Article

Not peer-reviewed version

# Identity-Based Efficient Secure Data Communication Protocol for Hierarchical Sensor Groups in Smart Grid

Yun Feng *, Yi Sun, Yongfeng Cao, Bin Xu, Yong Li

Posted Date: 12 June 2025

doi: 10.20944/preprints202506.1071.v1

Keywords: smart grid; identity-based encryption; lightweight authentication; dynamic key management; hierarchical sensors

 doi:10.20944/preprints202506.1071.v1



*Article*

# Identity-Based Efficient Secure Data Communication Protocol for Hierarchical Sensor Groups in Smart Grid

**Yun Feng [1,*], Yi Sun [2], Yongfeng Cao [1], Bin Xu [1] and Yong Li [3]**

[1] China Electric Power Research Institute, Beijing, China
[2] North China Electric Power University, Beijing, China
[3] State Grid Shandong Electric Power Institute, Shandong, China
* Correspondence: fengyun67067@163.com

**Abstract:** With the rapid evolution of smart grids, secure and efficient data communication among hierarchical sensor devices has become critical to ensure privacy and system integrity. However, existing protocols often fail to balance security strength and resource constraints of terminal sensors. In this paper, we propose a novel identity-based secure data communication protocol tailored for hierarchical sensor groups in smart grid environments. The protocol integrates symmetric and asymmetric encryption to enable secure and efficient data sharing. To reduce computational overhead, a Bloom filter is employed for lightweight identity encoding, and a cloud-assisted pre-authentication mechanism is introduced to enhance access efficiency. Furthermore, we design a dynamic group key update scheme with minimal operations to maintain forward and backward security in evolving sensor networks. Security analysis proves that the protocol is resistant to replay and impersonation attacks, while experimental results demonstrate significant improvements in computational and communication efficiency compared to state-of-the-art methods—achieving reductions of 73.94% in authentication computation cost, 37.77% in encryption, and 55.75% in decryption, along with a 79.98% decrease in communication overhead during authentication.

**Keywords:** smart grid; identity-based encryption; lightweight authentication; dynamic key management; hierarchical sensors

## 1. Introduction

With the wide application of machine learning and network technology, the power system has been rapidly developed, forming the idea of the smart grid consisting of multilayer smart sensors. Smart grids leverage and aggregate information collected by sensors to assist in making power management decisions [1,2]. As a new generation of power systems, the smart grid provides users with stable and reliable power services and achieves efficient operation and intelligent management [3,4]. However, since electric power information is relevant to user privacy, data communication among various sensors and the power service center leads to significant security concerns [5].

To achieve efficient communication while guaranteeing data privacy in smart grid systems, existing secure data communication protocols [6,7] combine asymmetric and symmetric encryption algorithms. With the high efficiency of symmetric encryption methods [8,9], sensitive electric data is masked by a predetermined symmetric secret key, which is only accessed by authorized groups. Due to the security of asymmetric encryption methods [10,11], a pair of asymmetric secret keys is utilized to share the symmetric secret key in a smart grid system.

Recently, identity-based encryption is one of the most effective asymmetric encryption methods for authentication among dynamic hierarchical sensor groups [12]. Specifically, the identity-based encryption method utilizes identity information to guarantee that only authorized users can obtain valuable information from ciphertext. Gupta et al. [13] propose an efficient identity-based protocol

 doi:10.20944/preprints202506.1071.v1



for authentication in transport systems. Zhao et al. [14] further introduce an identity-based broadcast signcryption scheme in the Internet of Vehicles. Shen et al. [15] focus on security enhancement and present an identity-based higncryption protocol. Particularly, for hierarchical architecture scenario applications, Pavithran et al. [16] propose a blockchain-aided protocol to utilize hierarchical identity-based encryption in the Internet of Things systems and Badar et al. [17] propose an identity-based authentication protocol using the physical unclonable function, particularly for smart grid scenarios.

However, these data encryption methods involve complex computations, such as pairing and modular inversion operations. In practice, smart sensors deployed on the terminal side have limited computing resources, while data communication and collaborative analysis require real-time feedback. To match the requirements of real-time grid data processing, developing a secure data communication protocol with efficiency improvement has become a hot topic.

Additionally, the dynamic sensor group leads to a huge cost for secret group key updates [18], particularly in smart grid scenarios. On the one hand, the dense update frequency of the sensor group has a significant impact on the performance of the smart grid system. On the other hand, the change of group members causes forward and backward security concerns for the secret group key.

Based on the analysis above, this paper proposes an identity-based efficient secure data communication protocol for hierarchical sensor groups in smart grid systems. The main contributions of this paper can be summarized in three aspects:

1. We propose a novel secure data communication protocol for sensor groups in the smart grid system, leveraging the symmetric encryption method to transmit obscured data and the identity-based encryption method to share group secret keys. Identities of authorized users are encoded by bloom filter and a cloud-aided pre-verification procedure is introduced. Efficient authentication is achieved by searching the pre-calculated authentication array table in the cloud server.

2. A dynamic update mechanism of the group secret key is designed corresponding with the proposed protocol for lower resource costs in smart grid scenarios. When the sensor group is changed, the proposed mechanism utilizes lightweight operations to implement dynamic updates of the group secret key, which guarantees forward and backward security for the smart grid system.

3. Theoretical analysis demonstrates that our protocol achieves forward and backward security of a dynamic sensor group and has the capability to resist the replay attack and impersonation attack. Experimental evaluation indicates that our protocol performs better than the state-of-the-art protocols. Specifically, for computation cost, the proposed protocol is 73.94% superior to others on average in the authentication process, 37.77% in the encryption process, and 55.75% in the decryption process. For communication cost, the proposed protocol is 79.98% superior to others on average in the authentication process.

The remainder of this paper is organized as follows: The subsequent section presents current research work relevant to this study. Section 3 introduces the system model. The definition related to this study is detailed in Section 4. Section 5 introduces the specific content of this protocol. Section 6 is the security analysis of this protocol, and Section 7 introduces performance analysis. The concluding section encapsulates the research presented in this paper.

## 2. Related Works

### *2.1. Broadcast Encryption Algorithm*

Broadcast encryption was first proposed in 1993 [19]. Recently, Boneh et al. [20] proposed a broadcast encryption scheme using bilinear mapping. The private key and ciphertext of the scheme reached the constant level, which supports the anti-collusion attack and proves that the broadcaster could no longer be a trusted authority but any legitimate user. However, the number of users for this solution has been set at the beginning of the solution, and subsequent users cannot be added. Lewko et al. [21] proposed a broadcast encryption scheme that supports the user's cancellation and non-monotonic mechanism, making the application of broadcast encryption more flexible. However, although the above scheme and subsequent schemes [22,23] realize the advantages of multiple users

 doi:10.20944/preprints202506.1071.v1



sharing the same message, they cannot achieve good access control. Kumar [24] et al. proposed a broadcast encryption technology based on threshold and wildcard. This technique uses hidden access policies to provide security for sensitive data broadcast over insecure channels. In addition, for any number of attributes, the technology realizes the fixed-length ciphertext, which greatly reduces the communication overhead and computational complexity. However, the proposed scheme does not consider the direct withdrawal of users and the dynamic addition of users in the system.

### *2.2. Authentication Protocol*

Wazid et al. [25] proposed a three-factor authentication protocol for remote users in a smart grid environment based on renewable energy. The protocol uses one-way hash functions, bitwise OR operations, and ECC operations to achieve lightweight encryption, the protocol supports the dynamic addition of smart sensors, flexibility of password and biometric updates, user anonymity, and non-traceability. Mahmood et al. [26] proposed a lightweight authentication protocol based on ECC, which used ProVerif, an automatic verification tool, to analyze its own security, and adopted Burrows-Abadi-Needham (BAN) logic to prove the completeness and completeness of the protocol. The downside is that the protocol does not support the anonymity of smart sensors. Kumar et al. [27] proposed a lightweight authentication and key agreement protocol that achieves anonymity, integrity, and security based on ECC, symmetric encryption, hash functions, and message authentication codes. Wang et al. [28] proposed a mutual authentication protocol based on edge computing in a smart grid system, which supports efficient conditional anonymity and key management based on blockchain technology. The proposed protocol ensures mutual authentication and anti-replay attacks and supports efficient key renewal and revocation to achieve conditional anonymity with lower computing and communication costs.

