{x_C} = g^{x_T x_C}$. When registering to the TTP for the first time, the CC transmits R_1 and ID_C to the TTP through the secure channel but keeps the private key x_C secret;
- When CC's registration request is received, the TTP calculates the public key for the CC as $pk_C = R^{1/x_T} = g^{x_C}$;
- TTP publishes the CC's public key.

2. SM registration

- The SM selects its identifier ID_S and randomly chooses $x_S \in Z_q$ as the private key. It calculates $R_2 = pk_t^{x_S} = g^{x_T x_S}$ and transmits R_2 and ID_S to the TTP through the secure channel;
- When SM's registration request is received, the TTP calculates the public key for this SM as $pk_S = R^{1/x_T} = g^{x_S}$;
- TTP publishes the SM's public key.

4.3. Authenticated key agreement phase

1. Key processing

- The SM randomly selects the private key $sk_S = (b, a)$, where $b \in \{0, 1\}$ and computes the public key $pk_S = (x_0, x_1)$, where $x_b = g^a, x_{1-b} \in G$. Then, the counter $ctr_S = 0$ is initialized;
- The CC randomly selects the private key $sk_C = (d, c)$, where $d \in \{0, 1\}$ and computes the public key $pk_C = (y_0, y_1)$, where $y_d = g^c, y_{1-d} \in G$. Then, the counter $ctr_C = 0$ is initialized.

2. Authentication

- The SM randomly selects $u_1, u_2 \in Z_q$, calculates $U_1 = g^{u_1}, U_2 = g^{u_2}$ and increases the value of counter ctr_S by one. The message m_1 is denoted as $m_1 = (SM, CC, ctr_S, U_1, U_2)$;
- The SM then randomly selects $t_S \in R$, calculates $y = H_1(t_S, m_1), z_b = y^a, z_{1-b} \in G$. It then generates the signature $\sigma_1 = (t_S, z_0, z_1, \pi)$, where the zero-knowledge proof π is $\pi \leftarrow \mathsf{ZPrv}(b, a; x_0, x_1; y, y, z_0, z_1)$. Finally, it sends (m_1, σ_1) to the CC.

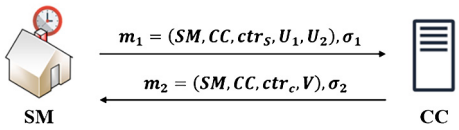


Fig. 7. Key Agreement Phase.

3. Session key encapsulation

- If $ctr_S > ctr_C$ and $\mathsf{ZVfy}(\pi, x_0, x_1, y, y, z_0, z_1) = 1$, the CC accepts the message (m_1, σ_1) and synchronizes the state $ctr_C = ctr_S$;
- The CC randomly selects $v \in Z_q$ and calculates $V = g^v$, then the CC encapsulates the session key as $K = H_0(U_1, U_2, V, U_1^v, U_2^v)$;
- The message m_2 is denoted as $m_2 = (SM, CC, ctr_C, V)$. The CC selects a random number $t_C \in R$, calculates $y' = H_1(t_C, m_1, m_2), z_d = y'^c, z_{1-d} \in G$, and then generates the signature $\sigma_2 = (t_s, z_0, z_1, \pi)$, where the zero-knowledge proof π is $\pi \leftarrow \mathsf{ZPrv}(c, d; y_0, y_1; y', y', z_0, z_1)$. Finally, the CC sends (m_2, σ_2) to the SM.

4. Session key decapsulation

- When receiving the message (m_2, σ_2), the SM checks whether $ctr_C = ctr_S$ and $\mathsf{ZVfy}(\pi, y_0, y_1, y, y, z_0, z_1) = 1$. If the check succeeds, the SM accepts the message (m_2, σ_2);
- The SM parses the message m_2 to obtained V and decapsulates the session key as $K = H_0(U_1, U_2, V, V^{u_1}, V^{u_2})$;
- Now, the SM and the CC have completed mutual authentication and a secure channel can is built using K and some block ciphers and message authentication codes. (See Fig. 7.)

5. Security analysis

Now, we show that the proposed scheme satisfies the desirable security properties, i.e. it is SK-secure and it can resist various security attacks.

