Hindawi
Security and Communication Networks
Volume 2022, Article ID 2734487, 12 pages
https://doi.org/10.1155/2022/2734487

Research Article

Integrity Protection for Data Aggregation in Smart Grid

Qing Zhu,[1,2] Huijie Lin,[1,2] Changsheng Wan ,[3] Yuan Xie,[3] and Shaowu Peng[3]

[1]*Nari Technology Co., Ltd, Nanjing 211106, China*
[2]*State Grid Electric Power Research Institute, Nanjing 211106, China*
[3]*School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China*

Correspondence should be addressed to Changsheng Wan; wan.changsheng@163.com

Received 12 July 2022; Revised 20 September 2022; Accepted 5 October 2022; Published 22 October 2022

Academic Editor: Youwen Zhu

To evaluate the working state of the smart grid, users need to count the status of devices collected by multiple smart meters. To avoid network congestion during data collection processes in the smart grid, an aggregation protocol is required to aggregate messages from multiple smart meters into one short message. However, since these messages may contain sensitive information of smart meters, privacy is a fundamental requirement for such a data aggregation protocol. At the same time, to avoid tampering of transmitted messages, data integrity is another important requirement during data aggregation. Currently, the privacy issue has been well addressed by using homomorphic encryption algorithms such as the Paillier cryptographic system, where multiple smart meters use the same encryption key for encrypting different messages. However, the integrity issue is much harder than the privacy issue since multiple smart meters use different private keys for signing different messages. To address the integrity issue, we propose a novel data aggregation protocol for the smart grid, called DASG. In DASG, we shall show how to aggregate signatures from multiple smart meters using the Chinese Remainder theorem and the Lagrange interpolation techniques, respectively. Since these two techniques can aggregate multiple messages into one, DASG is quite light-weight. Therefore, our newly designed data aggregation protocol for the smart grid can achieve both security and efficiency goals in the smart grid environment. Experimental results show that DASG is feasible for real world applications.

1. Introduction

Recently, the smart grid has been widely deployed all over the world, which provides an intelligent power supply for smart cities [1]. In the smart grid environment, smart meters, users, and the grid operator's network elements may communicate with each other for various applications, such as state estimation of the power distribution system and demand-side management [2]. Typically, traffic data in smart grid applications include billing data (e.g., active energy consumption data and reactive energy consumption data) and operational data (e.g., power, voltage, current, power outage logs, and alarms) [3, 4].

A big challenge is that the smart grid network has to handle a large number of messages collected by smart meters. As shown in [5, 6], each smart meter may send a few kilobytes of data every 15–60 minutes to grid operators. When there are a lot of smart meters, network congestion will occur [7–9]. Moreover, adversaries may track data flows and establish attacks on smart meters [10]. Therefore, to avoid network congestion and data-flow tracking in the smart grid, a data aggregation protocol for the smart grid is desired, which can aggregate multiple messages into one short message. By doing so, the number of messages transmitted across the operator's network is reduced, and the network congestion is avoided. Moreover, since messages from multiple smart meters are aggregated into one short message, adversaries will no longer be able to track the data flow from smart meters.

Regardless of the underground technology implemented, a typical Data Aggregation protocol for Smart Grid (DASG) includes five entities as shown in Figure 1 [11]: the Data Repository (DR) who stores aggregated data, the user who downloads aggregated data from the data repository, the gateway who aggregates messages from multiple smart meters and then sends the aggregated message to the data

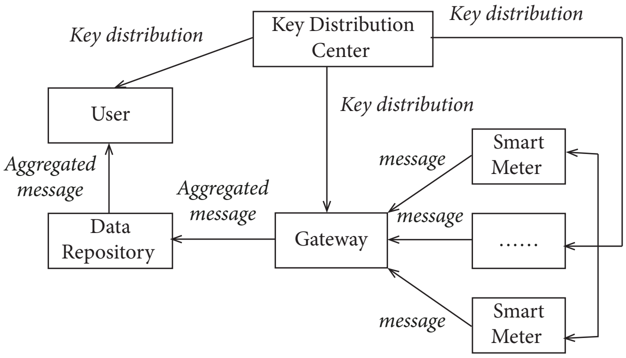

Figure 1: System model of DASG.

repository, the Smart Meter (SM) who collects data and sends back to the data repository over the gateway, the Key Distribution Center (KDC) who is a trusted entity for distributing keys to the gateway, smart meters, and the user. There are mainly two processes, namely, the key distribution process and the data aggregation process. During the key distribution process, the KDC distributes keys to smart meters, the gateway, and the user for protecting the subsequent data flow. During the data aggregation process, the gateway aggregates status messages from multiple smart meters into one short message and sends them to the data repository for storage. Then, the user downloads aggregated message for status statistics.

Security is the main concern for DASG, which mainly includes two parts, namely, privacy and integrity. Currently, the privacy issue is well addressed by homomorphic encryption algorithms. In this sort of schemes, smart meters encrypt transmitted messages with the user's public key, the gateway aggregates multiple encrypted messages into one short message without knowing the corresponding plaintexts, and the user decrypts the aggregated short message to get the status information. However, the integrity problem is still unsolved. Since multiple messages to be aggregated are signed with different private keys, the integrity problem is much harder. Unfortunately, the integrity issue is very important for DASG as illustrated below. If the adversary tampers with status information contained in the data flow, the user may make the wrong decision based on the tampered data, resulting in chaos over the smart grid. Therefore, the data aggregation protocol in the smart grid should provide integrity protection for collected data.

Efficiency is the second serious concern for DASG. First, due to limited resources, smart meters are seriously concerned about the high computation and communication costs for processing data illustrated as follows: (1) smart meters will typically process a variety of data. To provide privacy and integrity protections, smart meters have to run complicated cryptographic algorithms on collected data before sending it to the gateway, leading to high computation cost. (2) Transmitting a large volume of data to the gateway will result in high communication cost. Second, since the gateway has to aggregate a lot of messages from multiple smart meters, the computation cost on it will be quite high. Therefore, the data aggregation protocol in the smart grid should be highly efficient to handle a lot of messages.