### *2.3. Key Updating Protocol*

Group multicast effectively improves the efficiency of group communication. How to generate and update multicast keys efficiently has important practicability and wide application prospects in the smart grid. At present, a large number of group key generation and updating protocols have been proposed, which can be divided into three categories: key updating protocol based on binary key tree, key updating protocol based on multi-fork key tree, and key updating protocol based on polynomial. Lin et al. [29] proposed an M-fork tree key management and digital signature protocol based on elliptic curves. At the time of transmission, the protocol provides several flexible and scalable schemes to manage security issues that dynamically adapt the framework to the rapidly changing IoV topology, speeding up the time to synchronize system key reconstruction and reducing the number of stages to resynchronize system keys. Tan et al. [30] proposed a dynamic key management scheme based on attribute-based encryption. In the vehicular ad hoc network, identity authentication plays an important role in privacy protection. This protocol guarantees non-repudiation and authenticity properties while achieving efficient vehicular communications.

## 3. System Model

As shown in Figure 1, the system model consists of three entities: cloud server, authorization center, and smart sensor devices.

**Cloud server:** As one of the components of the power service center, the cloud server has sufficient computing resources and storage capacity, it is responsible for storing ciphertext and performing pre-authentication operations for device access requests.

**Authorization center:** The authorization center is a trusted entity that is responsible for generating system parameters, public and private key pairs, user keys, and session keys based on the identity of the smart device, authorization center sends them to the smart device. In addition, the authorization center is responsible for the plaintext encryption to generate ciphertext and send it to the cloud server.

 doi:10.20944/preprints202506.1071.v1



**Smart sensor device:** The smart sensor devices consist of a master station system, gateway devices, and smart sensors. The smart sensor device has limited computing power and storage capabilities, and every device has a public-private key pair, a user key, and a session key. The smart sensor devices use the session key to communicate with the power service center and decrypt the ciphertext using their private keys.

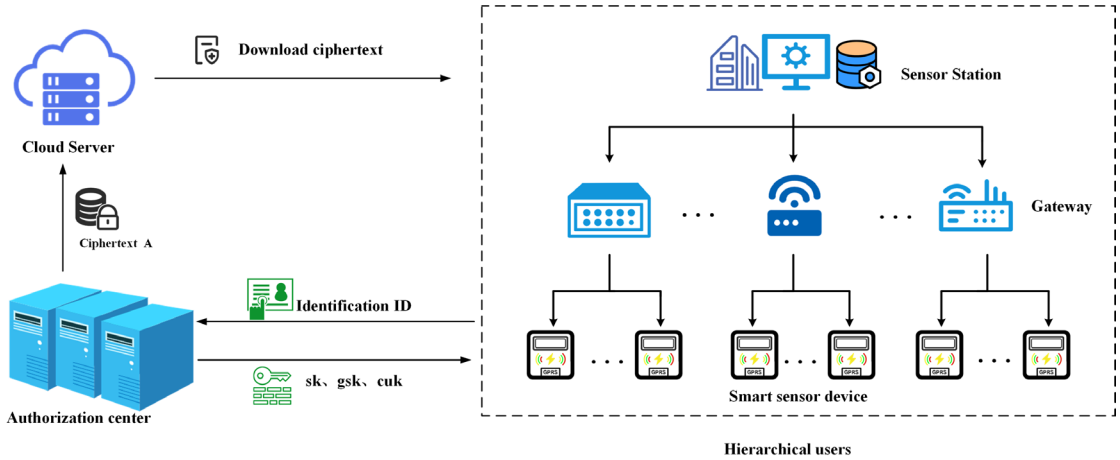


**Figure 1.** System model of the proposed secure data communication protocol.

## 4. Definition

### *4.1. Bloom Filter*

Bloom filter is a random data structure with high special efficiency. The set $S = \{x_1, x_2, \cdots, x_n\}$ is encoded in an array of $W$ bits and the bloom filter determines whether an element x belongs to the set S. The steps to build the (w, m, k, H) -Bloom filter BFS for set S are as follows. First, the set of hash functions $H = \{h_0, h_1, \cdots, h_{k-1}\}$ is selected, where the hash functions $h_0, h_1, \cdots, h_{k-1}$ are independent of each other and have the range $[0, w-1]$, All bits of BFS are then set to 0 initially. Finally, for all $x \in S$ and $0 \le i \le k-1$, let $BF(h_i(x)) = 1$. However, the Bloom filter may misjudge when determining whether the element belongs to the set S by mistaking $x \notin S$ for $x \in S$. When it is necessary to determine whether the element $y$ belongs to the set $S$, simply compute $h_i(y)(0 \le y \le k-1)$ and check $BF(h_i(y))$ whether all values are 1. If the result is not all 1, $y \notin S$, otherwise $y \in S$. The maximum error rate is

$$\varepsilon = p^k(1 + O(\frac{k}{p}\sqrt{\frac{\ln w - k\ln p}{w}})),$$

where $p = 1 - (1 - \frac{1}{w})^{km}$ and $\varepsilon$ is a negligible function of $k$.

### *4.2. Identity-Based Public Key Encryption Algorithm*

The identity-based cryptosystem makes use of the bilinear property of elliptic curves, and bilinear pairing is established by the relation between cyclic subgroups of elliptic curves and multiplicative cyclic subgroups of extended domains. When the difficulty of the extended domain discrete logarithm problem is similar to that of the elliptic curve discrete logarithm problem, an efficient and secure identity-based cryptosystem can be constructed. Identity-based data encryption algorithm is defined as follows:

P1 is set as the generator of the elliptic curve addition cyclic group G1, and P2 is the generator of the elliptic curve addition cyclic group G2. H (·) stands for Hash function; Enc (·) and Dec (·) respectively correspond to the operation modes of encryption and decryption. KDF (·) is a function involved in the key derivation process; MAC (·) is the authentication message code that carries the key in the authentication process; e (·) is a bilinear pair.

 doi:10.20944/preprints202506.1071.v1



The cryptographic function $H_1(Z,n)$ takes the bit string $Z$ and an integer $n$ and outputs an integer $h_1 \in [1,n-1]$. The cryptographic function $H_2(Z,n)$ takes the bit string $Z$ and an integer $n$ and outputs an integer $h_2 \in [1,n-1]$. The key generation center randomly selects $ke \in [1,N-1]$ as mater secret key, and computes $P_{pub} = [ke]P_1$ in $G_1$ as mater public key. The encryption master key pair is $(ke, Ppub)$. The key generation center is kept secret at $ke$ and publicly available at $Ppub$. The key generation center chooses and exposes the one-byte private key to generate the function identifier $hid$, The identity of the user A is $ID_A$, and generates A's private key $d_A$. The key generation center first computes $t_1 = H_1(ID_A||hid,N) + ke$ in the bounded domain $F_N$. If $t_1 = 0$, the system needs to re-generate the master private key, compute and expose the master public key, and update the existing user's private key; Otherwise, calculate $t_2 = ke \cdot t_1^{-1}$, then calculate $d_A = [t_2]P_1 = [S/(H_1(ID_A||hid) + s)]P_1$.

Assume that user A encrypts plaintext and sends it to user B, A computes element $Q_B = [H_1(ID_B||hid,N)]P_1 + P_{pub}$ of the group $G_1$, and then randomly selects $r \in [1,N-1]$, and computes $C_1 = [r]Q_B$, $g = e(P_{pub},P_2), w = g^r$. A calculates $K = KDF(C_1||w||ID_B)$. According to the classification of encryption, there are two ways to encrypt plaintext to generate $C_2$: stream cipher $C_2 = M + K$ and block cipher $C_2 = Enc(M,K)$. A computes $C_3 = MAC(K,C_2)$, and finally obtains the ciphertext $C = C_1||C_2||C_3$.