5.1. Correctness

Correctness implies that if all participants act up to the protocol honestly, correct outputs will be delivered by the protocol. The correctness of our proposed scheme is straightforward and the inequality $ctr_U \leq ctr_S$ always holds. This is because:

1. When the SM initiates a session, it always increases the counter by one. After the CC receives a valid message, it will synchronize its counter ctr_C, and we have $ctr_S = ctr_C$.
2. If the CC receives an invalid message, the counter ctr_C remains unchanged and $ctr_S > ctr_C$.

Therefore, $ctr_U \leq ctr_S$ always holds if all participants follow the protocol. The correctness of the session key is obvious. When both messages m_1 and m_2 are valid, the SM and the CC will obtain the same session key K.

$$K = H_0(U_1, U_2, V, U_1^v, U_2^v) = H_0(g^{u_1}, g^{u_2}, g^v, g^{u_1 v}, g^{u_2 v}) = H_0(U_1, U_2, V, V^{u_1}, V^{u_2})$$

5.2. SK-secure

Theorem 1. *Our proposed scheme is SK-secure assuming that the decisional Diffie-Hellman (DDH) assumption holds.*

Proof. To prove the above claim, one needs to prove that if there is a PPT adversary A who wins the SK-secure game with advantage $Adv(A)$, then one can employ A to generate a PPT adversary B that can violate the DDH assumption with the same advantage. Assume A is an adversary who wins the above game with an advantage ϵ which is non-negligible. We can generate a challenger B that uses A's ability to distinguish a DDH instance (g^x, g^y, g^{xy}) from a random one. Moreover, the hash function is modeled as a RO. B needs to maintain a hash list that is initialized to be empty. In the reduction game, B first initializes the authentication scheme, generates the public parameters as well as the key pairs. Then, all public keys and public parameters are made public. Challenger B needs to maintain a participants list to record the participants' public key and private key, in order to respond to Corrupt, Execute, Reveal, Test queries from the adversary A. When A issues the Execute query, B simulates a complete run of the protocol and sends the transcript to A. When A issues the Corrupt query, B sends the internal status of the corrupted party to A. When A issues the Reveal query, B responses with the session key. For a fresh instance, the challenger B embeds the DDH instance in the authentication scheme as follows.

- Challenger B gets the DDH instance (g^x, g^y, g^z)
- Challenger B randomly selects $u_1 \in Z_q$ and sets $U_1 = g^{u_1}, U_2 = g^x, V = g^y$
- Challenger B computes the session key $K = H(U_1, U_2, V, V^{u_1}, g^z)$

Table 1
Execution time of some basic cryptographic operations.

Symbol	Description	Time cost (ms)
T_{pm256}	Point multiplication in 256-bit curve	1.2205
T_{pm512}	Point multiplication in 512-bit curve	9.2541
T_p	Bilinear mapping in 512-bit curve	39.2713

When A launches the **Test** query, B flips a coin b. If $b = 1$, the session key K is returned. Otherwise, a random value is return. A will output b' after **Test** query. If $b = b'$, the adversary B claims that the tuple (g^x, g^y, g^z) is a DDH tuple. Otherwise, it claims that the tuple is a random. Recall that the session key is $K = H(g^{u_1}, g^{u_2}, g^v, g^{u_1 v}, g^{u_2 v})$. If $g^z = g^{xy}$, A will output $b' = 1$ with chance $1/2 + \epsilon$. If $g^z \neq g^{xy}$, A will output $b' = 1$ with chance exactly $1/2$. Hence, B's advantage in the SK-secure game is ϵ:

$$|Pr[B(g^x, g^y, g^{xy}) = 1] - Pr[B(g^x, g^y, g^z) = 1]| = |Pr[Asucc] - 1/2| = \epsilon$$

Because it is assumed that the probability of breaking the DDH assumption is negligible, A's advantage in the SK-secure game is also negligible. □

5.3. Secure against various other attacks

Theorem 2. *Our scheme is also secure against various other attacks, such as MITM attack, replay attack, impersonation attack and it satisfies perfect forward secrecy.*

1. Replay attack: In our proposed scheme, the adversary is assumed to be capable of eavesdropping all messages transmitted between the SM and the CC. But the adversary cannot replay these messages, since both parties maintain a counter as their state. When the counter value is inconsistent, the message will be rejected. Therefore, our scheme does not suffer the replay attack.