Taking both integrity and efficiency into account, we aim to design a data aggregation protocol for smart grids. This newly designed data aggregation protocol can aggregate messages collected from multiple smart meters into one short message. More specifically, the data aggregation protocol for the smart grid should fulfill the following requirements.

(1) Integrity. It should be guaranteed that the adversary cannot tamper with the transmitted data in this protocol.

(2) Status statistics. It should be guaranteed that the data aggregation protocol can evaluate the status of the smart grid. For example, the user may want to count the number of smart meters whose currents are higher than a dangerous value. Or, the user may want to count the number of smart meters whose devices are shut down. In all these kinds of applications, the gateway needs to compute the sum of messages from multiple smart meters.

(3) Computation cost. It should be guaranteed that the computation costs on smart meters, the gateway, and the user are low when running the data aggregation protocol.

(4) Communication cost. It should be guaranteed that the communication costs across the smart grid are low by running the data aggregation protocol.

Obviously, designing a data aggregation protocol for a smart grid is a nontrivial task illustrated as follows. First, the data aggregation protocol has to aggregate multiple messages signed by different private keys into one short message. Second, the data aggregation protocol should have the status statistics feature. Third, since there are a large number of messages to be aggregated by the gateway, the efficiency of a data aggregation protocol should be quite high. Recently, data aggregation protocols for smart grids have focused on the privacy feature during data aggregation, while the integrity feature has been largely neglected. Furthermore, when working on this research topic, we observe that there is no cryptographic primitive which can be directly applied to satisfy all the above requirements. Detailed analysis for arriving at this conclusion is presented in the next section. This issue is becoming more and more serious with the explosive deployment of smart grids in the real world. Motivated by this observation, we mainly make three contributions in this paper illustrated as follows.

(1) We first identify the characteristics of data aggregation in smart grids and then present a comprehensive set of requirements for the protocol of this kind. We show some problems of current data aggregation protocols in smart grids.

(2) We propose a novel data aggregation protocol for smart grids called DASG, which can fulfill all the

above-given security and efficiency requirements. Moreover, different from current data aggregation protocols in smart grids, DASG mainly focuses on the integrity issue. To satisfy all the above-given requirements, we'll design two homomorphic signing algorithms based on the Lagrange interpolation function [12] and the Chinese Remainder theorem [13], which allow the gateway to aggregate multiple messages signed by different private keys. Since these two techniques are quite light-weight, DASG can enjoy the high efficiency feature.

(3) We analyze the security of DASG, which shows that it satisfies the above requirements (1) and (2). And, we evaluate the efficiency of DASG, which shows that it satisfies the above requirements (3) and (4).

The remainder of this paper is organized as follows. First, we survey the related work in Section 2. Second, we present the DASG protocol in Section 3. Third, we analyze the security of DASG in Section 4. Fourth, we evaluate the efficiency of DASG in Section 5. Finally, we draw our conclusions in Section 6.

2. Related Work

Due to the large volume of data to be collected, data aggregation is an essential protocol in smart grid, which can reduce the volume of transmitted data significantly and hence avoid network congestion. Since adversaries may intercept and tamper with transmitted data, privacy and integrity are basic requirements for data aggregation protocols. Therefore, many works have focused on designing secure and efficient data aggregation protocols. Typically, these data aggregation protocols can be categorized into three types, namely, perturbation-based data aggregation, partition-based data aggregation, and homomorphic-encryption-based data aggregation. In the following paragraphs, we shall analyze them, respectively.

Perturbation-based data aggregation is the first technique for addressing the privacy issue in smart grid. In this kind of scheme, smart meters provide privacy protection for original data by adding random numbers to it [14–19]. For example, in [14], the authors introduced a cluster-based data aggregation protocol with privacy protection. In [15], the authors provided a K-indistinguishable privacy-preserving data aggregation protocol. In [16], the authors proposed a differential-privacy-based data aggregation protocol. In general, the random numbers can be random noises [17, 18] or interferences [19]. However, adding perturbations to the original data will lead to high computation costs. Moreover, perturbation-based data aggregation protocols cannot prevent adversaries from tampering data.

Partition-based data aggregation is the second technique for addressing the privacy issue in the smart grid. In this kind of scheme, original data are split into multiple partitions to achieve the privacy-preserving goal. For example, in [20], the authors designed a partition-mixture technique for data aggregation, and the authors in [21, 22] presented a data aggregation protocol based on secret-sharing. However, partition-based data aggregation protocols will lead to data loss an high communication costs among smart meters, resulting in failure of aggregation.

Homomorphic-encryption-based data aggregation is the third technique for addressing the privacy issue in the smart grid. In this kind of scheme, smart meters encrypt the original data, and the gateway aggregates data from multiple smart meters without decrypting it to provide privacy protection. For example, in [23–25], several addition homomorphic algorithms were designed for smart grid. In [26], the authors introduced a concealed data aggregation protocol. In [27, 28], the authors proposed a multi-layer security protection protocol for data aggregation. However, this sort of schemes will lead to high computation costs and cannot provide integrity protections for transmitted data.

There are several more approaches for data aggregation in smart grid. For example, in [29], the authors used the block-chain technique for managing the data aggregation processes [29], and in [4], the authors aggregated data according to the quality-of-service requirements.

From the above analysis, it can be seen that only the homomorphic encryption technique can provide privacy protection for aggregated data well. However, existing homomorphic-encryption-based data aggregation technique cannot provide integrity protection for transmitted data and lack the status statistics capacity. Moreover, many schemes will lead to low efficiency. The issues of existing schemes are shown in Table 1. Therefore, it is desirable to design a data aggregation protocol for a smart grid, which can efficiently aggregate data with integrity and statistical capacity.

3. DASG: The Protocol

3.1. Preliminaries

3.1.1. Lagrange Interpolation. The Lagrange interpolation function [12] is a linear combination $L(x) = \sum_{i=0}^{n} y_i l_i(x)$ of Lagrange basis polynomials $l_i(x) = \prod_{0 \le m \le n, m \ne i} x - x_m / x_i - x_m$, where $0 \le i \le n$, and $\{(x_i, y_i), 0 \le i \le n\}$ is a set of different 2-dimension vectors.