After receiving the ciphertext, user B calculates $w' = e(C_1, ID_B)$, $K' = KDF(C_1||w'||ID_B)$. According to the classification of encryption, there are two ways to decrypt ciphertext to generate $M'$: stream cipher $M' = C_2 + K_1$ and block cipher $M' = Dec(C_2 + K_1)$. Finally, B computes $u = MAC(K_2', C_2)$. If $u = C_3$, output $M'$, otherwise an error is reported.

## 5. Protocol

This protocol mainly includes 6 steps, which are system initialization, key generation, data encryption, user authentication, data decryption, and dynamic updating of terminal devices. The following is the specific interaction process.

### *5.1. System Initialization*

Given security parameter $\lambda$, the authorization center (CA) selects a particular elliptic curve $E_q(a,b)$ and a point $P$ of large prime order on an elliptic curve on a finite field, selects $G_1$、$G_2$ as addition cyclic groups of prime $N$, and $G_T$ selects a multiplicative cyclic group of primes $N$. Select bloom filter's bit array size $m$ and k hash functions which map every identity in the user's set to $\{1,2,\cdots,m\}$. Generally, $k$ hash functions are used to calculate the elements in the authorized user set, and the obtained results are modulo m, and the bloom filter's pre-authentication array table A is obtained and stored in the server.

### *5.2. Key Generation*

CA selects a number $x \in [1,N-1]$ at random as system master key $msk$, and the corresponding system master public key $mpk = x \cdot P$ is disclosed. CA selects and exposes the private key generating function identifier hid, which is represented by a single byte. AC computes $t_i^j = H_1(ID_i^j||hid,N) + x$ over a finite field. If $t_i^j \neq 0$, calculate the user's private key $sk_i^j = x \cdot (t_i^j)^{-1}$, and the corresponding user's public key is $pk_i^j = x \cdot (t_i^j)^{-1} \cdot P$. If $t_i^j = 0$, the system master key and public key need to be recalculated. CA randomly selects $R_i^j$, timestamp $T$, the system master public key $mpk$, the system private key $msk$, and user identification $ID_i^j$ to generate the user key $K_i^j = H(msk||ID_i^j||R_i^j||T)$. In addition, $KDC$ generates a random number $rand_i^j$ for every smart sensor device to generate session key $uk_i^j = H_1(K_i^j||rand_i^j)$.

**Broadcast key generation:** For every smart sensor or gateway device, CA generates the broadcast key $gsk = KGen(msk,GID)$ through the master public key $mpk$, the master private key $msk$ and $GID = H_1(ID_1^1||\cdots||ID_m^n)$ of all devices, where n represents the last layer, and m represents

 doi:10.20944/preprints202506.1071.v1



the last smart sensor device at the last layer. CA selects $grand$ at random to generate the broadcast session key $guk = H_1(gsk||grand)$.

**Multicast key generation:** When the power service center wants to multicast with the subset of users, the user key and the hash value of user identities are calculated as the multicast key:

$$csk = H_1(H_2(K_{t_1}||\cdots||K_{t_m})||H_2(ID_{t_1}||\cdots||ID_{t_m})||T),$$

where $T$ is the current timestamp, $K_{t_j}$ and $ID_{t_j}$ indicate the user key and identity of the user set, respectively. CA randomly selects grand to generate the multicast session key $cuk = H_1(csk||grand)$.

### *5.3. Data Encryption*

**a) Unicast encryption:** As shown in Figure 2, CA uses the user's public key $PK_i^j$ to encrypt the user key to get $C_i^j = PEnc(PK_i^j, K_i^j)$, and then uses the unicast key to encrypt the session key to obtain $Enc(K_i^j, uk_i^j)$, and finally uses the session key to encrypt the plaintext $M$ to obtain $Enc(uk_i^j, M)$, where $PEnc$ represents identity based public key encryption algorithm, and $Enc$ represents any symmetric encryption algorithm.

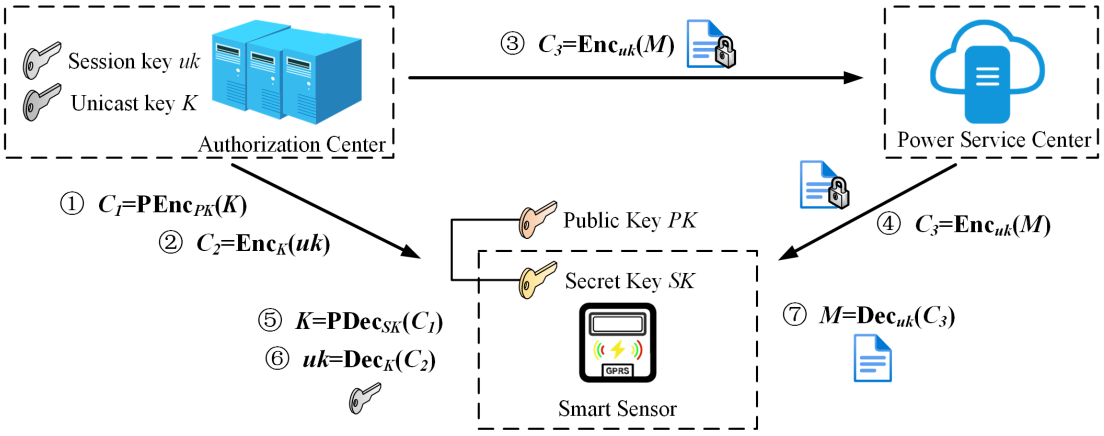


**Figure 2.** Workflow of the secure data unicast service.

**b) Broadcast encryption:** As shown in Figure 3, CA uses the user's public key $PK_i^j$ to encrypt the broadcast key to get $C = PEnc(PK_i^j, gsk)$, and then uses the broadcast key to encrypt the broadcast session key to obtain $Enc(gsk, guk)$, and finally uses the broadcast session key to encrypt the plaintext $M$ to obtain $Enc(guk, M)$.

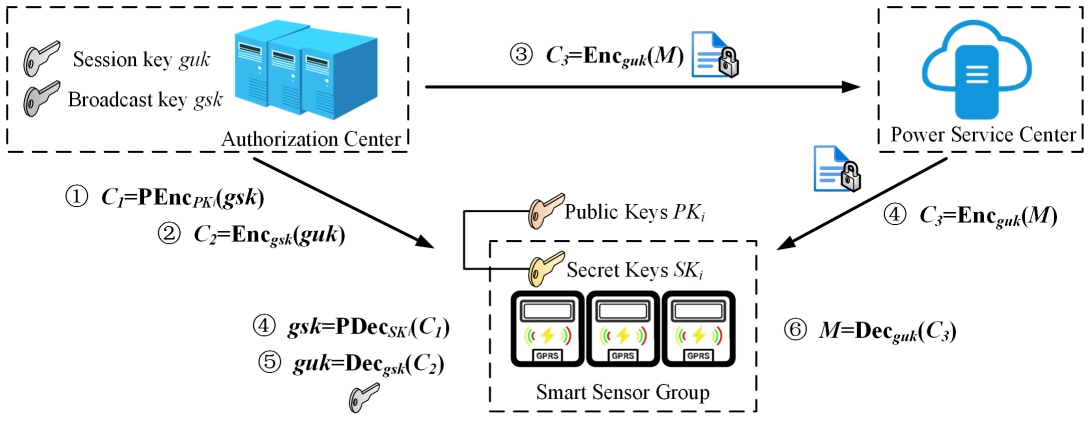


**Figure 3.** Workflow of the secure data broadcast service.

 doi:10.20944/preprints202506.1071.v1



**c) Multicast encryption:** As shown in Figure 4, CA uses the user's public key $PK_i^j$ to encrypt the multicast key to get $C' = PEnc(PK_i^j, csk)$, and then uses the multicast key to encrypt the multicast session key to obtain $Enc(csk, cuk)$, and finally uses the multicast session key to encrypt the plaintext $M$ to obtain $Enc(cuk, M)$.

