Research Article

EPPDC: An Efficient Privacy-Preserving Scheme for Data Collection in Smart Grid

Jie Chen, Junping Shi, and Yueyu Zhang

State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an 710071, China

Correspondence should be addressed to Jie Chen; jchen@mail.xidian.edu.cn

Received 25 July 2014; Revised 23 October 2014; Accepted 18 November 2014

Academic Editor: Georgios Kambourakis

Copyright © 2015 Jie Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Different from the traditional grid, smart grid builds a real-time connection network between the user and the grid company by smart terminals, which can achieve bidirectional data transmission and information control. In smart grid, the smart meters send various information to the power generators and substations. Frequent data collection meets real-time management, but it tends to raise privacy concerns from the users about privacy information leakage. Based on the blind signature and the key distribution scheme, an efficient and privacy-preserving data collection (EPPDC) scheme is proposed for smart grid to cope with the above problems. In EPPDC scheme, the users’ data information is transmitted to the local aggregator by building gateway with privacy preserving. In addition, the security analysis indicates that EPPDC scheme not only can resist replay attack, but also has source authentication and data integrity, confidentiality, unforgeability, nonrepudiation, and evolution of shared keys. Furthermore, performance analysis shows that EPPDC scheme has less computation cost than existing scheme.

1. Introduction

Smart grid is one of the most important public infrastructures for smart cities. It builds a real-time connection network between the user and the grid company by smart terminals and supports bidirectional data transmission and information control. Depending on it, smart cities could ensure resilient supply and delivery of energy, which help smart cities to fulfill many enhanced and innovative functions and even more efficiencies compared with traditional cities. Furthermore, smart grid can also facilitate coordination among people who are responsible for public safety and the public, such as urban officials and infrastructure operators [1]. The advantage of smart grid is attracting more and more attention and research in smart city projects. Currently, around one-third of the smart city projects are primarily focused on smart grid or other energy innovations. Almost half of smart city strategies include energy-focused projects [2].

Smart grid can support the bidirectional information flow between the power consumer and the utility provider [3]. This two-way interaction allows electricity to be generated in real-time based on consumers’ demands and power requests. As an important technique in smart cities, the advanced metering will affect not only the power sector but also other utilities such as gas/heating and water which will make use of smart meters to read and process consumption data remotely [4]. In smart cities, each house may contain a smart meter connecting to all electric appliances in the house. The utility transmits requests and commands to the smart meters and gathers and analyzes power usage data responded by each smart meter. If being leaked, that information will indicate not only the amount of energy consumed by each user but also behaviors like when they are at home, at work, or traveling [5]. Furthermore, it is possible to infer what types of home appliances are used by attackers who compromise users’ home area networks. If a criminal or malicious attacker can determine when a user is not at home, they may break into his/her house at such a time. And energy information can support burglars or provide business intelligence to competitors [6]. By this information, the users’ habits or lifestyles can be tracked. And a series of problems arises in case of information leakage. Thus, authentication and user privacy preservation are two important security issues on the information flow in smart grid.
Thus, there is need to design schemes which can achieve data transmission between smart meters and smart grid provider with privacy preserving. This paper just studies the privacy-preserving data collection scheme for smart grid.

2. Related Works

The proposed privacy-preserving schemes about smart grid are mainly constructed by two kinds of cryptographic tools, homomorphic encryption [7–9] and signcryption [10, 11]. By homomorphic encryption, smart meters (SMs) encrypt the messages and send them to gateway (BGW), but gateway cannot get any users’ messages without the system private key. Then, gateway signs encrypted messages and sends them to control center (CC). Based on the property of homomorphic encryption, control center can make use of the system private key to recover every user’s messages.

Secure data aggregation schemes in smart grid have been investigated by several researchers. Lu et al. proposed a privacy-preserving aggregation scheme [7], which is based on the homomorphic Paillier cryptosystem. But in [7], it is assumed that the session keys between SM and BGW are unchanged. Once an adversary $\mathcal{D}$ compromises the session keys, $\mathcal{D}$ can decrypt any previous response message. Based on [7], Li et al. proposed a privacy-preserving demand and response scheme with adaptive key evolution [9]. Both [7, 9] make use of homomorphic encryption to achieve privacy preserving, and they can meet aggregation for some data. In addition, several researchers focused on privacy-preserving aggregation in different conditions by using multiparty computation [12, 13], differential privacy [14], and the aggregated pseudostatus variation [15]. As signcryption based schemes, they can complete digital signature and encryption for a message in one time. In particular, SMs signcrypt the messages and send them to gateway. Gateway cannot get any users’ messages from the encrypted messages. Then, gateway signs and sends encrypted messages to control center. Control center can recover every user’s messages by the shared key between CC and SMs. In [10], an identity-based signcryption scheme for smart grid was proposed. But in [10], the management of pseudonymous ID is a problem. Reference [11] adopted the pseudonym technology to achieve the user identity anonymity and adopted the signcryption to complete digital signature and encryption in one time in smart grid.

However, the smart grid needs not only to protect users’ sensitive information but also to meet their demands for personalized data application with multilevel and multigranularity. Thus many users’ messages cannot be aggregated, which should be sent to control center detail by detail. And existing homomorphic encryption to achieve privacy preserving is based on the computational expensive operations [7, 9], which may not be desirable for smart grids with limited resources in terms of both bandwidth and computation. And the existing signcryption based schemes in smart grid [10, 11] cannot meet forward secrecy.

This paper proposes an efficient privacy-preserving data collection scheme for smart grid, which is based on the blind signature and the key distribution scheme. In this scheme, users’ data information is transmitted to the local aggregator via gateway, while gateway cannot get any users’ messages. Moreover, this scheme can achieve forward secrecy of SM’s session key, and evolution of SM’s private keys.

The remainder of paper is organized as follows. Section 3 introduces models and design goal. Section 4 describes preliminaries. Section 5 presents the proposed EPPDC scheme. Section 6 shows the security analysis and the computation overhead of the scheme in this paper, respectively. Finally, Section 7 makes a conclusion.

3. Models and Design Goal

In this section, we give the system model, security model, and the design goal.

3.1. System Model. As shown in Figure 1, smart grid is divided into a number of hierarchical networks, which is comprised of control center (CC), district area network (DAN), building area network (BAN), and home area network (HAN). The CC covers $n_1$ DANs. For the sake of simplicity, we assume that each DAN comprises $n_2$ BANs and each BAN comprises $n_3$ HANs. Each HAN is assigned a smart meter (SM) enabling an automated, bidirectional communication between the CC and the HAN users. Meantime, each BAN is equipped with a gateway (BGW) and each DAN is equipped with a local aggregator (LAG). And each SM can directly communicate with LAG via the BGW.