Specifically, for $n = 1$, the following equation holds: $L(0) = y_0 0 - x_1 / x_0 - x_1 + y_1 0 - x_0 / x_1 - x_0$.

3.1.2. Chinese Remainder Theorem. Given a set of k integers $S = \{n_1, \dots, n_k\}$ whose elements are pairwise relatively prime, and a system of simultaneous congruences $\{x = a_1 \mathrm{mod} n_1, x = a_2 \mathrm{mod} n_2, \dots, x = a_k \mathrm{mod} n_k\}$, the unique solution for x modulo $n = n_1 \dots n_k$ for the following system of simultaneous congruences is $x = \sum_{i=1}^{k} a_i N_i M_i$ where $N_i = n / n_i$ and $M_i = N_i^{-1} \mathrm{mod} n_i$ [13].

3.1.3. Bilinear Map. Let G and G_T be two cyclic groups with the same prime order p (i.e., $|G| = |G_T| = p$). Let g be the generator of G. Then, a bilinear map group is defined as $\widehat{e}: G \times G \longrightarrow G_T$, where $\widehat{e}$ has the following properties:

(1) Bilinearity. For $\forall x, y \in Z_p$, $\widehat{e}(g^x, g^y) = \widehat{e}(g, g)^{xy} = \widehat{e}(g^{xy}, g) = \widehat{e}(g, g^{xy}) = \widehat{e}(g^y, g^x)$. For $\forall A, B \in G$, $\widehat{e}(AB, g) = \widehat{e}(A, g)\widehat{e}(B, g)$

Table 1: Comparison of different schemes.

	Perturbation-based schemes	Partition-based schemes	Homomorphic-encryption-based schemes
Integrity protection	×	×	×
Privacy protection	Partially	Partially	√
High efficiency	×	×	×

Table 2: Notations in this paper.

Notation	Description
$\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}$	The private key and the public key of the key distribution center, respectively
λ	The security strength of this data aggregation system
$\mathrm{sk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{GW}}$	The private key and the public key of the gateway, respectively
$\mathrm{sk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{SMi}}$	The private key and the public key of the i th smart meter, respectively
m_i, τ_i	The status reported by SM_i, and the corresponding signed and encrypted ciphertext
$\eta = \{m_i, 1 \le i \le t\}$	The set of messages to be aggregated by the gateway
$\tau = \{\tau_i, 1 \le i \le t\}$	The set of signatures to be aggregated by the gateway
t	The number of smart meters in DASG
π	The short message aggregated by the gateway
Ω	The short signature aggregated by the gateway
UID, GW, SMi	The identities of the user, the gateway, and the smart meters, respectively
$a_1, a_2, \ldots, a_l, A_m, \mathrm{sk}_s$	Private keying materials of the key distribution center in the construction based on the Chinese remainder theorem
$G, g, p, \mathrm{pk}_s, A, u$	Public keying materials of the key distribution center in the construction based on the Chinese remainder theorem
$A_1, B_1, \mathrm{sk}_{\mathrm{gw}}$	Private keying materials of the gateway in the construction based on the Chinese remainder theorem
$\mathrm{sk}_{\mathrm{SMi}}, A_{i+1}, B_{i+1}$	Private keying materials of the SMi in the construction based on the Chinese remainder theorem
$\mathrm{sk}_a, \mathrm{sk}_b$	Private keying materials of the key distribution center in the construction based on the Lagrange interpolation function
$G, g, p, \mathrm{pk}_a, \mathrm{pk}_b, u$	Public keying materials of the key distribution center in the construction based on the Lagrange interpolation function
$h(\cdot)$	h: $Z_p \rightarrow Z_p$ is a hash function
$H(\cdot)$	H: $Z_p \rightarrow G$ is a hash function
ID_i	The index for uniquely identifying m_i

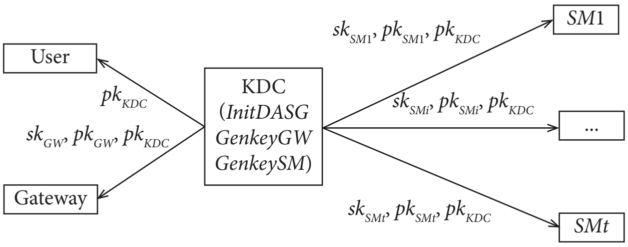
User
pk_{KDC}
$sk_{GW}, pk_{GW}, pk_{KDC}$
Gateway
KDC
(InitDASG
GenkeyGW
GenkeySM)
$sk_{SM1}, pk_{SM1}, pk_{KDC}$
$sk_{SMi}, pk_{SMi}, pk_{KDC}$
$sk_{SMt}, pk_{SMt}, pk_{KDC}$
SM1
...
SMt

Figure 2: The key distribution process of DASG.

(2) Nondegeneracy. $\exists A, B \in G$ such that $\hat{e}(A, B) \neq 1_{G_T}$

(3) Computability. For $\forall A, B \in G$, it is efficient to calculate $\hat{e}(A, B)$

The above-given $\hat{e}$ can be constructed by the Weil or Tate pairings [30, 31] on elliptic curves.

3.2. System Model. The system model of DASG is described in the following subsections, and Table 2 lists the notations used in this paper.

3.2.1. The Key Distribution Process. Figure 2 shows the key distribution process of DASG. During this process, the KDC first initializes the set of public and private keying materials of the data aggregation system. The initialization algorithm (InitDASG) is defined as follows.

$\{\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}\} \leftarrow \mathrm{InitDASG}(\lambda)$. This algorithm is run by the key distribution center for initializing system parameters for DASG. It takes the parameter of security strength (λ) as input, and outputs the private key of the KDC ($\mathrm{sk}_{\mathrm{KDC}}$) and the corresponding public key of the KDC ($\mathrm{pk}_{\mathrm{KDC}}$).

For the user (UID) who wants to verify and use the aggregated message, the KDC distributes $\mathrm{pk}_{\mathrm{KDC}}$ to it.