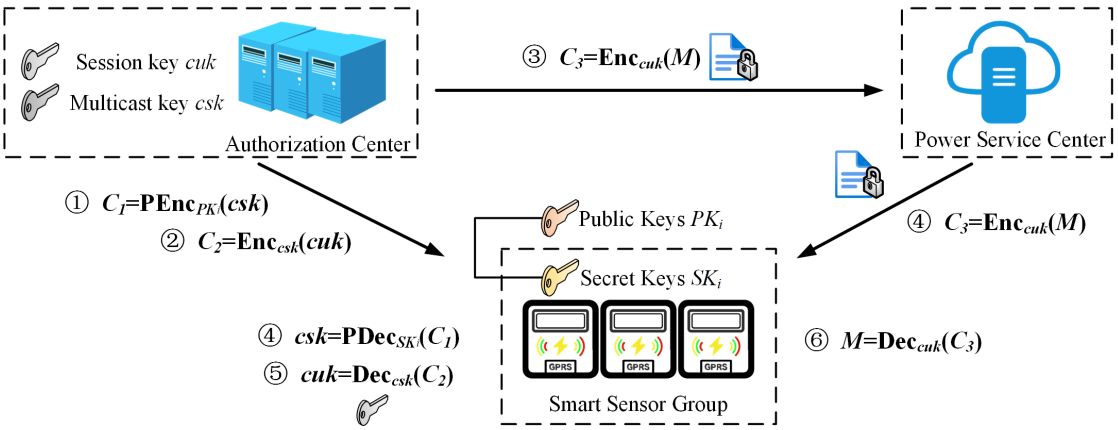


**Figure 4.** Workflow of the secure data multicast service.

*5.4. User Authentications*

If the user wants to make a data access request to the power service center, it first verifies the legitimacy of the identity. As shown in Figure 5, user authentications include pre-authentication and main authentication, the specific process is as follows.

**Pre-authentication:** The power service center uses the hash function disclosed by CA in the initialization stage to hash the user identity and get the pre-authentication array $A'$ generated by the bloom filter. The power service center pre-verifies the user identity: If $A' \subseteq A$, the user passes the authentication and obtains the pre-authentication certification. The power service center calculates $R_B = msk \cdot pk_i^j = x \cdot x(t_i^j)^{-1} \cdot P$.

**Main authentication:** After passing the pre-authentication, the user uses his private key and the system master public key to calculate the shared key $R_A = sk_i^j \times mpk$, and calculates

$Auth = H_1(R_A||T_A||timestamp)$.

Then, the user sends $Auth||timestamp$ to the power service center. The power service center receives $Auth||timestamp$ at $timestamp'$ moment and firstly checks whether the time interval between $timestamp$ and $timestamp'$ meets $|timestamp' - timestamp| \leq \Delta T$, where $\Delta T$ represents the transmission delay. If the timestamp falls within the accepted time window, the power service center calculates $Auth' = H_1(R_B||T_A||timestamp)$, if $Auth' = Auth$, then it proves that the user has the correct certification and public and private key pair, and the user successfully passes the authentication.

 doi:10.20944/preprints202506.1071.v1



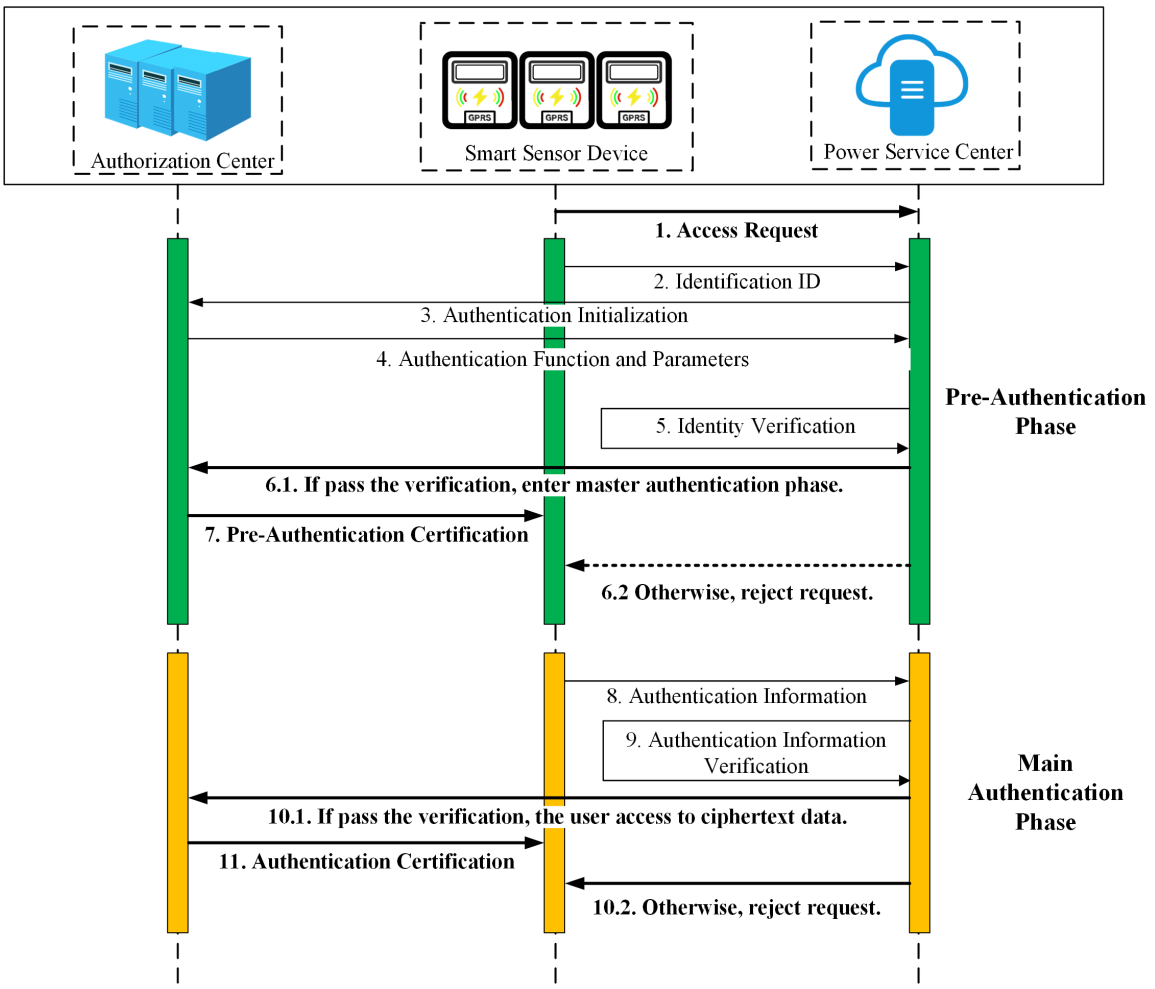


**Figure 5.** Workflow of the user authentication phase.

*5.5. Data Decryption*

**a) Unicast decryption:** As shown in Figure 2, the smart sensor device uses its private key $SK_i^j$ to calculate $PK_i^j = PDec(SK_i^j, C_i^j)$, decrypts the unicast key, then the session key $uk_i^j$ by using the user key, and finally decrypts the plaintext $M$ by using the session key.

**b) Broadcast decryption:** As shown in Figure 3, the smart sensor device uses its own private key $SK_i^j$ to calculate $gsk = PDec(SK_i^j, C)$ to obtain the user's broadcast key, and then the broadcast session key is obtained by using the broadcast key. Finally, the plaintext $M$ is decrypted by using the broadcast session key.