2. MITM attack: Since digital signatures are used to authenticate the exchanged messages, the adversary cannot forge the signature since it does not know the private keys. Therefore, our scheme is secure against the MITM attack.

3. Impersonation attack: To impersonate to be a legitimate SM, A must either launch a replay attack or fabricate a signature for some message. However, since A cannot launch the replay attack and A cannot obtain the SM's private key, it cannot generate a valid message that will be accepted by the CC. The same argument also applies when the adversary intends to impersonate as the CC.

4. Forward secrecy: This property requires that even if user's long-term private keys were leaked, the session keys established in the previous sessions still can be kept private. In our scheme, the session keys are computed using the ephemeral keys through the Diffie-Hellman paradigm, and the long-term private key is only used to sign messages. If the ephemeral keys are properly erased after each session, the adversary cannot obtain the session keys in the previous sessions even if the private key is leaked. Therefore, our scheme satisfies forward secrecy.

6. Efficiency analyses

6.1. Computational efficiency

First, we analyze the SM's computational time. Suppose the security level is 128-bit in the experiment. An elliptic curve is initialized, where $y^2 = x^3 + ax + b \bmod p$ and p is a 256-bit prime. In comparison, the loss in security reduction is quadratic to the number of users in traditional AKE schemes, and these schemes need to use larger elliptic curves. As noted in [22], the loss in security reduction is quadratic to the number of users and the number of sessions in traditional digital signature schemes. For example, 2^{32} users running 2^{32} sessions each need to use elliptic curves with 512-bit to achieve 128-bit security level.

We implement the experiment using a Samsung Galaxy S5 smartphone with the quad-core processor, 2.45 GHz clock frequency, 4 GB random access memory and Android 6.0 operation system. The average costs of some basic cryptographic operations are listed in Table 1, and they are used to evaluate the execution time across different AKE schemes. The computational costs of hash operation and point addition operation are ignored because their costs are relatively small.

For convenience of comparison, we initialize all schemes in the elliptic curve group. The SM needs to perform 4 operations of point-multiplication and generates 1 signature, while the CC needs to perform 3 operations of point-multiplication and generates 1 signature. The comparison between our scheme and some related schemes [24,8] is shown in Table 2. Moreover, these schemes are implemented multiple times and the results can be found in Fig. 8.

Table 2
Comparison of computation costs.

	SM	CC	Total
He et al. [24]	$4T_{pm512}$	$6T_{pm512}$	92.541 ms
Gu et al. [8]	$6T_{pm512}$	$5T_{pm512} + T_p$	141.066 ms
Our	$19T_{pm256}$	$18T_{pm256}$	45.159 ms

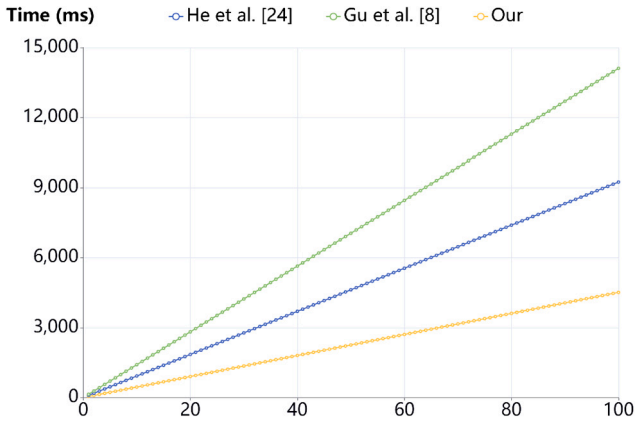


Fig. 8. Comparison of computation costs.

Table 3
Comparison of communication costs.