For the gateway (GW) who aggregates messages received from multiple SMs, the KDC generates the private key ($\mathrm{sk}_{\mathrm{GW}}$) and the corresponding public key ($\mathrm{pk}_{\mathrm{GW}}$) for the gateway. And then, the KDC distributes ($\mathrm{sk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{KDC}}$) to the gateway. The key generating algorithm for the gateway is illustrated as follows.

$\{\mathrm{sk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{GW}}\} \leftarrow \mathrm{GenkeyGW}(\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}, \mathrm{GW})$. This algorithm is run by the key distribution center for generating public and private keys for the gateway. It takes as inputs the private key of the KDC ($\mathrm{sk}_{\mathrm{KDC}}$), the public key of the KDC ($\mathrm{pk}_{\mathrm{KDC}}$), and the gateway's identity (GW), and outputs the private key of the gateway ($\mathrm{sk}_{\mathrm{GW}}$) and the corresponding public key of the gateway ($\mathrm{pk}_{\mathrm{GW}}$).

For the i th smart meter (SMi) who sends a message to the gateway for aggregation, the KDC generates the private key ($\mathrm{sk}_{\mathrm{SMi}}$) and the corresponding public key ($\mathrm{pk}_{\mathrm{SMi}}$) for the SMi. And then, the KDC distributes ($\mathrm{sk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{KDC}}$) to the i th smart meter. The key generating algorithm for the SMi is illustrated as follows.

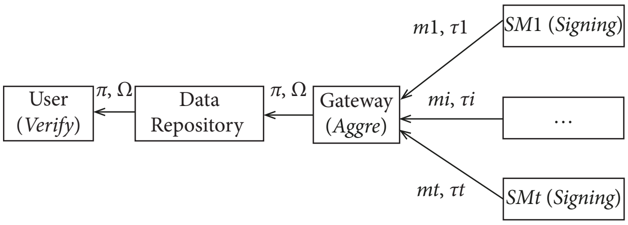

Figure 3: The data aggregation process of DASG.

$\{\mathrm{sk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{SMi}}\} \leftarrow \mathrm{GenkeySM}(\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}, \mathrm{SMi})$. This algorithm is run by the key distribution center for generating public and private keys for the i th smart meter. It takes as inputs the private key of the KDC ($\mathrm{sk}_{\mathrm{KDC}}$), the public key of the KDC ($\mathrm{pk}_{\mathrm{KDC}}$), and the i th smart meter's identity (SMi), and outputs the private key of the i th smart meter ($\mathrm{sk}_{\mathrm{SMi}}$) and the corresponding public key of the i th smart meter ($\mathrm{pk}_{\mathrm{SMi}}$).

After the initialization phase, the data center holds ($\mathrm{sk}_{\mathrm{KDC}}$ and $\mathrm{pk}_{\mathrm{KDC}}$), the user holds $\mathrm{pk}_{\mathrm{KDC}}$, the gateway holds ($\mathrm{sk}_{\mathrm{GW}}$, $\mathrm{pk}_{\mathrm{GW}}$, and $\mathrm{pk}_{\mathrm{KDC}}$), and the SMi holds ($\mathrm{sk}_{\mathrm{SMi}}$, $\mathrm{pk}_{\mathrm{SMi}}$, and $\mathrm{pk}_{\mathrm{KDC}}$).

3.2.2. The Data Aggregation Process. Figure 3 shows the data aggregation process of DASG. Assuming there are t smart meters, the data aggregation process mainly includes three steps. First, each smart meter will send a message to the gateway. Second, the gateway will aggregate all the messages into one short message, and compute the corresponding signature from multiple signatures from SMs. Finally, the user will verify the aggregated message. The details of these three steps are illustrated as follows.

Step 1. Before each smart meter (SMi) sends a message (m_i) to the gateway for aggregation, it will sign this message using the Signing algorithm, which is illustrated as follows.

$\{\tau_i\} \leftarrow \mathrm{Signing}(m_i, \mathrm{sk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{KDC}})$. This algorithm is run by the SMi for signing the message m_i. It takes as inputs the message to be signed (m_i), the SMi's private key ($\mathrm{sk}_{\mathrm{SMi}}$), the SMi's public key ($\mathrm{pk}_{\mathrm{SMi}}$), the KDC's public key ($\mathrm{pk}_{\mathrm{KDC}}$), and outputs the signature (τ_i).

Step 2. Upon receiving the set of t messages ($\eta = \{m_i, 1 \le i \le t\}$) and the corresponding signatures ($\tau = \{\tau_i, 1 \le i \le t\}$), the gateway aggregates η, τ into one short message π and one short signature Ω using the Aggre algorithm, which is illustrated as follows.

$\{\pi, \Omega\} \leftarrow \mathrm{Aggre}(\eta, \tau, \mathrm{sk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{GW}}, \mathrm{pk}_{\mathrm{KDC}})$. This algorithm is run by the gateway for aggregating the set of messages (η) and the corresponding set of signatures (τ) from multiple smart meters. It takes as inputs the set of messages (η), the set of signatures (τ), the gateway's private key ($\mathrm{sk}_{\mathrm{GW}}$), the gateway's public key ($\mathrm{pk}_{\mathrm{GW}}$), the KDC's public key ($\mathrm{pk}_{\mathrm{KDC}}$), and outputs the aggregated message (π), and the corresponding aggregated signature (Ω).

Step 3. After data aggregation, the gateway stores π and Ω to the data repository, and the user downloads π and Ω for verifying the aggregated message (π) and the corresponding signature (Ω) using the Verify algorithm, which is illustrated as follows.

$\{\mathrm{True}, \mathrm{False}\} \leftarrow \mathrm{Verify}(\pi, \Omega, \mathrm{pk}_{\mathrm{KDC}})$. This algorithm is run by the user for verifying the aggregated message π. It takes as inputs the aggregated message (π), the aggregated signature (Ω), the KDC's public key ($\mathrm{pk}_{\mathrm{KDC}}$), and outputs True if the message passed the verification. Otherwise, it outputs False.

After the data aggregation process, the user gets the sum of messages from multiple smart meters (π), and checks the integrity of π using the Verify algorithm, which is the status statistics information of the smart grid.