**c) Multicast decryption:** As shown in Figure 4, the smart sensor device uses its own private key $SK_i^j$ to calculate $csk = PDec(SK_i^j, C')$ to obtain the user's multicast key, and then multicast session key $cuk$ is obtained by using the multicast key. Finally, the plaintext $M$ is decrypted by using the multicast session key.

 doi:10.20944/preprints202506.1071.v1



*5.6. Dynamic Updating of Terminal Devices*

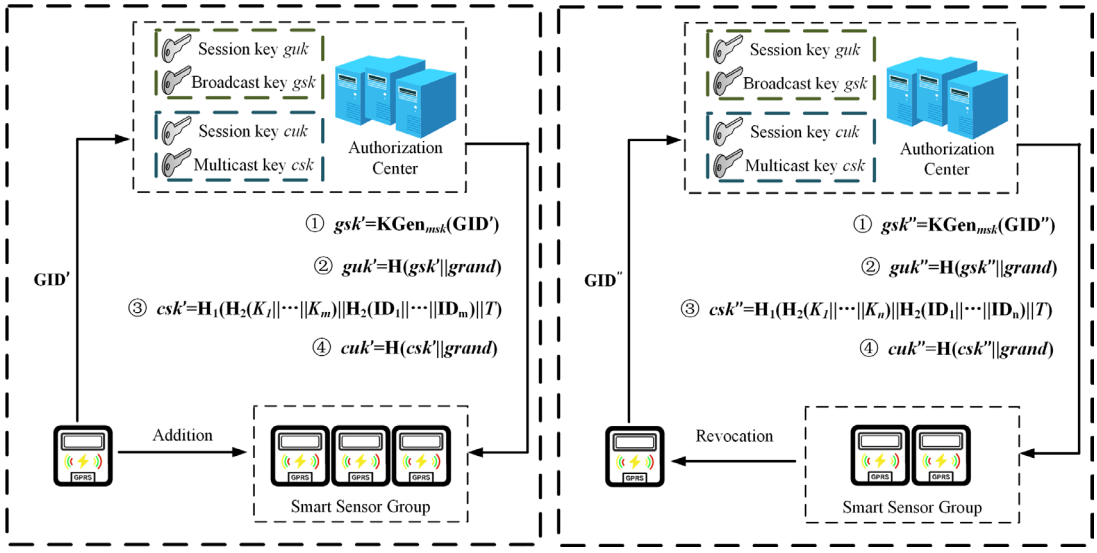


**Figure 6.** Workflow of the dynamic key management.

5.6.1. Smart Sensor Device Addition

As shown in Figure 6, when a new user is added to the system, the system generates the public and private keys pair and user key and updates the broadcast key, broadcast session key, multicast key, and multicast session key. The public key, private key, and user key follow the steps for key generation.

**a) Updating the broadcast key and broadcast session key:** CA generates the broadcast key $gsk' = KGen(msk, GID)$ using the master public key $mpk$, master private key $msk$, and identity of the existing device $GID$. AC selects $grand$ at random to generate the broadcast session key $guk' = H_1(gsk||grand)$.

**b) Updating the multicast key and multicast session key:** When the power service center wants to multicast with a subset of users containing new users, it firstly recalculates the hash value of the user keys and identities for users: $csk' = H_1(H_2(K_{t_1}||\cdots||K_{t_m})||H_2(ID_{t_1}||\cdots||ID_{t_m})||T)$, where $T$ indicates the current timestamp, and $K_{t_j}, ID_{t_j}$ indicates the user keys and identities of users respectively. CA randomly selects $grand$ to generate the multicast session key $cuk' = H_1(csk'||grand)$.

5.6.2. Smart Sensor Device Revocation

As shown in Figure 6, when a smart sensor device is revoked, its public key, private key, and authentication certification will expire, and it can no longer be used for data access.

**a) Updating the broadcast key and broadcast session key:** When a smart device is removed, the broadcast key needs to be recalculated. According to the key generation algorithm, the broadcast key $gsk''$ is recalculated, and then the broadcast session key $guk''$ is generated.

**b) Updating the multicast key and multicast session key:** When a smart device is removed, the multicast key needs to be recalculated. According to the key generation algorithm, the multicast key $csk''$ is recalculated, and then the multicast session key $cuk''$ is generated.

**c) Smart sensor device migration:** When a subset of user devices is migrated from one gateway to another gateway device, CA needs to recalculate the hash values of the user keys and identities for all nodes on the path from this subset of users to the root node, and then obtain a new multicast key $csk''$, which is based on the current new timestamp $\tilde{T}$. $csk''$ is broadcast to users in this subset with the previous multicast session key, and upon receiving the message, users decrypt and update the multicast session key $cuk''$.

 doi:10.20944/preprints202506.1071.v1



## 6. Security Analysis

This section formally analyzes the security properties of the proposed protocol. The security relies on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), the Computational Diffie-Hellman (CDH) problem on elliptic curves, and the security of the underlying identity-based encryption (IBE) scheme $(\mathrm{PEnc}, \mathrm{PDec})$ and symmetric encryption scheme $(\mathrm{Enc}, \mathrm{Dec})$. The hash functions $H, H_1, H_2$ are assumed to be cryptographically secure (e.g., behaving as random oracles).

### *6.1. Forward Security*

**Theorem 1.** Forward security ensures that a smart sensor device $SM_r$ (with identity $ID_r$), once revoked from a group at time $T_{rev}$, cannot decrypt messages intended for the group after $T_{rev}$. This means $SM_r$ cannot obtain any newly generated group keys (e.g., $gsk'$, $csk'$) or session keys (e.g., $guk'$, $cuk'$).

**Proof.**

Consider the broadcast key $gsk$. When $SM_r$ is revoked, a new broadcast key $gsk'$ is generated by CA using $gsk' = \mathrm{KGen}(msk, GID')$, where $GID'$ represents the identities of the remaining authorized devices. This $gsk'$ is then used to derive a new broadcast session key $guk' = H_1(gsk'||grand')$. The plaintext $M$ is encrypted as $\mathrm{Enc}(guk', M)$. The key $gsk'$ is distributed to authorized users by encrypting it with their respective public keys $PK_u$ (derived from $ID_u$) using the IBE scheme: $C_u = \mathrm{PEnc}(PK_u, gsk')$.

Since $SM_r$ is revoked, its identity $ID_r$ is not part of $GID'$, and CA will not provide $SM_r$ with $\mathrm{PEnc}(PK_r, gsk')$. Even if $SM_r$ possesses its old private key $sk_r$, it cannot decrypt $C_u$ for any $ID_u \in GID'$ (assuming $ID_u \neq ID_r$) to obtain $gsk'$, due to the security of the IBE scheme. Without $gsk'$, $SM_r$ cannot compute $guk'$ via $H_1(gsk'||grand')$ (as $grand'$ is fresh and $H_1$ is one-way). Therefore, $SM_r$ cannot decrypt $\mathrm{Enc}(guk', M)$.

The argument for the multicast key $csk'$ is similar. When $SM_r$ is revoked from a multicast group, a new $csk''$ is computed based on the remaining members' keys and identities, and a new timestamp. This $csk''$ is distributed encrypted with the previous multicast session key $cuk$. Since $SM_r$ is no longer part of the group, it will not receive this update, or if it does, the subsequent session key $cuk''$ will be derived from $csk''$ which it cannot obtain if $csk''$ is re-encrypted using IBE for the new group. If $csk''$ is broadcast using the old $cuk$, then the subsequent $cuk''$ derived from $csk''$ and a new $grand''$ will be unknown to $SM_r$. The core idea is that new keys are generated that $SM_r$ does not have the components to derive or decrypt.

Thus, the protocol ensures forward security provided the IBE scheme is secure and hash functions are one-way. An adversary controlling $SM_r$ cannot gain access to future group communications.

### *6.2. Backward Security*

**Theorem 2.** Backward security ensures that a newly added smart sensor device $SM_n$ (with identity $ID_n$) at time $T_{add}$ cannot decrypt messages encrypted for the group before its addition.

**Proof.**

Before $SM_n$ joins, group communications use keys like $gsk_{old}$ (and $guk_{old}$) or $csk_{old}$ (and $cuk_{old}$). These keys were generated based on the identities and keys of members existing before $T_{add}$. For instance, $gsk_{old} = \mathrm{KGen}(msk, GID_{old})$, where $ID_n \notin GID_{old}$.

When $SM_n$ joins, it receives its own private key $sk_n$, user key $K_n$, and session key $uk_n$. However, $SM_n$ does not receive past group keys like $gsk_{old}$ or $csk_{old}$. The IBE-encrypted $gsk_{old}$ was distributed only to members of $GID_{old}$. Since $ID_n$ was not in $GID_{old}$, $sk_n$ cannot be used to decrypt ciphertexts containing $gsk_{old}$.