In this paper, the system model of smart grid contains 5 parties, including trusted authority (TA), central aggregator (CAG), LAG, BGW, and SM. LAG is the entity that can directly communicate with CAG on behalf of those geographically dispersed HANs. TA belongs to some independent organizations like Regional Transmission Organizations (RTO) or Independent System Operators (ISO). TA does the system initiation, such as generating public system parameters and assigning private key for each entity.

Then we give a partial relationship for smart grid in China, which is shown in Figure 2. The provincial operator is viewed as central aggregator CAG, and municipal operator is viewed as LAG. For example, there is focus on the Northwest China. It has a CAG located in Shaanxi and multiple LAGs dispersed in Xi’an, Hanzhong, Yulin, and other towns. And ISO, Northwest China, plays the role of TA. The provincial operator is responsible for generating and transmitting parameters of CAG, predicting flexible power demand and managing renewable generation in its province. Generally, the provincial operator can refer to municipal electricity demand curve to make power supply plan and generation dispatching plan with a day ahead. The municipal operator is responsible for generating networks parameters and aggregating the power demand. At present in China, the communication between the provincial operator and the municipal operator is used by fiber optic link which is assumed to be safe. The municipal operator can refer to load curve. According to preferred load curve, time of use (TOU) prices, and customers demand, each BAN makes bids
in the electricity market. Both municipal and BAN operator need real-time communication and data management. Usually, wireless communication is used to transfer data between BGW and SM.

3.2. Security Model. In our security model, CAG and LAG are trusted by all parties and are infeasible for any adversary to compromise. BGW can comply with the scheme but with diligent curiosity. Thus BGW possibly gets the user’s privacy information in the process of implementing the scheme. We consider the following security goals.

(1) Confidentiality: the messages sent to LAG from SM should be confidential; that is, if an adversary \( A \) captures the messages, it cannot identify the encrypted messages.

(2) Authenticity and data integrity: BGW and HAN users should be authenticated by LAG and BGW each other, respectively. Meanwhile, if an adversary \( A \) modifies the messages, the malicious operations can be detected.

(3) Privacy preservation: the users’ electricity information should not be disclosed to the undesirable entities. Privacy preservation should meet the anonymous authentication and data encryption, which make attacker not able to get any information from any of the users. In smart grid, if an adversary \( A \) hacks into the database of BGWs, it cannot determine the contents of ciphertexts. In order to protect the users’ privacy, even BGW cannot determine the detailed electrical information to certain users.

(4) Evolution of users’ private keys: the evolution of users’ private keys should be achieved. If an adversary \( A \) compromises any previous private key of a HAN user, \( A \) cannot use it currently or in the future.

3.3. Design Goal. Under the above models, our design goal is to develop an efficient privacy-preserving scheme for data collection in smart grid. Specifically, the following two desirable objectives will be achieved.

(1) The proposed privacy-preserving scheme should achieve the message source authentication, data integrity, and the confidentiality of the messages.

(2) The proposed scheme should be cost-effective in terms of computation and communication overheads.

4. Preliminaries

In this section, we review bilinear pairings, hash function and HMAC [16], group key distribution scheme [17], and Nyberg-Rueppel blind signature technology [18], which will serve as the basis of the proposed scheme.

4.1. Bilinear Pairings. Let \( G_1 \) be a cyclic additive group of prime order \( q \) and let \( G_T \) be a cyclic multiplicative group of
the same order. A map $e: G_1 \times G_1 \rightarrow G_T$ is called a bilinear map if it satisfies the following properties:

1. Bilinearity: $e(Q, W + Z) = e(Q, W)e(Q, Z)$ and $e(Q + W, Z) = e(Q, Z)e(W, Z)$, for all $Q, W, Z \in G_1$;
2. Nondegeneracy: there exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$;
3. Computability: there is an efficient algorithm to compute $e(P, Q)$ for $P, Q \in G_1$.

4.2. Hash Function and HMAC. A one-way hash function $h(\cdot)$ is said to be secure if the following properties are satisfied:

1. $h(\cdot)$ can take a message of arbitrary length as input and produce a message digest of a fixed-length output.
2. Given $x$, it is easy to compute $h(x) = y$. However, it is hard to compute $h^{-1}(y) = x$ given $y$.
3. Given $x$, it is computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$.

Hash-based message authentication code (HMAC) is a specific construction for computing a message authentication code (MAC) using a cryptographic hash function in combination with a secret key. Both data integrity and authenticity of a message can be achieved using such a technique. Due to the property of hash functions, an HMAC value can be computed in a much shorter time than a traditional digital signature. In this paper, we denote the HMAC value on message $M$ is HMAC$_K(M)$ using the secret key $K$.

4.3. Group Key Distribution Scheme. The purpose of group key distribution is to distribute keys to selected group members so that each of the selected group members shares a distinct personal key with the group manager, but the other group members cannot get any information of the keys. In [17], the group manager broadcasted a message, and all these selected group members could derive their keys from the message. The approach of [17] chose a random $t$-degree polynomial $f(x)$ from $F_q[x]$ and selected $f(i)$ for each group member $U_j$ as the shared person key. The group manager constructed a single broadcast polynomial $w(x)$ such that, for a selected group member $U_j$, $f(i)$ could be recovered from the knowledge of $w(x)$ and the personal secret $S_j$. But for any revoked group member, $U_j'$, $f(i')$ could not be determined from $w(x)$ and $S_j$.

In [17], $w(x) = g(x)f(x) + h(x)$ was constructed by $f(x)$ with the help of a revocation polynomial $g(x)$ and

![Figure 2: An envisioned relationship diagram for smart grid in China.](image-url)
a masking polynomial \( h(x) \). The revocation polynomial \( g(x) \) was constructed in such a way that \( g(i) \neq 0 \) for any selected group member \( U_i \), but \( g(i') = 0 \) for any revoked group member \( U_{i'} \). During setup phase, each group member \( U_i \) had its own personal secret \( S_i = \{ h(i) \} \), which might be distributed by the group manager through the secure communication channel between each group member and the group manager. Thus, for any selected group member \( U_i \), new personal key \( f(i) \) could be computed by \( f(i) = \{ w(i) - h(i) \} / g(i) \), but for any revoked group member \( U_{i'} \), new personal key could not be computed because \( g(i') = 0 \). Specific steps were as follows.

1. Setup: the group manager randomly picked a \( 2t \)-degree masking polynomial, \( h(x) = h_0 + h_1 x + \cdots + h_\omega x^\omega \), from \( F_q[x] \). Each group member \( U_i \) got the personal secret \( S_i = \{ h(i) \} \) from the group manager.

2. Broadcast: given a set of revoked group members \( R = \{ r_1, r_2, \ldots, r_\ell \} \) (\( |R| \leq t \)), the group manager distributed the shares of \( t \)-degree polynomial \( f(x) \) to nonrevoked group members via the following broadcast message: \( B = \{ R \} \cup \{ w(x) = g(x) f(x) + h(x) \} \), where the revocation polynomial \( g(x) = (x - r_1)(x - r_2) \cdots (x - r_\ell) \).