In the above-given system model, the set of messages (η) is signed by smart meters and aggregated by the gateway, the user can make sure whether η is tampered by verifying π and Ω. Therefore, the newly designed protocol has the integrity feature. Since π is the sum of messages from multiple smart meters, the newly designed protocol has the status statistics feature. So, DASG can achieve the security goals described in Section 1. In Section 4, we shall further analyze the security of DASG.

3.3. Construction. The construction of DASG is a tuple (InitDASG, GenkeyGW, GenkeySM, Signing, Aggre, Verify) of probabilistic polynomial time algorithms. In the following two subsections, we shall present two constructions based on the Chinese Remainder theorem and the Lagrange interpolation function, respectively.

3.3.1. Construction Based on the Chinese Remainder theorem. $\{\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}\} \leftarrow \mathrm{InitDASG}(\lambda)$. The key distribution center runs this algorithm for initializing system parameters for DASG as follows. First, the key distribution center generates a group G with a random generator g and a λ-bit prime order p. Second, the key distribution center randomly generates l numbers $a_1 < a_2 < \ldots < a_l \in Z_p$ where $a_1, \ldots, a_l$ are pairwise relatively prime and computes $A_m = \prod_{i=1}^{2} a_i$, $A = \sum_{i=1}^{l} a_i$. Third, the key distribution center randomly generates $u \in G$. Fourth, the key distribution center randomly generates the main signing key as $\mathrm{sk}_s \in (a_l, A_m)$ and computes the corresponding public key $\mathrm{pk}_s = g^{\mathrm{sk}_s}$. Finally, the key distribution center gets $\mathrm{sk}_{\mathrm{KDC}} = \{a_1, a_2, \ldots, a_l, A_m, \mathrm{sk}_s\}$ and $\mathrm{pk}_{\mathrm{KDC}} = \{G, g, p, pk_s, A, u\}$.

$\{sk_{GW}, pk_{GW}\} \leftarrow \mathrm{GenkeyGW}(\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}, \mathrm{GW})$. The key distribution center runs this algorithm for generating public and private keys for the gateway as follows. First, the key distribution center computes the aggregating key as $sk_{gw} = sk_s \mathrm{mod} a_1$. Second, the key distribution center computes the related materials $A_1 = A/a_1$ and $B_1 = A_1^{-1} \mathrm{mod} a_1$. Finally, the key distribution center gets $\mathrm{sk}_{\mathrm{GW}} = \{A_1, B_1, \mathrm{sk}_{\mathrm{gw}}\}$ and $pk_{GW} = g^{sk_{gw}}$.

$\{\mathrm{sk}_{\mathrm{SMi}}, \mathrm{pk}_{\mathrm{SMi}}\} \leftarrow \mathrm{GenkeySM}(\mathrm{sk}_{\mathrm{KDC}}, \mathrm{pk}_{\mathrm{KDC}}, \mathrm{SMi})$. The key distribution center runs this algorithm for generating public and private keys for the i th smart meter as follows. First, the key distribution center computes $sk_{SMi} = sk_s \mathrm{mod}\, a_{i+1}$. Second, the key distribution center computes the related materials $A_{i+1} = A/a_{i+1}$ and $B_{i+1} = A_{i+1}^{-1} \mathrm{moda}_{i+1}$. Third, the key distribution center computes the public key as $pk_{SMi} = g^{sk_{SMi}}$. Finally, the key distribution center gets $\mathrm{sk}_{\mathrm{SMi}} = \{\mathrm{sk}_{\mathrm{SMi}}, A_{i+1}, B_{i+1}\}$ and $pk_{SMi} = g^{sk_{SMi}}$.

$\{\tau_i\}\leftarrow\text{Signing}(m_i, \text{sk}_{\text{SM}i}, \text{pk}_{\text{SM}i}, \text{pk}_{\text{KDC}})$. The SM$i$ runs this algorithm for signing the message m_i as $\tau_i = (H(I\ Di)u^{m_i})^{sk_{SMi}A_{i+1}B_{i+1}}$, where $H\colon Z_p \longrightarrow G$ is a hash function, and IDi is the index for uniquely identifying m_i.

$\{\pi, \Omega\}\leftarrow\text{Aggre}(\eta, \tau, \text{sk}_{\text{GW}}, \text{pk}_{\text{GW}}, \text{pk}_{\text{KDC}})$. The gateway runs this algorithm for aggregating the set of messages ($\eta = \{m_1, \ldots, m_t\}$) and the corresponding set of signatures ($\tau = \{\tau_1, \ldots, \tau_t\}$) and from multiple smart meters as follows. First, the gateway aggregates messages as $\pi = \sum_{i=1}^{t} m_i$. Second, the gateway computes $\Omega = (u^{\pi}\prod_{i=1}^{t} H(IDi))^{sk_{gw}A_1B_1}\prod_{i=1}^{t}\tau_i$.

$\{\text{True}, \text{False}\}\leftarrow\text{Verify}(\pi, \Omega, \text{pk}_{\text{KDC}})$. The user runs this algorithm for verifying the aggregated message π as $\widehat{e}(\Omega, g)? = \widehat{e}(u^{\pi}\prod_{i=1}^{t} H(\text{IDi}), \text{pk}_s)$. If the above-given equation holds, this algorithm returns True. Otherwise, it returns False.

3.3.2. Construction Based on the Lagrange Interpolation Function. $\{\text{sk}_{\text{KDC}}, \text{pk}_{\text{KDC}}\}\leftarrow\text{InitDASG}(\lambda)$. The key distribution center runs this algorithm for initializing system parameters for DASG as follows. First, the key distribution center generates a group G with a random generator g and a λ-bit prime order p. Second, the key distribution center randomly generates two private keys $\text{sk}_a, \text{sk}_b \in Z_p$, and gets $L(x) = \text{sk}_a + \text{sk}_b x \text{mod} p$. Third, the key distribution center randomly generates $u \in G$. Fourth, the key distribution center computes $\text{pk}_a = g^{\text{sk}_a}$ and $\text{pk}_b = g^{\text{sk}_b}$. Finally, the key distribution center gets $\text{sk}_{\text{KDC}} = \{\text{sk}_a, \text{sk}_b\}$ and $\text{pk}_{\text{KDC}} = \{G, g, p, \text{pk}_a, \text{pk}_b, u\}$.