The user keys $K_u = H(msk||ID_u||R_u||T)$ are unique to each user and timestamp. A new user $SM_n$ cannot derive past user keys of other members due to the one-way nature of $H$ and the secrecy of $msk$ and other users' $R_u$. Similarly, past multicast keys $csk_{old} =$

 doi:10.20944/preprints202506.1071.v1



$H_1(H_2(K_{t_1}||\dots)||H_2(ID_{t_1}||\dots)||T_{old})$ depend on keys and identities of the old group and an old timestamp, which $SM_n$ cannot reconstruct.

Therefore, $SM_n$ cannot access messages encrypted prior to its joining the group, ensuring backward security. This relies on the IBE security and the one-way property of hash functions.

### *6.3. Replay Attack Resistance*

**Theorem 3.** The protocol resists replay attacks during user authentication. The user sends $Auth = H_1(R_A||T_A||timestamp)$ and $timestamp$ to the power service center. $R_A = sk_i^j \cdot mpk$.

**Proof.**

The power service center (server) receives $(Auth, timestamp)$ at its current time $timestamp'$. It first verifies the freshness of the $timestamp$ by checking if $|timestamp' - timestamp| \leq \Delta T$, where $\Delta T$ is a predefined small interval for network delay.

If an adversary intercepts a valid $(Auth_1, timestamp_1)$ and replays it at a significantly later time ${timestamp'}_2$, then $|{timestamp'}_2 - timestamp_1| > \Delta T$. The server will detect this stale timestamp and reject the message.

If the adversary attempts to use a fresh $timestamp_2$ with the old $Auth_1$, the server will compute ${Auth'}_{check} = H_1(R_B||T_A||timestamp_2)$ (where $R_B = msk \cdot pk_i^j = R_A$ ). Since $timestamp_2 \neq timestamp_1$ and $H_1$ is collision-resistant, $Auth_1 = H_1(R_A||T_A||timestamp_1)$ will not be equal to ${Auth'}_{check}$ (except with negligible probability). Thus, the authentication will fail.

To successfully replay with a fresh timestamp $timestamp_2$, the adversary would need to compute a new $Auth_2 = H_1(R_A||T_A||timestamp_2)$. This requires knowledge of $R_A = sk_i^j \cdot mpk$. Since the user's private key $sk_i^j$ is secret, and computing $sk_i^j \cdot mpk$ without $sk_i^j$ is hard (related to the CDH problem, given $pk_i^j = sk_i^j P$ and $mpk = xP$), the adversary cannot forge a valid $Auth_2$.

The inclusion of a fresh timestamp in the hash computation for $Auth$ and the server's freshness check effectively prevent replay attacks.

### *6.4. Impersonation Attack Resistance*

**Theorem 4.** An adversary attempts to impersonate a legitimate smart sensor device $SM_u$ (with identity $ID_u$ and private key $sk_u$) to the power service center.

**Proof.**

To impersonate $SM_u$, the adversary must successfully complete the authentication process. This involves computing $Auth = H_1(R_A||T_A||timestamp)$, where $R_A = sk_u \cdot mpk$. The adversary knows $ID_u$, $mpk = xP$, and $pk_u = sk_u P$. The system master key $x$ and user private key $sk_u$ are secret.

The private key $sk_u$ is computed as $x \cdot (t_u)^{-1}$, where $t_u = H_1(ID_u||hid, N) + x$. Thus, $R_A = (x \cdot (t_u)^{-1}) \cdot (xP) = x^2(t_u)^{-1}P$.

An adversary faces the following difficulties:

Deriving $sk_u$ from $pk_u = sk_u P$: This is the ECDLP, which is assumed to be hard.

Deriving $x$ from $mpk = xP$: This is also the ECDLP.

Computing $R_A = sk_u \cdot mpk$ directly from $pk_u$ and $mpk$ without knowing $sk_u$ or $x$: This is computationally equivalent to solving the CDH problem (given $P, sk_u P, xP$, compute $sk_u xP$). While $R_A$ is not exactly $sk_u xP$, computing $x^2(H_1(ID_u||hid, N) + x)^{-1}P$ without $x$ is infeasible.

The pre-authentication step using the Bloom filter adds another layer. If the adversary's chosen (or forged) identity $ID_{adv}$ is not in the authorized set encoded in the Bloom filter $A$ on the server, ${A'}_{adv} \notin A$, and pre-authentication fails. Even if a Bloom filter collision occurs for a random $ID_{adv}$ (with small probability $\epsilon$), the subsequent main authentication requiring $R_A$ will fail.

Since the adversary cannot compute $sk_u$ or $R_A$ without breaking underlying hard problems (ECDLP or CDH), they cannot generate a valid $Auth$ message. Therefore, the protocol is resistant to impersonation attacks.

 doi:10.20944/preprints202506.1071.v1



## 7. Performance Analysis

### *7.1. Theoretical Analysis*

This section will be analyzed in terms of computational cost and communication cost. In computational cost analysis, we focus only on the bilinear pairing, multiplication, hashing, and modular inverse algorithms performed by every device, and ignore other lightweight operations. We represent $P$ as a bilinear pairing operation, $m$ as a multiplication operation, $h$ as a hash operation, and $r$ as a modular power operation. In the registration phase, the power service center generates the unicast key for every smart grid device, which requires three multiplications, three hashes, and one modular power, namely $3m + 3h + 1r$. The power service center needs $2h$ operations to generate the broadcast key, and $2h$ operations to generate the multicast key. In the encryption phase, unicast encryption, broadcast encryption, and multicast encryption all require one asymmetric encryption operation and two symmetric encryption operations. In the authentication phase, CA authenticates user identity by one multiplication and one hash. In communication cost analysis, CA generates the private key, session key, broadcast key, broadcast session key, multicast key, and multicast session key for the user. Their key's bit length is $\lambda$. During data access, the user downloads the ciphertext from the power service center. We will ignore other transmitted data.

Table 1 shows the cost comparison of the certification process between the proposed protocol and the protocol of MAH [26], WAN [28], and ZHA [14]. In our protocol, CA requires one multiplication and one hash operation to verify the validity of the terminal device. Therefore, the time consumed by CA authentication increases linearly with the increasing number of terminal devices. As shown in Table 2, the cost of the proposed protocol is lower than other protocols, both in terms of computation cost and communication cost. MAH protocol has the highest computational cost, and WAN protocol has the highest communication cost.

Table 2 shows the cost comparison of encryption and decryption between the proposed protocol and the protocol of CHE [22], ACH [23], and KUM [27]. The protocol of CHE and KUM has low encryption costs but requires a lot of bilinear pairing operations in the key generation and decryption stage, which is not suitable for resource-limited terminal devices in smart grids. In the decryption cost, the proposed protocol has a lower cost.

**Table 1.** Theoretical cost comparison of the certification process for different schemes.

| Protocol | Computation cost | Communication cost |
|---|---|---|
| MAH | $5m + 5h$ | $3\lambda + 96$ |
| WAN | $6m + 3h$ | $3\lambda^2 + 2\lambda + 64$ |
| ZHA | $5m + 4h$ | $5\lambda$ |
| Ours | $m + h$ | $\lambda + 32$ |

**Table 2.** Theoretical cost comparison of encryption and decryption for different schemes.

| Protocol | Computation cost | Communication cost |
|---|---|---|
| CHE | $5r$ | $2P + nm$ |
| ACH | $3P + 4m$ | $3P$ |
| KUM | $m + 3r$ | $2P + r$ |
| Ours | $P + 2m + h + r$ | $P$ |

### *7.2. Experimental Analysis*

In this section, we conduct some experiments to prove that the protocol we designed is efficient. We compare the proposed scheme with the schemes in CHE, ACH, and KUM in terms of encryption cost and decryption cost.

In our experiments, we used a computer with an Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz to simulate the CA in the design system. In the experiment, the C++ programming

 doi:10.20944/preprints202506.1071.v1



language is used to implement the designed protocol, and PBC library is used to multiply and modular exponents of elliptic curves. The hash function in our experiment is SHA-256.