	Number	Size	Rounds
He et al. [24]	5	2560	3
Gu et al. [8]	8	4096	3
Our scheme	11	2816	2

6.2. Communication efficiency

The communication costs are evaluated from two aspects: the size of data that is transmitted in the protocol and the rounds of communication required between the SM and the CC. In the first aspect, since our scheme can work in 256-bit elliptic curve group and the existing AKE schemes need to work in 512-bit elliptic curve group, the size of group elements is smaller in our proposed scheme. In the second aspect, the existing AKE schemes require at least three rounds of communication to achieve explicit key confirmation, while our scheme can provide explicit key confirmation in two rounds. An illustration of the comparison is shown as in Table 3. Moreover, our proposed scheme and two related schemes [24,8] are run multiple times and the experimental results can be found in Fig. 9. Based on the analyses, although our scheme needs to transmit a larger number of group elements, the total amount of data transmitted is less than the related schemes. Moreover, our scheme is more efficient in rounds of communication. Therefore, it enjoys lower latency in communication.

7. Conclusion

In this paper, we proposed a secure and efficient authenticated key exchange scheme for smart grid. First, its security can be proved with tight reduction. Hence smaller security parameters can be used to achieve the same level of security, and the benefit is that the computational overheads can be reduced. Second, our proposed scheme can achieve explicit key confirmation in two rounds which is more efficient compared with the existing AKE schemes. Both of these features are particularly attractive for smart meters with limited resources. Hence, our work contributes to a more practical solution for secure key establishment in smart grid.

In the future, we plan to investigate two possible improvements of the proposed scheme. First, how multiple smart meters can authenticate them to the control center simultaneously and negotiate a secure group session key. The purpose is to further improve efficiency in computation and communication using the group-oriented design. Second, our scheme is only proved in the RO model. Although schemes designed in this model are generally more efficient, it is also desirable to explore how AKE schemes with tight security in the standard model can be designed. Obviously, such a scheme needs a tightly secure digital signature in the standard model.

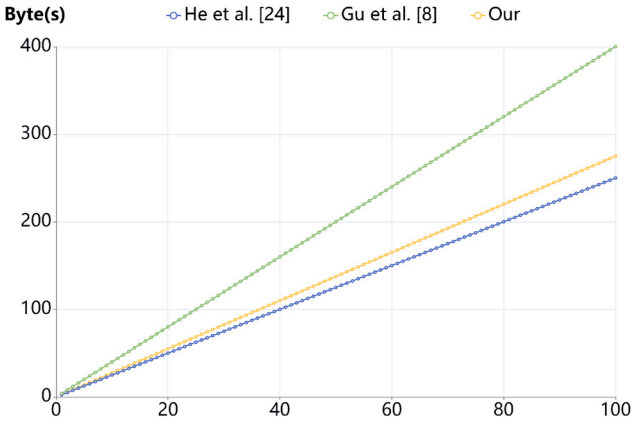


Fig. 9. Comparison of communication costs.

Author contribution statement

Zhe Xia: conceived and designed the experiments; analyzed and interpreted the data.
Tao Liu: conceived and designed the experiments; performed the experiments.
Jingjing Wang: performed the experiments; wrote the paper.
Shi Chen: contributed reagents, materials, analysis tools or data; wrote the paper.

Declaration of competing interest

The authors declare no conflict of interest.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work was partially funded by the Key Research and Development Program of Hubei Province (Grant No. 2022BAA050).