$\{\text{sk}_{\text{GW}}, \text{pk}_{\text{GW}}\}\leftarrow\text{GenkeyGW}(\text{sk}_{\text{KDC}}, \text{pk}_{\text{KDC}}, \text{GW})$. The key distribution center runs this algorithm for generating public and private keys for the gateway as follows. First, the key distribution center computes the aggregating key as $\text{sk}_{\text{GW}} = \text{sk}_a + \text{sk}_b h(\text{GW}) \in Z_p$, where $h\colon Z_p \longrightarrow Z_p$ is a hash function. Second, the key distribution center computes the corresponding public key as $pk_{GW} = g^{sk_{GW}}$.

$\{\text{sk}_{\text{SMi}}, \text{pk}_{\text{SMi}}\}\leftarrow\text{GenkeySM}(\text{sk}_{\text{KDC}}, \text{pk}_{\text{KDC}}, \text{SMi})$. The key distribution center runs this algorithm for generating public and private keys for the i th smart meter as follows. First, the key distribution center computes the signing key as $\text{sk}_{\text{SMi}} = \text{sk}_a + \text{sk}_b h(\text{SMi}) \in Z_p$, where $h\colon Z_p \longrightarrow Z_p$ is a hash function. Second, the key distribution center computes the corresponding public key as $pk_{SMi} = g^{sk_{SMi}}$.

$\{\tau_i\}\leftarrow\text{Signing}(m_i, sk_{SMi}, pk_{SMi}, pk_{KDC})$. The SMi runs this algorithm for signing the message m_i as $\tau_i = (H(I\ Di)u^{m_i})^{sk_{SMi}}$, where $H\colon Z_p \longrightarrow G$ is a hash function, and IDi is the index for uniquely identifying m_i.

$\{\pi, \Omega\}\leftarrow\text{Aggre}(\eta, \tau, \text{sk}_{\text{GW}}, \text{pk}_{\text{GW}}, \text{pk}_{\text{KDC}})$. The gateway runs this algorithm for aggregating the set of messages ($\eta = \{m_1, \ldots, m_t\}$) and the corresponding set of signatures ($\tau = \{\tau_1, \ldots, \tau_t\}$) and from multiple smart meters as follows. First, the gateway aggregates messages as $\pi = \sum_{i=1}^{t} m_i$. Second, the gateway computes $\Omega = \prod_{i=1}^{t}((H(\text{IDi})u^{m_i})^{\text{sk}_{\text{GW}}h(\text{SMi})/h(\text{SMi}))-h(\text{GW})}\tau_i^{h(\text{GW})/h(\text{GW})-h(\text{SMi})})$.

$\{\text{True}, \text{False}\}\leftarrow\text{Verify}(\pi, \Omega, \text{pk}_{\text{KDC}})$. The user runs this algorithm for verifying the aggregated message π as $\widehat{e}(\Omega, g)? = \widehat{e}(u^{\pi}\prod_{i=1}^{t} H(\text{IDi}), \text{pk}_a)$. If the above-given equation holds, this algorithm returns True. Otherwise, it returns False.

3.3.3. Discussions on the Two Constructions. In the above-given constructions, the message containing the status information (m_i) is signed by the smart meter. Then, the gateway aggregates multiple messages and the corresponding signatures into one short message and the corresponding short signature. Finally, the user checks the integrity of the aggregated short message using the key distribution center's public key (pk_{KDC}). So, DASG can achieve the integrity goal. In Section 4, we shall further analyze the integrity of DASG.

In the above-given constructions, multiple messages containing the status information ($m_1, \ldots, m_t$) are added up by the gateway, and the user can get the sum contained in the aggregated short message (π). So, DASG can achieve the status statistics goal.

In the above constructions, the smart meters and the gateway only uses a few modular exponentiation operations for signing and aggregating multiple messages. And, the user only needs two bilinear pairing operations for multiple messages from smart meters. Therefore, DASG can enjoy high efficiency. In Section 5, we shall analyze the efficiency of DASG in detail.

In the above-given constructions, we assume smart meters are trustworthy. In some scenarios where smart meters are not trustworthy, two smart meters may be in collusion with each other to get the main signing key sk_{KDC}. For example, in the construction based on the Chinese Remainder Theorem, two smart meters may compute the main signing key as $\text{sk}_s = \text{sk}_{\text{SMi}}A_{i+1}B_{i+1} + \text{sk}_{\text{SMj}}A_{j+1}B_{j+1}$. Similarly, in the construction based on the Lagrange interpolation function, two smart meters may compute the main signing key as $sk_a = sk_{SMj}h(SMi)/h(SMi) - h(SMj) + sk_{SMi}h(SMj)/h(SMj) - h(SMi)$. Once these two smart meters get the main signing key sk_a (or sk_s), they can tamper the aggregated message π, and the corresponding signature Ω. To address this issue, the KDC may distribute more private keys to the gateway for data aggregation. For example, in the construction based on the Chinese Remainder Theorem, the KDC may constructs a (x, l) threshold signature system instead of current $(2, l)$ threshold system in the InitDASG and distribute a set of private keys ($\text{sk}_{\text{gw1}} = \text{sk}_s\text{moda}_{\text{gw1}}, \text{sk}_{\text{gw2}} = \text{sk}_s\text{moda}_{\text{gw2}}, \text{sk}_{\text{gw3}} = \text{sk}_s\text{moda}_{gw3}, \ldots$) to the gateway instead of only one private key sk_{gw}. In this case, two smart meters will not be able to compute sk_s anymore. Moreover, when $x > t$, even t smart meters are in collusion, they will not be able to compute sk_s.

In the above-given constructions, we mainly focus on the integrity problem. To provide privacy protection for the message m_i, a homomorphic encryption algorithm can be used on m_i, and our signing and verification algorithms are run over encrypted messages.