As shown in Figure 7, both the computation and communication costs of the proposed protocol are less than other authentication protocols. Since the bloom filter is introduced to encode the identities of smart sensor devices, the proposed protocol reduces the computation and communication cost of the certification phase. Detailed experimental results are shown in Table 3 and Table 4. With the increasing number of terminal devices, the advantage of our protocol becomes more obvious. Specifically, the proposed protocol is 73.94% superior to others on average in terms of computation and 79.98% in terms of communication.

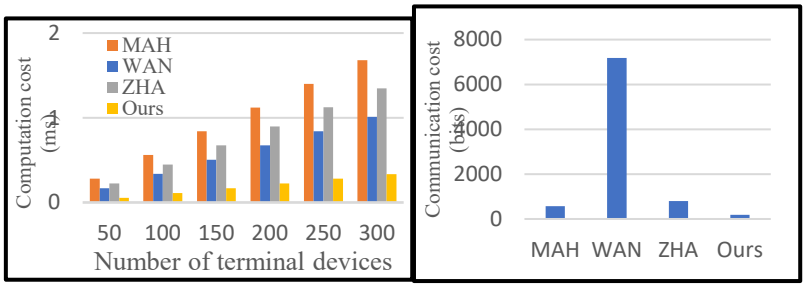


**Figure 7.** Comparison of computation cost and communication cost of different protocols.

**Table 3.** Computation cost comparison of the certification process for different schemes.

| Protocol | Number of Devices | | | | | |
|---|---|---|---|---|---|---|
| | **50** | **100** | **150** | **200** | **250** | **300** |
| MAH | 0.28 ms | 0.56 ms | 0.84 ms | 1.12 ms | 1.40 ms | 1.68 ms |
| WAN | 0.16 ms | 0.34 ms | 0.51 ms | 0.67 ms | 0.84 ms | 1.01 ms |
| ZHA | 0.22 ms | 0.45 ms | 0.67 ms | 0.90 ms | 1.12 ms | 1.35 ms |
| Ours | 0.06 ms | 0.11 ms | 0.17 ms | 0.22 ms | 0.28 ms | 0.34 ms |

**Table 4.** Communication cost comparison of the certification process for different schemes.

| **Protocol** | MAH | WAN | ZHA | Ours |
|---|---|---|---|---|
| **Cost** | 576 bits | 7184 bits | 800 bits | 192 bits |

Figure 8 and Figure 9 show comparisons of data encryption and decryption costs for each scheme with the increasing number of terminal devices. The proposed scheme is obviously superior to other schemes in terms of encryption and decryption cost. Detailed experimental results are shown in Table 5 and Table 6. Specifically, the running time of unicast encryption and multicast encryption increases linearly with the increase of terminal devices. Broadcast encryption does not consume excessive uptime due to changes in the number of devices. No matter how many devices are added, broadcast encryption is calculated only once using the device's identity.

 doi:10.20944/preprints202506.1071.v1



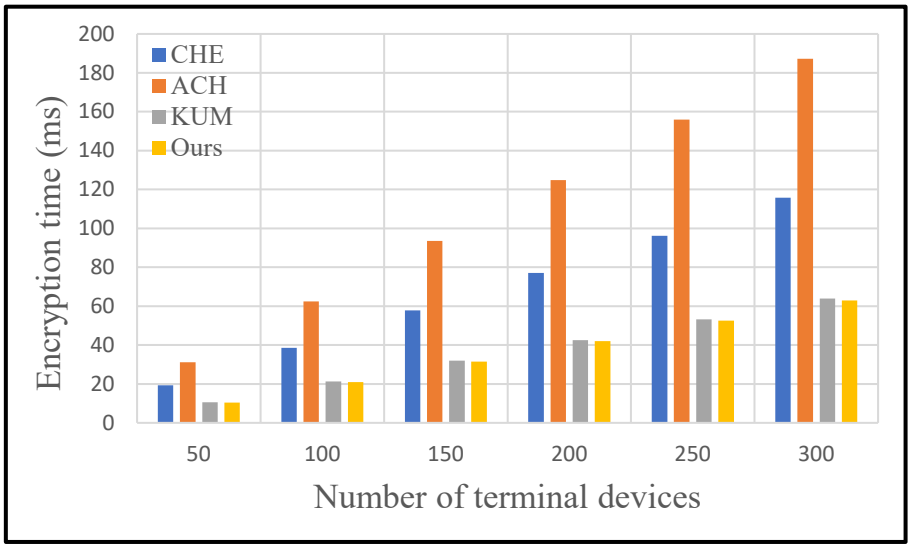


**Figure 8.** Comparison of encryption time of different protocols.

**Table 5.** Computation cost comparison of the encryption process for different schemes.

| Protocol | Number of Devices | | | | | |
|---|---|---|---|---|---|---|
| | 50 | 100 | 150 | 200 | 250 | 300 |
| CHE | 19.29 ms | 38.58 ms | 57.87 ms | 77.16 ms | 96.25 ms | 115.74 ms |
| ACH | 31.20 ms | 62.44 ms | 93.61 ms | 124.87 ms | 156.34 ms | 187.24 ms |
| KUM | 10.65 ms | 21.30 ms | 31.95 ms | 42.60 ms | 53.25 ms | 63.90 ms |
| Ours | 10.50 ms | 21.00 ms | 31.50 ms | 42.00 ms | 52.51 ms | 63.12 ms |

For the encryption process, the proposed protocol performs best among four secure data communication protocols. KUM protocol also performs well with the increasing number of smart sensor devices, followed by CHE protocol, while ACH protocol showcases the worst performance. Specifically, the proposed protocol is 37.77% superior to others on average in terms of encryption cost.

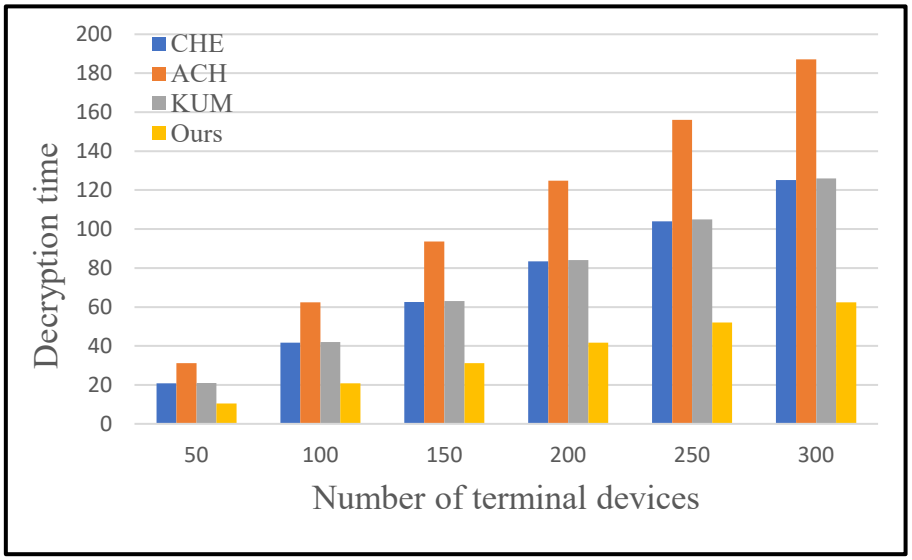


**Figure 9.** Comparison of decryption time of different protocols.

 doi:10.20944/preprints202506.1071.v1



**Table 6.** Computation cost comparison of the decryption process for different schemes.

| Protocol | Number of Devices | | | | | |
|---|---|---|---|---|---|---|
| | 50 | 100 | 150 | 200 | 250 | 300 |
| CHE | 20.85 ms | 41.57 ms | 62.55 ms | 83.41 ms | 104.02 ms | 125.13 ms |
| ACH | 31.20 ms | 63.91 ms | 94.34 ms | 123.99 ms | 154.97 ms | 188.01 ms |
| KUM | 21.05 ms | 42.17 ms | 63.18 ms | 84.06 ms | 105.21 ms | 126.20 ms |
| Ours | 10.39 ms | 20.82 ms | 31.27 ms | 41.60 ms | 51.98 ms | 62.42 ms |

For the decryption process, the proposed protocol is the most efficient protocol among four secure data communication protocols as well. Particularly, the KUM protocol and CHE protocol demonstrate similar performance on efficiency in the decryption phase, which significantly falls behind our protocol. In terms of decryption efficiency, the ACH protocol is the worst protocol as well. Specifically, the proposed protocol is 55.75% superior to others on average in terms of decryption cost.