3. Personal key recovery: if any nonrevoked group member \( U_i \) received such a broadcast message, it evaluated the polynomial \( w(x) \) at point \( i \) and got \( w(i) = g(i) f(i) + h(i) \). Because \( U_i \) knew \( h(i) \) and \( g(i) \neq 0 \), it could compute the new personal key \( f(i) = \{ w(i) - h(i) \} / g(i) \).

4.4. Nyberg-Rueppel Blind Signature. Blind signatures enable users to obtain valid signatures on a message without revealing its content to the signer. Nyberg-Rueppel blind signature scheme was proposed by Camenisch et al., which was based on the discrete logarithm problem [18]. The scheme had three parts as follows.

1. Setup system parameters: the system parameters consisted of a prime \( p \), a prime factor \( q \) of \( p - 1 \), and an element \( g \in Z_q^* \) of order \( q \). The signer’s private key was a random element \( x \in Z_q^* \), while the corresponding public key was \( y = g^x \).

2. Sign: Bob could obtain a valid signature on a message \( m \) from Alice without revealing its content to Alice.

a. Alice randomly selected \( k' \in Z_q^* \), computed \( r' = g^{k'} \mod p \), and sent \( r' \) to Bob.

b. Bob selected \( \alpha, \beta \in Z_q^* \) at random and computed \( r = m g^{r' \beta} \mod p \) and \( m' = r^{s'} \mod p \). Bob checked whether it was satisfied with \( m' \in Z_q^* \). If this was not the case, a new \((\alpha, \beta)\) would be chosen until it was satisfied with \( m' \in Z_q^* \). Then, Bob sent \( m' \) to Alice.

c. Alice computed \( s' = m' x + k' \mod q \) and sent \( s' \) to Bob.

(d) Bob computed \( s = s' \beta + \alpha \mod q \). \((r, s)\) was the signature of Alice on message \( m \).

3. Verify: anyone could verify the validity of the signature \((r, s)\) on message \( m \) by

\[
g^{-1} y r = mg^{-s - \alpha x + r' k' \beta + \alpha} = mg^{-m' x \beta - k' \beta x + r' k' \beta} = m \mod p .
\]

5. EPPDC Scheme

Based on the blind signature and the key distribution scheme, this section gives the EPPDC scheme for smart grid. This scheme can achieve that the users’ data is transmitted to the LAG via BGW with privacy preserving. In this section, we propose the scheme, which consists of five phases: system initialization, certificate issuing, user registration, data collection, and key evolution.

5.1. System Initialization. We assume that TA will initialize the whole system. TA chooses the following:

1. primes \( p \) and \( q \) such that \( q \mid p - 1 \), \( q \geq 2^{140} \) and \( p \geq 2^{512} \);
2. an element \( g \in Z_p^* \) with order \( q \); that is, \( g^q = 1 \mod p \), and \( q \neq 1 \);
3. a one-way hash function \( h(0, 1)^* \rightarrow (0, 1)^* \);
4. a security parameter \( \lambda \in N \) and bilinear group \((G, G_T)\) with prime order \( p > 2^\lambda \), which satisfies with \( g \) being a generator of \( G \);
5. a random number \( s \in Z_q^* \) as TAs private key so that \( SK_{TA} = s \).

Then, TA computes its public key \( PK_{TA} = g^s \) and publishes the tuple \((p, q, g, h, PK_{TA})\) as the system parameters.

5.2. Certificate Issuing. During this phase, TA verifies the identity and issues the certificate for every entity. These entities include all the CAGs, LAGs, and BGWs. As an example, TA issues the certificate for a certain LAG, as follows.

1. TA chooses a random number \( SK_{LAG_i} \in Z_q^* \) as the LAG’s private key and computes the LAG’s public key \( PK_{LAG_i} = g^{SK_{LAG_i}} \mod p \).
2. TA generates the signature \( \sigma_{TA,LAG_i} \), where \( \sigma_{TA,LAG_i} = Sig_{SK_{TA}}(PK_{LAG_i}) \) is a signature on \( PK_{LAG_i} \) using TAs private key \( SK_{TA} \).
3. TA delivers \( SK_{LAG_i} \) and \( Cert_{TA,LAG_i} \) to LAG, where \( Cert_{LAG_i} = (PK_{LAG_i}, \sigma_{TA,LAG_i}) \). The delivery of \( SK_{LAG_i} \) must be via a secure channel, such as a Secure Socket Layer.

5.3. User Registration. Before accessing smart grid, every SM needs to get a certificate from TA and register in certain LAG which SM belongs to. Assume \( SM_{LAG_i} \) indicates that a certain SM belongs to LAG. In Figure 3, an example of user registration for \( SM_{LAG_i} \) is as follows.
Figure 3: The flow chart of SM\textsubscript{i−LAG\textsubscript{g}} registration.

1. After TA verifies the identity of SM\textsubscript{i−LAG\textsubscript{g}}, TA delivers the SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} and certificate Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} to SM\textsubscript{i−LAG\textsubscript{g}}, which is the same as the process in Section 5.2.

2. SM\textsubscript{i−LAG\textsubscript{g}} encrypts the message Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} || T || Sig\textsubscript{SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}}(Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} || T) by PK\textsubscript{LAG\textsubscript{i}} and sends it to LAG\textsubscript{i}, where T is the current timestamp.

3. LAG\textsubscript{i} decrypts the received message by its private key SK\textsubscript{LAG\textsubscript{i}}. LAG\textsubscript{i} verifies the validity of timestamp T, Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} and Sig\textsubscript{SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}}(Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}} || T). If all those are available, go to step (4). Otherwise, go back to Step (2).

4. LAG\textsubscript{i} chooses a random number S_{LAG\textsubscript{i−SM\textsubscript{i}}} ∈ Z^{*}\textsubscript{q} and computes P_{LAG\textsubscript{i−SM\textsubscript{i}}} = g^{S_{LAG\textsubscript{i−SM\textsubscript{i}}}}, K_{LAG\textsubscript{i−SM\textsubscript{i}}} = (PK_{SM\textsubscript{i−LAG\textsubscript{i}}}, S_{LAG\textsubscript{i−SM\textsubscript{i}}}, H_{LAG\textsubscript{i−SM\textsubscript{i}}}) – HMAC\_K_{LAG\textsubscript{i−SM\textsubscript{i}}}(P_{LAG\textsubscript{i−SM\textsubscript{i}}}). Then, LAG\textsubscript{i} sends P_{LAG\textsubscript{i−SM\textsubscript{i}}} and H_{LAG\textsubscript{i−SM\textsubscript{i}}} to SM\textsubscript{i−LAG\textsubscript{g}}.