4. Security Analysis

In this section, we shall first prove the correctness of DASG. Then, we shall analyze the integrity requirement described in Section 1.

4.1. Correctness. In Section 3.3, smart meters sign status messages using the Signing algorithm, the gateway aggregates multiple status messages and their corresponding signatures into one short message and one short signature using the Aggre algorithm, and the user checks the integrity of the aggregated message using the Verify algorithm. In this subsection, we shall prove the correctness of the Verify algorithm. That is to say, if the user, the gateway and the smart meters run their algorithms correctly, the Verify algorithm will return True.

4.1.1. Correctness of the Chinese-Remainder-Theorem-Based Construction. First, from the Aggre algorithm, we can see that

$$\begin{aligned}\Omega &= \left(u^{\pi}\prod_{i=1}^{t} H(IDi)\right)^{sk_{gw}A_1B_1},\\ \prod_{i=1}^{t}\tau_i &= \left(\prod_{i=1}^{t}\left(H(IDi)u^{m_i}\right)\right)^{sk_{gw}A_1B_1},\\ \prod_{i=1}^{t}\tau_i &= \prod_{i=1}^{t}\left(H(IDi)u^{m_i}\right)^{sk_{gw}A_1B_1+sk_{SMi}A_{i+1}B_{i+1}}.\end{aligned} \tag{1}$$

Second, according to the Chinese Remainder Theorem, we can get $\mathrm{sk}_s = \mathrm{sk}_{\mathrm{gw}}A_1B_1 + \mathrm{sk}_{\mathrm{SMi}}A_{i+1}B_{i+1}$. Therefore, the above equation can be further computed as follows:

$$\begin{aligned}\Omega &= \prod_{i=1}^{t}\left(H(IDi)u^{m_i}\right)^{sk_s}\\ &= \left(\prod_{i=1}^{t} H(IDi)\prod_{i=1}^{t} u^{m_i}\right)^{sk_s}\\ &= \left(u^{\pi}\prod_{i=1}^{t} H(IDi)\right)^{sk_s}.\end{aligned} \tag{2}$$

Third, taking the above equation $\Omega = (u^{\pi}\prod_{i=1}^{t} H(\mathrm{IDi}))^{sk_s}$ into the Verify algorithm, we can get $\widehat{e}(\Omega, g) = \widehat{e}(\,(u^{\pi}\prod_{i=1}^{t} H(\mathrm{IDi}))^{sk_s}, g) \quad = \widehat{e}(u^{\pi}\prod_{i=1}^{t} H(\mathrm{IDi}), \mathrm{pk}_s)$.

From the above-given analysis, we can see that our construction based on the Chinese Remainder theorem is correct.

4.1.2. Correctness of the Lagrange-Interpolation-Function-Based Construction. First, from the Aggre algorithm, we can see that

$$\begin{aligned}\Omega &= \prod_{i=1}^{t}\Big(\left(H(\mathrm{IDi})u^{m_i}\right)^{\mathrm{sk}_{\mathrm{GW}}h(\mathrm{SMi})/h(\mathrm{SMi})-h(\mathrm{GW})}\tau_i^{h(GW)/h(GW)-h(\mathrm{SMi})}\Big)\\ &= \prod_{i=1}^{t}\Big(\left(H(\mathrm{IDi})u^{m_i}\right)^{\mathrm{sk}_{\mathrm{GW}}h(\mathrm{SMi})/h(\mathrm{SMi})-h(\mathrm{GW})+\mathrm{sk}_{\mathrm{SMi}}h(\mathrm{GW})/h(\mathrm{GW})-h(\mathrm{SMi})}\Big).\end{aligned} \tag{3}$$

Second, according to the Lagrange interpolation function, we can get $sk_a = sk_{GW} h(SMi)/h(SMi) \;\; - h(GW) + sk_{SMi} h(GW)/h(GW) - h(SM)$. Therefore, the above equation can be further computed as DSLQ

$$\begin{aligned}\Omega &= \prod_{i=1}^{t} \left((H(\mathrm{IDi}) u^{m_i})^{\mathrm{sk}_a} \right)\\ &= \left(\prod_{i=1}^{t} H(\mathrm{IDi}) \prod_{i=1}^{t} u^{m_i} \right)^{\mathrm{sk}_a}\\ &= \left(u^{\pi} \prod_{i=1}^{t} H(\mathrm{IDi}) \right)^{\mathrm{sk}_a}.\end{aligned} \tag{4}$$

Third, taking the above-given equation $\Omega = (u^{\pi} \prod_{i=1}^{t} H(\mathrm{IDi}))^{\mathrm{sk}_a}$ into the Verify algorithm, we can get $\hat{e}(\Omega, g) = \hat{e}((u^{\pi} \prod_{i=1}^{t} H(I\ Di))^{\mathrm{sk}_a}, g) = \hat{e}(u^{\pi} \prod_{i=1}^{t} H(\mathrm{IDi}), \mathrm{pk}_a)$.

From the above-given analysis, we can see that our construction based on the Lagrange interpolation function is correct.

4.2. Integrity. From Section 3.3, it can be seen that the newly designed Signing algorithm is a variation of the famous BLS signature [32], whose security has been proven in the random oracle model [33]. The integrity of our constructions can be proven in a similar way as shown in Theorem 1.

Theorem 1. *After q_h signing queries, if the adversary A can forge $((ID', M'), \tau_{M'})$ with the probability ε in time t, C can solve the GDH problem with the probability $\varepsilon' \approx \varepsilon / e q_h$ in time $t' = t + (2q_h + 3)T_{me}$, where T_{me} is the time cost of modular exponentiation.*

Proof. The detailed security analysis is shown in the online supplementary material (available here).

So, both constructions given in Section 3.3 can satisfy the integrity requirement described in Section 1. □

5. Efficiency Evaluation

In this section, we shall evaluate the efficiency of DASG according to the requirements described in Section 1, namely the computation and communication costs.