According to the evaluations above, our protocol showcases comprehensive advantages in the secure data communication process. The proposed protocol obviously demonstrates the best performance on authentication, encryption, and decryption progress. In conclusion, the proposed protocol provides an effective and efficient solution for secure data communication in smart grid scenarios.

## 8. Conclusions

In this paper, we propose a novel secure data communication protocol for sensor groups in the smart grid system. The proposed protocol leverages the symmetric encryption method to transmit obscured data and utilizes the identity-based encryption method to share group secret keys. To improve communication efficiency, the identities of authorized users are encoded by the bloom filter, and a cloud-aided pre-verification procedure is introduced. Correspondingly, a novel dynamic update mechanism of the group secret key is designed in smart grid scenarios. When the sensor group is changed, the proposed mechanism utilizes lightweight operations to implement dynamic updates of the group secret key. Theoretical analysis demonstrates that our protocol achieves forward and backward security of a dynamic sensor group and has the capability to resist the replay attack and impersonation attack. Experimental evaluation indicates that our protocol performs better than the state-of-the-art protocols. Specifically, for computation cost, the proposed protocol is 73.94% superior to others on average in the authentication process, 37.77% in the encryption process, and 55.75% in the decryption process. For communication cost, the proposed protocol is 79.98% superior to others on average in the authentication process.

**Author Contributions:** Conceptualization, Yun Feng; methodology, Yun Feng and Bin Xu; software, Yongfeng Cao; validation, Yongfeng Cao.; formal analysis, Yi Sun; writing—original draft preparation, Yun Feng; writing—review and editing, Yong Li; visualization, Yun Feng; supervision, Bin Xu; funding acquisition, Yun Feng. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by a grant from the National Key Research and Development Program of China (No. 2022YFB2402900) Key Techniques of Adaptive Grid Integration and Active Synchronization for Extremely High Penetration Distributed Photovoltaic Power Generation and Science and Technology Project of State Grid Corporation of China under grant 52060023001T(Key Techniques of Adaptive Grid Integration and Active Synchronization for Extremely High Penetration Distributed Photovoltaic Power Generation).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

 doi:10.20944/preprints202506.1071.v1



**Acknowledgments**: The authors acknowledge Wenting Wang and Sun Li for their assistance in this study.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Fang X, Misra S, Xue G, et al. Smart grid—The new and improved power grid: A survey. IEEE Communications Surveys & Tutorials, 2011, 14(4), 944-980.
2. Gungor V C, Sahin D, Kocak T, et al. Smart grid technologies: Communication technologies and standards. IEEE Transactions on Industrial Informatics, 2011, 7(4), 529-539.
3. Liu Z, Cao Z, Dong X, et al. EPMDA-FED: Efficient and Privacy-Preserving Multidimensional Data Aggregation Scheme With Fast Error Detection in Smart Grid[J]. IEEE Internet of Things Journal, 2022, 9(9), 6922-6933.
4. Zhang X, You L, Hu G. An Efficient and Robust Multidimensional Data Aggregation Scheme for Smart Grid Based on Blockchain. IEEE Transactions on Network and Service Management, 2022, 19(4), 3949-3959.
5. Zuo X, Li L, Peng H, et al. Privacy-Preserving Multidimensional Data Aggregation Scheme Without Trusted Authority in Smart Grid. IEEE Systems Journal, 2021, 15(1), 395-406.
6. Halak B, Yilmaz Y, Shiu D. Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications. IEEE Access, 2022, 10, 76707-76719.
7. Patgiri R, Muppalaneni N B. Stealth: A highly secured end-to-end symmetric communication protocol. International Symposium on Networks, Computers and Communication, 2022, 1-8.
8. Gadhiya N, Tailor S, Degadwala S. A review on different level data encryption through a compression techniques, International Conference on Inventive Computation Technologies, 2024, 1378-1381.
9. Jimale M A, Z'aba M R, Kiah M L B M, et al. Authenticated encryption schemes: A systematic review. IEEE Access, 2022, 10, 14739-14766.
10. Shen Y, Sun Z, Zhou T. Survey on asymmetric cryptography algorithms. International Conference on Electronic Information Engineering and Computer Science, 2021, 464-469.
11. Gadhiya N, Tailor S, Degadwala S. A review on different level data encryption through a compression techniques. International Conference on Inventive Computation Technologies, 2024, 1378-1381.
12. Al-Shareeda M A, Anbar M, Manickam S, et al. Security and privacy schemes in vehicular ad-hoc network with identity-based cryptography approach: A survey. IEEE Access, 2021, 9, 121522-121531.
13. Gupta B B, Gaurav A, Hsu C H, et al. Identity-based authentication mechanism for secure information sharing in the maritime transport system. IEEE Transactions on Intelligent Transportation Systems, 2021, 24(2), 2422-2430.
14. Zhao Y, Wang Y, Liang Y, et al. Identity-based broadcast signcryption scheme for vehicular platoon communication. IEEE Transactions on Industrial Informatics, 2022, 19(6), 7814-7824.
15. Shen S, Wang H, Zhao Y. Identity-based authenticated encryption with identity confidentiality. Theoretical Computer Science, 2022, 901, 1-18.
16. Pavithran D, Al-Karaki J N, Shaalan K. Edge-based blockchain architecture for event-driven IoT using hierarchical identity based encryption. Information Processing & Management, 2021, 58(3), 102528.
17. Badar H M S, Qadri S, Shamshad S, et al. An identity based authentication protocol for smart grid environment using physical uncloneable function. IEEE Transactions on Smart Grid, 2021, 12(5), 4426-4434.
18. Samiullah F, Gan M L, Akleylek S, et al. Group key management in internet of things: A systematic literature review. IEEE Access, 2023, 11,77464-77491.
19. Fiat A, Naor M. Broadcast encryption. Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference, 1994, 480-491.
20. Boneh D, Gentry C, Waters B. Collusion resistant broadcast encryption with short ciphertexts and private keys. Annual international cryptology conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, 258-275.
21. Lewko A, Sahai A, Waters B. Revocation systems with very small private keys. IEEE Symposium on Security and Privacy, 2010, 273-285.

 doi:10.20944/preprints202506.1071.v1



22. Chen L, Li J, Zhang Y. Adaptively secure efficient broadcast encryption with constant-size secret key and ciphertext. Soft Computing, 2020, 24, 4589-4606.
23. Acharya K. Secure and efficient public key multi-channel broadcast encryption schemes. Journal of Information Security and Applications, 2020, 51, 102436.
24. Sravan Kumar G, Sri Krishna A. Privacy sustaining constant length ciphertext-policy attribute-based broadcast encryption. Soft Computing and Signal Processing, 2019, 313-324.
25. Wazid M, Das A K, Kumar N, et al. Secure three-factor user authentication scheme for renewable-energy-based smart grid environment. IEEE Transactions on Industrial Informatics, 2017, 13(6), 3144-3153.
26. Mahmood K, Chaudhry S A, Naqvi H, et al. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. Future Generation Computer Systems, 2018, 81, 557-565.
27. Kumar P, Gurtov A, Sain M, et al. Lightweight authentication and key agreement for smart metering in smart energy networks. IEEE Transactions on Smart Grid, 2018, 10(4), 4349-4359.
28. Wang J, Wu L, Choo K K R, et al. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. IEEE Transactions on Industrial Informatics, 2019, 16(3), 1984-1992.
29. Lin H Y, Hsieh M Y. A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles. Connection Science, 2022, 34(1), 1089-1118.
30. Tan H, Zheng W, Guan Y, et al. A privacy-preserving attribute-based authenticated key management scheme for accountable vehicular communications. IEEE Transactions on Vehicular Technology, 2022, 72(3), 3622-3635.