5. SM\textsubscript{i−LAG\textsubscript{g}} can get noninteractively shared key K_{LAG\textsubscript{i−SM\textsubscript{i}}} = (P_{LAG\textsubscript{i−SM\textsubscript{i}}})^{SK_{SM\textsubscript{i−LAG\textsubscript{g}}}} and can verify the correctness of H_{LAG\textsubscript{i−SM\textsubscript{i}}}. If H_{LAG\textsubscript{i−SM\textsubscript{i}}} is consistent, go to Step (6). Otherwise, go back to Step (2).

6. SM\textsubscript{i−LAG\textsubscript{g}} sends P_{LAG\textsubscript{i−SM\textsubscript{i}}} || TS || Sig\textsubscript{SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}}(P_{LAG\textsubscript{i−SM\textsubscript{i}}} || TS) to LAG\textsubscript{i}, where TS is the current timestamp.

7. LAG\textsubscript{i} verifies the validity of signature in Step (6). If it is available, go to Step (8). Otherwise, go back to Step (4).

8. LAG\textsubscript{i} stores the Cert\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}, S_{LAG\textsubscript{i−SM\textsubscript{i}}}, and K_{LAG\textsubscript{i−SM\textsubscript{i}}} in its database. Then LAG\textsubscript{i} sends the permit defined as PK_{SM\textsubscript{i−LAG\textsubscript{g}}} || TS\_permit || d || Sig\textsubscript{SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}}((PK_{SM\textsubscript{i−LAG\textsubscript{g}}} || TS\_permit || d) to SM\textsubscript{i−LAG\textsubscript{g}}, where TS\_permit is the current timestamp and d is the expiry length of time.

5.4 Data Collection. When a certain LAG\textsubscript{j} needs to make statistical analysis and collect energy information in its DAN, LAG\textsubscript{j} broadcasts the data collection command to its subordinate BGWs. Similarly, each BGW will broadcast the data collection command to its subordinate SMs. As an example, the process of data collection from SM\textsubscript{i−LAG\textsubscript{g}} to LAG\textsubscript{i} is as follows.

1. Each BGW Collects Eligible SMs’ Permits

(1) After receiving the data collection command from BGW\textsubscript{j−LAG\textsubscript{i}}, SM\textsubscript{j−LAG\textsubscript{i}} chooses a random number u_{SM\textsubscript{i−LAG\textsubscript{g}}} ∈ Z^{*}\textsubscript{q} and computes K_{1u_{SM\textsubscript{i−LAG\textsubscript{g}}}} = g^{u_{SM\textsubscript{i−LAG\textsubscript{g}}}}. Here BGW\textsubscript{j−LAG\textsubscript{i}} is a certain BGW who is LAG\textsubscript{i}’s subordinate and SM\textsubscript{j−LAG\textsubscript{i}}’s superior. Then, SM\textsubscript{j−LAG\textsubscript{i}} encrypts the message C_{1u_{SM\textsubscript{i−LAG\textsubscript{g}}}} = permit || K_{1u_{SM\textsubscript{i−LAG\textsubscript{g}}}} || T_{1} || Sig\textsubscript{SK\textsubscript{SM\textsubscript{i−LAG\textsubscript{g}}}}(permit || K_{1u_{SM\textsubscript{i−LAG\textsubscript{g}}}} || T_{1}) by
PK_{BGW_j-LAG_i} and sends it to BGW_{j-LAG_i}, where T_1 is the current timestamp.

(2) BGW_{j-LAG_i} verifies the validity of C_{1_{BGW_i-LAG_i}} using Algorithm 1. For eligible SM_{j-LAG_i}, BGW_{j-LAG_i} chooses two random numbers u_{BGW_i-LAG_i} \in \mathbb{Z}_q^* and computes r_{BGW_i-LAG_i}' \equiv \text{SK}_{BGW_i-LAG_i} + K_{BGW_i-LAG_i} \pmod{p}, K_{SM-BGW_i-LAG_i}, where r_{BGW_i-LAG_i}' = g^{r_{BGW_i-LAG_i}} \pmod{p}, K_{SM-BGW_i-LAG_i} = g^{u_{BGW_i-LAG_i}} \pmod{p},

and K_{SM-BGW_i-LAG_i} = u_{BGW_i-LAG_i} \pmod{p}.

(3) BGW_{j-LAG_i} sends the message C_{2_{SM-LAG_i}} = r_{BGW_i-LAG_i}' || K_{SM-BGW_i-LAG_i} || T_2 || \text{HMAC}_{K_{SM-BGW_i-LAG_i}}(r_{BGW_i-LAG_i}' || K_{SM-BGW_i-LAG_i} || T_2) \mod(p).

(4) After receiving C_{2_{SM-LAG_i}}, SM_{j-LAG_i} can get noninteractively shared key K_{SM-BGW_i-LAG_i} = K_{SM-BGW_i-LAG_i} \pmod{p} and compute \text{HMAC}_{K_{SM-BGW_i-LAG_i}}(r_{BGW_i-LAG_i}' || K_{SM-BGW_i-LAG_i} || T_2).

(5) If the value of HMAC is consistent, go to Step (5). Otherwise, go back to Step (1).

(6) SM_{j-LAG_i} sends permit || T_3 || \text{HMAC}_{K_{SM-BGW_i-LAG_i}}(permit || T_3) to BGW_{j-LAG_i}, where T_3 is the current timestamp.

(7) When T_3 - T_2 < \Delta T, BGW_{j-LAG_i} verifies the correctness of received messages in Step (5) by HMAC, where \Delta T is the limit for time difference. Then, BGW_{j-LAG_i} collects all eligible SMs’ permits and sends them to LAG_i.

(II) Generate the Shared Blind Factors between LAG and Every SM

(1) After receiving all the permits from BGW_{j-LAG_i}, LAG_i confirms the total of permits as N_{BGW_i-LAG_i}. Then LAG_i finds the corresponding K_{LAG_i-SM_i} in its database, where K_{LAG_i-SM_i} is the shared key of LAG_i with SM_{j-LAG_i}. And LAG_i generates the message \{B_{BGW_i-LAG_i}, K_{BGW_i-LAG_i}\} by using Algorithm 2.

(2) LAG_i sends message \{B_{BGW_i-LAG_i}, K_{BGW_i-LAG_i}\} to BGW_{j-LAG_i}.

(3) After receiving the message from LAG_i, BGW_{j-LAG_i} stores K_{SM-BGW_i-LAG_i} in its database and broadcasts B_{BGW_i-LAG_i} to its every subordinate SM. If SM’s permit is eligible, SM can recover K_{LAG_i-BGW_i-LAG_i} and K_{LAG_i-BGW_i-LAG_i}^2. As an example, SM_{j-LAG_i} can get K_{LAG_i-BGW_i-LAG_i} = g^{r_{BGW_i-LAG_i}}(K_{LAG_i-SM_i}) + K_{LAG_i-BGW_i-LAG_i} and K_{LAG_i-BGW_i-LAG_i}^2 = g^{r_{BGW_i-LAG_i}}(K_{LAG_i-SM_i}) + K_{LAG_i-BGW_i-LAG_i}, respectively.