References

[1] Gurkan Solmaz, et al., Toward understanding crowd mobility in smart cities through the Internet of things, IEEE Commun. Mag. 57 (4) (2019) 40–46.
[2] Rich Castagna, et al., The smart grid and renewable energy, in: IEEE Innovation at Work, 2020.
[3] Ilhami Colak, et al., A survey on the critical issues in smart grid technologies, Renew. Sustain. Energy Rev. 54 (2016) 396–405.
[4] Piotr Szczechowiak, Martin Collier, Practical identity-based key agreement for secure communication in sensor networks, in: 2009 Proceedings of 18th International Conference on Computer Communications and Networks, IEEE, 2009.
[5] Chao-Liang Liu, et al., Ephemeral-secret-leakage secure ID-based three-party authenticated key agreement protocol for mobile distributed computing environments, Symmetry 10 (4) (2018) 84.
[6] Mihir Bellare, Bennet Yee, Forward-security in private-key cryptography, in: Cryptographers' Track at the RSA Conference, Springer, Berlin, Heidelberg, 2003.
[7] Christoph Bader, et al., Tightly-secure authenticated key exchange, in: TCC'2015, 2015, pp. 629–658.
[8] Wei Gu, Jian Shen, Yongjun Ren, Secure key-sharing algorithm based on smart grid, Chin. J. Netw. Inf. Secur. 7 (4) (2021) 141–146.
[9] Dapeng Wu, Chi Zhou, Fault-tolerant and scalable key management for smart grid, IEEE Trans. Smart Grid 2 (2) (2011) 375–381.
[10] Jinyue Xia, Yongge Wang, Secure key distribution for the smart grid, IEEE Trans. Smart Grid 3 (3) (2012) 1437–1443.
[11] Rucha Sule, Raj S. Katti, Rajesh G. Kavasseri, A variable length fast message authentication code for secure communication in smart grids, in: 2012 IEEE Power and Energy Society General Meeting, IEEE, 2012.
[12] Jia-Lun Tsai, Nai-Wei Lo, Secure anonymous key distribution scheme for smart grid, IEEE Trans. Smart Grid 7 (2) (2015) 906–914.
[13] Vanga Odelu, et al., Provably secure authenticated key agreement scheme for smart grid, IEEE Trans. Smart Grid 9 (3) (2016) 1900–1910.
[14] Khalid Mahmood, et al., An elliptic curve cryptography based lightweight authentication scheme for smart grid communication, Future Gener. Comput. Syst. 81 (2018) 557–565.
[15] Dariush Abbasinezhad-Mood, Morteza Nikooghadam, Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications, Future Gener. Comput. Syst. 84 (2018) 47–57.
[16] Yuwen Chen, et al., A bilinear map pairing based authentication scheme for smart grid communications: pauth, IEEE Access 7 (2019) 22633–22643.
[17] Liping Zhang, et al., A lightweight authentication scheme with privacy protection for smart grid communications, Future Gener. Comput. Syst. 100 (2019) 770–778.
[18] Xiong Li, et al., A provably secure and anonymous message authentication scheme for smart grids, J. Parallel Distrib. Comput. 132 (2019) 242–249.
[19] Libing Wu, et al., Anonymous and efficient message authentication scheme for smart grid, Secur. Commun. Netw. (2019) 2019.
[20] Neeraj Kumar, et al., ECCAuth: a secure authentication protocol for demand response management in a smart grid system, IEEE Trans. Ind. Inform. 15 (12) (2019) 6572–6582.

[21] Canetti Ran, Hugo Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: Eurocrypt'2001, 2001, pp. 453–474.
[22] Kristian Gjøsteen, Tibor Jager, Practical and tightly-secure digital signatures and authenticated key exchange, in: Crypto'2018, 2018, pp. 95–125.
[23] Xiangyu Liu, et al., Two-pass authenticated key exchange with explicit authentication and tight security, in: Asiacrypt'2020, 2020, pp. 664–673.
[24] Debiao He, et al., Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography, IET Commun. 10 (14) (2016) 1795–1802.
[25] David Pointcheval, Stern Jacques, Security proofs for signature schemes, in: Eurocrypt'1996, 1996, pp. 387–398.
[26] Jangirala Srinivas, et al., Designing anonymous signature-based authenticated key exchange scheme for Internet-of-Things enabled smart grid systems, IEEE Trans. Ind. Inform. 17 (7) (2020) 4425–4436.
[27] Prosanta Gope, Biplab Sikdar, A privacy aware reconfigurable authenticated key exchange scheme for secure communication in smart grids, IEEE Trans. Smart Grid 12 (6) (2021) 5335–5348.
[28] Seyed Baghestani, et al., Lightweight authenticated key agreement for smart metering in smart grid, IEEE Syst. J. 16 (3) (2022) 4983–4991.
[29] Lunzhi Deng, Ronghai Gao, Certificateless two-party authenticated key agreement scheme for smart grid, Inf. Sci. 543 (2021) 143–156.