In the following, SM_{j-LAG_i} is an example of data collection, which is the same as other SMs.

(4) SM_{j-LAG_i} computes two blind factors \alpha_{i-LAG_i-SM_i} = H(K_{LAG_i-SM_i}) \pmod{p}, \beta_{i-LAG_i-SM_i} = H(K_{LAG_i-SM_i}) \pmod{p}, where H is a one-way hash function.

(III) LAG Collects Data from SM via BGW

(1) SM_{j-LAG_i} reads the current electricity information m_{i-LAG_i} and computes r_{i-LAG_i} = m_{i-LAG_i}g^{\alpha_{LAG_i-SM_i}}r_{BGW_i-LAG_i} \pmod{p}, m_{i-LAG_i} = r_{i-LAG_i}K_{i-LAG_i}^{-1} \pmod{p}. SM_{j-LAG_i} sends C_{3_{SM_i-LAG_i}} = m_{i-LAG_i} || T_4 || \text{Sig}_{SK_{SM_i-LAG_i}}(m_{i-LAG_i} || T_4) to BGW_{j-LAG_i}, where T_4 is the current timestamp.

(2) BGW_{j-LAG_i} verifies the validity of signature in Step (1). If the signature is valid, go to Step (3). Otherwise, BGW_{j-LAG_i} requires SM_{j-LAG_i} resending the message as in Step (1) within the valid time range.

(3) BGW_{j-LAG_i} chooses a random number r_{nBGW_i-LAG_i} \in \mathbb{Z}_q^* and computes \alpha_{BGW_i-LAG_i} = g^{r_{BGW_i-LAG_i}} \pmod{p}, \beta_{BGW_i-LAG_i} = g^{\alpha_{BGW_i-LAG_i}} \pmod{p}.

BGW_{j-LAG_i} signs the message m_{i-LAG_i} with \text{Sig}_{SK_{BGW_i-LAG_i}}(m_{i-LAG_i} || T_4) and \text{HMAC}_{K_{SM_i-BGW_i-LAG_i}}(C_{3_{SM_i-LAG_i}}) to LAG_i.

(5) LAG_i can get SM_{j-LAG_i}’s current electricity information m_{i-LAG_i} using Algorithm 3.

5.5. Key Evolution. The SM’s permit is only valid from T_{perm} to T_{perm} + d. After T_{perm} + d, the permit is automatically revoked if SM does not apply for a new permit. Assuming d’ is the corresponding number of days for d, here we extend the shared key K_{LAG_i-SM_i} to d’ dimensional vector (K_{1_LAG_i-SM_i}, K_{2_LAG_i-SM_i}, \ldots, K_{d_LAG_i-SM_i}). For d’ dimensional vector K_{LAG_i-SM_i}, K_1 = K_{LAG_i-SM_i}, K_2 = H(K_1), and K_3 = H(K_2), \ldots, K_{d'} = H(K_{d-1}), where H is a one-way hash function. Note that the extension does not influence the previous EPPDC scheme. Then, both SM and LAG can deduce d’ dimensional shared key K_{LAG_i-SM_i} = (K_1, K_2, \ldots, K_{d'}) within the terms of permit.

Here we assume that every shared key is valid for one day. d’ dimensional shared key is valid for d’ days during the validity of permit. Accordingly, both SM and LAG can confirm the inadray shared key K_{LAG_i-SM_i} from d’ dimensional key K_{LAG_i-SM_i}.
Require: EncPKBGW_{j-LAG_i} [C_{1SM_i-LAG_i}]
(1) Decrypt EncPKBGW_{j-LAG_i} [C_{1SM_i-LAG_i}] by BGW_{j-LAG_i}’s private key SKBGW_{j-LAG_i}.
(2) Verify $T_1 = TS_{permit} < d$, $d$ is the expiry length of time.
(3) if $T_1$ is valid then
    Verify $\text{Sig}_{\text{SM}_i-LAG_i} \left( \text{permit} \parallel K_{1SM_i-LAG_i} \parallel T_1 \right)$ by PK_{SM_i-LAG_i}.
    if $\text{Sig}_{\text{SM}_i-LAG_i} \left( \text{permit} \parallel K_{1SM_i-LAG_i} \parallel T_1 \right)$ is valid then
        Verify $\text{Sig}_{\text{LAG}_i} \left( \text{permit} \parallel K_{1SM_i-LAG_i} \parallel T_1 \right)$ by PK_{LAG_i}.
        if $\text{Sig}_{\text{LAG}_i} \left( \text{permit} \parallel K_{1SM_i-LAG_i} \parallel T_1 \right)$ is valid then
            accept $C_{1SM_i-LAG_i}$.
        end if
    end if
end if

Algorithm 1: The process of verifying $C_{1SM_i-LAG_i}$.

Require: $g$, $p$, $q$, $K_{LAG_i-SM_1}$, $K_{LAG_i-SM_2}$, $K_{LAG_i-SM_3}$, $K_{LAG_i-SM_4}$
(1) Construct function $g_{BGW_{j-LAG_i}} (x) = (x - K_{LAG_i-SM_1}) \cdot (x - K_{LAG_i-SM_2}) \cdot \ldots \cdot (x - K_{LAG_i-SM_4})$,
which from $K_{LAG_i-SM_1}$ to $K_{LAG_i-SM_4}$ indicates $N_{BGW_{j-LAG_i}}$ shared keys of LAG with SMs which permit past validation.
(2) Select random numbers $K_{1LAG_i-BGW_{j-LAG_i}} \in Z_q^*$ and $K_{2LAG_i-BGW_{j-LAG_i}} \in Z_q^*$.
(3) Generate $B_{BGW_{j-LAG_i}} = \{g_{BGW_{j-LAG_i}} (x) + K_{1LAG_i-BGW_{j-LAG_i}} \cdot g_{BGW_{j-LAG_i}} (x) + K_{2LAG_i-BGW_{j-LAG_i}} \}$.
(4) Select a random number $u'_{BGW_{j-LAG_i}} \in Z_q^*$.
(5) Compute $K_{1BGW_{j-LAG_i}} = g_{BGW_{j-LAG_i}} (u'_{BGW_{j-LAG_i}}) \pmod{p}$.
Return $\{B_{BGW_{j-LAG_i}}, K_{1BGW_{j-LAG_i}} \}$

Algorithm 2: Message generation algorithm for LAG.

(1) Verify $T_2 - T_5 < \Delta T$, where $\Delta T$ is the limit for time difference.
(2) if $T_2 - T_5 < \Delta T$ then
    compute $K_{SM_i-BGW_{j-LAG_i}}^{r_i} = K_{BGW_{j-LAG_i}} \cdot u_{BGW_{j-LAG_i}}^{r_i} \pmod{p}$.
(3) Verify the validity of HMAC of $C_{1SM_i-LAG_i}$.
(4) if HMAC is valid then
    find $K_{LAG_i-SM_1}$ in its database according to PK_{SM_i-LAG_i}.
(5) Computes $\alpha_{LAG_i-SM_1} = H(K_{1LAG_i-BGW_{j-LAG_i}} \parallel K_{LAG_i-SM_1}), \beta_{LAG_i-SM_1} = H(K_{2LAG_i-BGW_{j-LAG_i}} \parallel K_{LAG_i-SM_1})$.
(6) Get $s_{SM_i-BGW_{j-LAG_i}} = s_{SM_i-BGW_{j-LAG_i}}^r + \alpha_{LAG_i-SM_1}, r_{LAG_i} = m_{LAG_i}^r, m_{LAG_i} = r_{LAG_i} \cdot g_{\text{SM}_i-BGW_{j-LAG_i}}^{r_{LAG_i}} \pmod{p}$,
    if $\alpha_{LAG_i-SM_1} = m_{LAG_i}^r, \beta_{LAG_i-SM_1}^r \pmod{p}$.
(7) Judge $g_{\text{SM}_i-BGW_{j-LAG_i}} \cdot PK_{BGW_{j-LAG_i}} \cdot r_{LAG_i} \cdot m_{LAG_i} \pmod{p}$.
    Record the message $m_{LAG_i}$ and signature $(s_{SM_i-BGW_{j-LAG_i}}, r_{LAG_i})$ in the corresponding database.
(8) end if
(9) end if
(10) end if

Algorithm 3: The process of recovering message for LAG.
Every SM only stores the intraday shared key $K_{\text{LAG-SM}}$ within $permit$ validity period. The day before the expiration of $permit$, SM applies for registration in LAG again. For eligible SM, LAG will issue the new $permit$ and form the new shared key $K_{\text{LAG-SM}}$ to SM. When previous $permit$ has expired, SM deletes previous $permit$ and $K_{\text{LAG-SM}}$. Thus, the shared key $K_{\text{LAG-SM}}$ evolution is achieved.

6. Security Analysis and Computation Overhead

In this section, we analyze the security properties and the computation of the EPPDC scheme.

6.1. Security Analysis. EPPDC scheme can achieve data collection from SM to LAG via BGW. And EPPDC scheme not only can resist replay attack, but also has source authentication and data integrity, confidentiality, unforgeability, nonrepudiation, and evolution of shared keys.

Property 1 (correctness). In EPPDC scheme, LAG can verify the blind signature of BGW and recover the message sent by SM.

Proof. During Stage III of data collection in EPPDC scheme, $BGW_{j-LAG}$ sends the $(r'_{BGW_{j-LAG}}, s_{SM-BGW_{j-LAG}})$ message $m'_{j-LAG}$, and $PK_{SM-LAG}$ to LAG. LAG can find the corresponding $K_{\text{LAG-SM}}$ in its database according to $PK_{SM-LAG}$ and the current timestamp. Then, LAG recovers the current electricity information $m_{j-LAG}$ using Algorithm 3, which is proved by

\[
r_{j-LAG} \cdot g^{a_{SM-LAG}, r'_{BGW_{j-LAG}}} \cdot g^{-\alpha_{SM-LAG}, \beta_{SM-LAG}} \pmod{p} = \left( m_{j-LAG} \cdot g^{a_{SM-LAG}, r'_{BGW_{j-LAG}}} \cdot g^{-\alpha_{SM-LAG}, \beta_{SM-LAG}} \right) \pmod{p} = m_{j-LAG}. \tag{2}
\]

The $BGW_{j-LAG}$'s signature on $m_{j-LAG}$ is $(s_{SM-BGW_{j-LAG}}, r_{j-LAG})$, which can be proved by

\[
g^{s_{SM-BGW_{j-LAG}}} \cdot (PK_{BGW_{j-LAG}})^{-a_{SM-LAG}} \cdot r_{j-LAG} \pmod{p} = g^{s_{SM-BGW_{j-LAG}}} \cdot r_{j-LAG} \pmod{p} = g^{s_{SM-BGW_{j-LAG}} \cdot r_{j-LAG}} \pmod{p} = g^{s_{SM-BGW_{j-LAG}} \cdot r_{j-LAG}} \pmod{p} = g^{s_{SM-BGW_{j-LAG}} \cdot r_{j-LAG}} \pmod{p} = g^{s_{SM-BGW_{j-LAG}} \cdot r_{j-LAG}} \pmod{p}.
\]

Property 2 (to resist replay attack). In EPPDC scheme, we assume that there is an adversary $\mathcal{A}$ who can intercept and capture the messages sent by SMs. When adversary $\mathcal{A}$ resends the messages, BGW can detect the replay attack based on the SM's signature or HMAC with the current timestamp. For the same reason, LAG can also detect the replay attack, when an adversary $\mathcal{A}$ resends the BGW's messages.

Property 3 (confidentiality). In EPPDC scheme, the messages maintain their confidentiality when messages are sent to LAG from SM.

Proof. During Stage III of data collection in EPPDC scheme, $SM_{i-LAG}$ has processed the information $m_{i-LAG}$ into $m'_{i-LAG}$ by blind factors $\alpha_{i-LAG-SM}$ and $\beta_{i-LAG-SM}$, before $SM_{i-LAG}$ sends the messages to $BGW_{j-LAG}$.

We consider the following game played between a challenge $\mathcal{C}$ and an adversary $\mathcal{A}$. $\mathcal{C}$ runs the system initialization and sends the system parameters to $\mathcal{A}$. $\mathcal{A}$ performs a polynomial bounded number of queries (these queries may be made adaptively; that is, each query may depend on the answer to the previous queries). By queries, $\mathcal{A}$ can get many couples of $(m_{i-LAG}^{'}, m_{i-LAG}^{' \cdot \ast})$, where $(m_{i-LAG}^{' \cdot \ast}, m_{i-LAG}) \neq m_{i-LAG}$. Based on discrete logarithm problem, $\mathcal{A}$ cannot get the blind factors $\alpha_{i-LAG-SM}$ and $\beta_{i-LAG-SM}$ from $(m_{i-LAG}^{' \cdot \ast}, m_{i-LAG})$. Therefore, an adversary $\mathcal{A}$ cannot get $m_{i-LAG}$, even if $BGW_{j-LAG}$ cannot get $m_{i-LAG}$ too. Thus the messages maintain their confidentiality when messages are sent to LAG from SM.

Property 4 (nonrepudiation and unforgeability). During Stage III of data collection in EPPDC scheme, $SM_{i-LAG}$ sends $m_{i-LAG}^{' \cdot \ast}$ and its signature to $BGW_{j-LAG}$, which is sent to LAG by $BGW_{j-LAG}$ at a later step.

We consider the following game played between a challenge $\mathcal{C}$ and an adversary $\mathcal{A}$. $\mathcal{A}$ performs a polynomial bounded number of adaptive queries. By queries, $\mathcal{A}$ can get many couples of $(m_{i-LAG}^{' \cdot \ast}, \| T_{4} \|, T_{4})^\ast$. Based on the properties of signature [19], $\mathcal{A}$ cannot forge the $SM_{i-LAG}$'s signature $Sig_{SM_{i-LAG}}(m_{i-LAG}^{' \cdot \ast}, \| T_{4} \|, T_{4})^\ast$ on message $(m_{i-LAG}^{' \cdot \ast}, \| T_{4} \|, T_{4})^\ast$. Thus, $SM_{i-LAG}$ cannot repudiate $m_{i-LAG}^{' \cdot \ast}$, $\| T_{4} \|$, being sent by himself.

In the following steps, $BGW_{j-LAG}$ signs the message $m_{i-LAG}^{' \cdot \ast}$ via blind signature with its private key $SK_{BGW_{j-LAG}}$. By adaptive queries, $\mathcal{A}$ can get many couples
of \( s^*_{\text{SM}} \) and \( r^*_{\text{LAG}} \). Based on discrete logarithm problem, \( \mathcal{A} \) cannot get \( \text{BGW}\_j\underaccent{bar}{\text{LAG}} \)'s private key \( \text{SK}_{\text{BGW}\_j\underaccent{bar}{\text{LAG}}} \). Based on discrete logarithm problem, if \( \mathcal{A} \) forges \( \text{BGW}\_j\underaccent{bar}{\text{LAG}} \)'s blind signature, the signature will not be authenticated by \( \text{LAG}_i \) making use of (3). Thus, \( \text{BGW}\_j\underaccent{bar}{\text{LAG}} \) cannot repudiate \( (m^*_i, s^*_{\text{SM}}) \) being sent by himself.

Furthermore, based on the properties of signature, the messages sent by \( \text{SM}\_j\underaccent{bar}{\text{LAG}} \) and \( \text{BGW}\_j\underaccent{bar}{\text{LAG}} \) are provided with the source authentication and data integrity.

Property 5 (forward security). In EPPDC scheme, the permit has a validity period \( d \). When permit is valid, both \( \text{SM} \) and \( \text{LAG} \) can noninteractively share \( d \)-dimensional key \( K_{\text{LAG}\_j\_\text{SM}} = (K_1, K_2, \ldots, K_d) \), where \( d \) is the corresponding number of days for \( d \). Thereby in \( d \) days, \( \text{SM} \) and \( \text{LAG} \) use different shared key every day.

In the key evolution phase, \( \text{SM} \) deletes the previous shared key after it has computed the new shared key \( K_{\text{LAG}\_j\_\text{SM}} \). If an adversary \( \mathcal{A} \) compromises a HAN user's \( \text{SM} \), it gets the current shared key \( H(K_{\text{LAG}\_j\_\text{SM}}) \). Assume that an adversary \( \mathcal{A} \) can perform a polynomial bounded number of adaptive queries for a challenge \( \mathcal{C} \). By queries, \( \mathcal{A} \) can get many shared keys \( K_{\text{LAG}\_j\_\text{SM}}^* \). Based on the property of one-way for hash function, \( \mathcal{A} \) cannot get any previous shared key \( K_{\text{LAG}\_j\_\text{SM}} \) making use of the current shared key \( H(K_{\text{LAG}\_j\_\text{SM}}^*) \). Thus adversary \( \mathcal{A} \) cannot compute the previous blind factor, and then it cannot get previous message \( m \) sent by \( \text{SM} \). Therefore, EPPDC scheme provides the evolution and forward secrecy of shared key \( K_{\text{LAG}\_j\_\text{SM}} \).

Finally, we present the comparison results of security levels in Table 1. It can be seen that scheme [7] and scheme [10] achieve confidentiality, authenticity, and data integrity and scheme [9] cannot resist replay attack.

### 6.2. Computation Overhead

In EPPDC scheme, every \( \text{SM} \) sends its processed messages \( m_i \) and its corresponding signatures to \( \text{BGW} \). \( \text{BGW} \) verifies the validity of \( \text{SM} \)'s signature. For available signature, \( \text{BGW} \) signs message \( m_i \) making use of blind signature and sends it to \( \text{LAG} \). \( \text{LAG} \) can verify that the message is indeed sent by \( \text{BGW} \) and \( \text{SM} \). Furthermore, \( \text{LAG} \) can recover the original message \( m_i \).

In EPPDC, we assume that the \( \text{SM} \)'s signature can be converted into the same as the existing literatures such as [7-11] using bilinear pairing, which can also perform the bath verification [7]. So here we mainly discuss the computation complexity of messages \( m \) turning to \( m^*_i \), the signature of \( \text{BGW} \), verification of \( \text{BGW} \)'s signature, and recovery for the original message \( m \). Because the computation complexity is similar in [7, 9] which are both realized by homomorphic encryption, we only consider [7] in the following comparisons.

In EPPDC scheme, \( \text{SM}\_j\underaccent{bar}{\text{LAG}} \) needs 2 exponentiation operations, 3 multiplication operations, and 1 inverse operation in \( Z_p^\ast \) to blind message \( m \) to \( m^*_i \). In [7], \( \text{SM}\_j\underaccent{bar}{\text{LAG}} \) needs 2 exponentiation operations and 1 multiplication operation in \( Z_p^\ast \) to encrypt message \( m \) to \( C \) using homomorphic encryption. In [10], \( \text{SM}\_j\underaccent{bar}{\text{LAG}} \) encrypts message \( m \) to \( C \) making use of AES block cipher.

In EPPDC scheme, \( \text{BGW} \) can perform the bath verification. \( \text{BGW} \) makes use of blind signature to sign blinded messages which are authenticated. Here \( \text{BGW} \) only needs 1 addition operation and 1 multiplication operation in \( G_1 \) to verify that the message is indeed sent by \( \text{BGW} \) and recover the original message \( m \). In [7], \( \text{BGW} \) needs 1 multiplication operation in \( G_1 \) and 1 hash operation. In [10], \( \text{BGW} \) needs 1 addition operation and 2 multiplication operations in \( G_1 \).

In EPPDC scheme, \( \text{LAG} \) verifies that the message is indeed sent by \( \text{BGW} \) and recovers the original message \( m \) which needs 4 exponentiation operations, 6 multiplication operations, 1 addition operation, and 2 inverse operations in \( Z_p^\ast \). In [7], \( \text{LAG} \) can verify that the message is indeed sent by \( \text{BGW} \) and recover the original message \( m \), which needs 1 exponentiation operation, 1 multiplication operation in \( Z_p^\ast \), 2 pairing operations, and 1 hash operation. In [10], \( \text{LAG} \) can verify that the message is indeed sent by \( \text{BGW} \) and recover the original message \( m \), which needs 2 pairing operations, 1 multiplication operation in \( G_1 \), 1 exponentiation operation in \( G_7 \), 2 exponentiation operations in \( Z_p^\ast \), and AES decryption.

Since the AES encryption or decryption and hash function are negligible compared with exponentiation and pairing operations, here we mainly consider the computation overhead for other operations. Table 2 gives the test time for the involved cryptography operations [20]. The experiments are conducted on a computer with Intel i5-3210-2.5 GHz CPU and 4-GB RAM. When a message is sent by \( \text{SM} \), the comparisons of computation complexity for \( \text{SM} \), \( \text{BGW} \), and \( \text{LAG} \) are shown in Table 3.

### Table 1: Comparison of security level.

|                  | [7] | [9] | [10] | EPPDC |
|------------------|-----|-----|------|-------|
| Confidentiality  | Yes | Yes | Yes  | Yes   |
| Authenticity     | Yes | Yes | Yes  | Yes   |
| Data integrity   | Yes | Yes | Yes  | Yes   |
| To resist replay attack | No  | No  | No   | Yes   |
| Forward secrecy  | No  | No  | No   | Yes   |
| Private key evolution | No  | Yes | No   | Yes   |

### Table 2: Cryptographic operations execution time.

| Operation       | Denotation | Time (ms) |
|-----------------|------------|-----------|
| An exponentiation in \( Z_p^\ast \) | \( T_{\text{EX}} \) | 0.067     |
| An addition in \( Z_p^\ast \) | \( T_{\text{Add}} \) | 0.001     |
| A multiplication in \( Z_p^\ast \) | \( T_{\text{Mul}} \) | 0.004     |
| An inverse operation in \( Z_p^\ast \) | \( T_{\text{In}} \) | 0.004     |
| An addition in \( G_1 \) | \( T_{\text{Add}} \) | 0.038     |
| A multiplication in \( G_1 \) | \( T_{\text{Mul}} \) | 8.006     |
| An exponentiation in \( G_T \) | \( T_{\text{Exp}} \) | 1.882     |
| A pairing operation | \( T_P \) | 16.064    |
Table 3: The comparisons of computation complexity.

| SM      | BGW          | LAG          | Reference |
|---------|--------------|--------------|-----------|
| $2T_{EX} + T_{Mu}$ | $T_{PM}$     | $T_{EX} + T_{Mu} + 2T_{P}$ | [7]       |
| 0       | $2T_{PM} + T_{Add}$ | $2T_{EX} + T_{Exp} + T_{PM} + 2T_{P}$ | [10]      |
| $2T_{EX} + 3T_{Mu} + T_{In}$ | $T_{Mu} + T_{Ad}$ | $4T_{EX} + 6T_{Mu} + T_{Ad} + 2T_{In}$ | This paper |

When $n$ messages are sent by different SMs to the same LAG, LAG can make use of the batch verification to reduce pairing operation from $2n$ to $n + 1$ [7, 10].

With the exact operation costs, we depict the variation of computation costs in terms of the message number $n$ in Figures 4 and 5, which is for BGW and LAG, respectively. From the figures, it can be obviously shown that the EPPDC scheme largely reduces the computation complexity for both BGW and LAG.

7. Conclusions

This paper proposes an efficient and privacy-preserving data collection scheme for smart grid, which is based on the blind signature and the key distribution scheme. This scheme can achieve that the users’ data information is transmitted to the local aggregator through building gateway. And we analyze the EPPDC scheme. The analysis shows that the scheme not only provides privacy preserving but also has less computation cost than existing schemes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China (61102056, 61201132), Fundamental Research Funds for the Central Universities of China (K5051301013), and the 111 Project of China (B08038).

References

[1] K. Geisler, “The Relationship Between Smart Grids and Smart Cities,” 2013, http://smartgrid.ieee.org/may-2013/869-the-relationship-between-smart-grids-and-smart-cities.
[2] E. Woods, “Why Smart Cities Need Smart Grids,” March 2013, http://www.navigantresearch.com/blog/why-smart-cities-need-smart-grids.
[3] U.S. Department of Energy, The Smart Grid: An Introduction, U.S. Department of Energy, Washington, DC, USA, 2008.
[4] S. Bojanić and O. N.-T. Garcia, “Security aspects of advanced metering infrastructures,” in Proceedings of the 9th Symposium on Industrial Electronics (INDEL ’12), pp. 205–208, Banja Luka, Bosnia and Herzegovina, November 2012.
[5] M. Jawurek, F. Kerschbaum, and G. Danezis, “SoK: Privacy Technologies for Smart Grids—A Survey of Options,” http://www.research.microsoft.com/pubs/178055/paper.pdf.
[6] R. Cristina, V. Giacomo, and C. Antonio, “Privacy-preserving smart metering with multiple data consumers,” Computer Networks, vol. 57, no. 7, pp. 1699–1713, 2013.
[7] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, “EPPA: an efficient and privacy-preserving aggregation scheme for secure
smart grid communications,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1632, 2012.

[8] N. Saputro and K. Akkaya, “On preserving user privacy in Smart Grid advanced metering infrastructure applications,” *Security and Communication Networks*, vol. 7, no. 1, pp. 206–220, 2014.

[9] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, “EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.

[10] H. K.-H. So, S. H. M. Kwok, E. Y. Lam, and K.-S. Lui, “Zero-configuration identity-based signcryption scheme for smart grid,” in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm ’10)*, pp. 321–326, October 2010.

[11] J. Chen and Y. Y. Zhang, “The scheme of identity-based aggregation signcryption in smart grid,” *Advanced Materials Research*, vol. 960-961, pp. 832–835, 2014.

[12] C. Rottondi, G. Verticale, and A. Capone, “Privacy-preserving smart metering with multiple data consumers,” *Computer Networks*, vol. 57, no. 7, pp. 1699–1713, 2013.

[13] C. Rottondi, G. Verticale, and C. Krauß, “Distributed privacy-preserving aggregation of metering data in smart grids,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1342–1354, 2013.

[14] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, “Human-factor-aware privacy-preserving aggregation in smart grid,” *IEEE Systems Journal*, vol. 8, no. 2, pp. 598–607, 2014.

[15] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, “Aggregated-proofs based privacy-preserving authentication for V2G networks in the smart grid,” *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1722–1733, 2012.

[16] H. Krawczyk, R. Canetti, and M. Bellare, “HMAC: keyed-hashing for message authentication,” RFC 2104, 1997.

[17] D. Liu, P. Ning, and K. Sun, “Efficient self-healing group key distribution with revocation capability,” in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS ’03)*, pp. 231–240, Washington, DC, USA, October 2003.

[18] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, “Blind signatures based on the discrete logarithm problem,” in *Advances in Cryptology—EUROCRYPT ’94*, vol. 950 of *Lecture Notes in Computer Science*, pp. 428–432, 1995.

[19] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, PTR, 2003.

[20] PBC Library, http://crypto.stanford.edu/pbc.