From Section 3, we can see that DASG includes two processes, namely the key distribution process and the data aggregation process. The computation and communication costs are mainly consumed during the data aggregation process, while there are a variety of messages to be signed and aggregated. And, the key distribution process is run by the key distribution center only once before the data aggregation process. Therefore, in the following subsections, we mainly focus on the data aggregation process.

5.1. Computation Costs. During the data aggregation process, multiple smart meters sign their messages using the Signing algorithm with different private keys, the gateway aggregates multiple messages into one short message using the Aggr algorithm, and the user checks the integrity of the aggregated short message using the Verify algorithm. Therefore, we mainly analyze the computation costs consumed by these three algorithms.

As shown in Section 3, the Signing, Aggr and Verify algorithms can be constructed based on the Chinese Remainder theorem and the Lagrange interpolation function. In both constructions, the mathematical operations include modular exponentiation, bilinear pairing, modular multiplication, modular addition, and hash function. Compared with those of the modular exponentiation and bilinear pairing operations, the computation costs of the modular multiplication, the modular addition, and the hash operations can be omitted. Therefore, in the following evaluation, we only take the modular exponentiation and bilinear pairing operations into account.

Currently, there is no data aggregation scheme with integrity protection. Therefore, to evaluate the computation costs reduced by our data aggregation constructions, we consider a benchmark scheme, where the gateway just relays multiple signed messages to the data repository without aggregating them and the user checks each message using the pairing algorithm as $\hat{e}(\tau_i, g)? = \hat{e}(H(\mathrm{IDi})u^{m_i}, \mathrm{pk}_{\mathrm{SMi}})$, where $\tau_i = (H(\mathrm{IDi})u^{m_i})^{\mathrm{sk}_{\mathrm{SMi}}}$ is the signature of m_i.

Assuming there are t smart meters, we can get the computation costs consumed by smart meters, the gateway, and the user, respectively. The results are shown in Table 3. Note that, we assume the Signing algorithm is run t times when computing the total computation costs, since there are t smart meters.

From Table 3, we can draw the following conclusions:

(1) Computation costs on smart meters. The computation costs of the Signing algorithms on each smart meter in all the three schemes are the same ($2T_{\mathrm{me}}$) and are independent of the number of smart meters (t). Therefore, data aggregation will not increase the computation costs on smart meters.

(2) Computation costs on the gateway. The computation costs of the Aggr algorithms in both the Chinese-Remainder-Theorem-based construction and the Lagrange-interpolation-function-based construction are linear to the number of smart meters (t). In addition, it is obvious that the computation cost on the gateway is 0 when the gateway does not aggregate messages. Therefore, data aggregation will increase additional computation costs on the gateway

(3) Computation costs on the user. The computation costs of the Verify algorithms in all the two schemes with data aggregation are the same ($T_{\mathrm{me}} + 2T_p$). However, when messages are not aggregated by the

Table 3: Computation costs of algorithms (t: the number of smart meters; T_{me}: the computation cost of the modular exponentiation operation; T_p: the computation cost of the bilinear pairing operation.).

	Signing	Aggr	Verify	Total
Construction based on Chinese remainder theorem (with data aggregation)	$2T_{\mathrm{me}}$	$(t+1)T_{\mathrm{me}}$	$T_{\mathrm{me}} + 2T_p$	$(3t+2)T_{\mathrm{me}} + 2T_p$
Construction based on Lagrange interpolation function (with data aggregation)	$2T_{\mathrm{me}}$	$3tT_{\mathrm{me}}$	$T_{\mathrm{me}} + 2T_p$	$(5t+1)T_{\mathrm{me}} + 2T_p$
The benchmark construction (without data aggregation)	$2T_{\mathrm{me}}$	0	$(T_{\mathrm{me}} + 2T_p)t$	$3tT_{\mathrm{me}} + 2tT_p$

Table 4: Time costs of algorithms (Unit: ms; t: the number of smart meters).

	Signing	Aggr	Verify	Total
Construction based on Chinese remainder theorem (with data aggregation)	1.4	$0.7t + 0.7$	58.7	$2.1t + 59.4$
Construction based on Lagrange interpolation function (with data aggregation)	1.4	$2.1t$	58.7	$3.5t + 58.7$
The benchmark construction (without data aggregation)	1.4	0	$58.7t$	$60.1t$

gateway, the user will have to run the Verify algorithm once for each smart meter. Therefore, data aggregation will decrease the computation cost on the user.

(4) The total computation costs. The total computation costs in all the three schemes are linear to the number of smart meters (i.e., t). However, the numbers of bilinear pairings in schemes with data aggregation are independent of the number of smart meters, but this in the benchmark without data aggregation depends on the number of smart meters. Since bilinear pairing is the most costly operation, data aggregation will potentially decrease the total computation costs as shown in the following paragraphs.

Then, to further compare the computation costs of these three schemes, we implemented our experiments on a Laptop with an Intel i7 processor whose clock frequency is 3.40 GHz. The operating system of this Laptop is Win10. Cryptographic libraries installed on this Laptop include OPENSSL [34] and PBC [35]. For investigating the computation costs of modular exponentiation and bilinear pairing, we used the 160-bit elliptic curve group [36, 37]. After experimentation, we get $T_{\mathrm{me}} = 0.7$ms and $T_p = 29$ms. Taking these two results into Table 3, we get the time costs of algorithms used in these three schemes, as shown in Table 4.

From Table 4, we can draw the following conclusions:

(1) The computation costs of the Signing algorithms in all the three schemes are 1.4ms, which are quite light weight. Therefore, smart meters enjoy high efficiency, which is suitable for the smart grid, where smart meters are low-power devices.

(2) Aggregating multiple messages into one short message will increase the computation cost on the gateway. In both the Chinese-Remainder-Theorem-based and the Lagrange-interpolation-function-based constructions, the computation costs are linear to the number of smart meters. When there are a lot of smart meters, the computation cost of the Chinese-Remainder-Theorem-based construction is around 1/3 to that of the Lagrange-interpolation-function-based construction, since $\lim_{t\longrightarrow\infty}(0.7t + 0.7/2.1t) = (1/3)$.

(3) In both Chinese-Remainder-Theorem-based and Lagrange-interpolation-based constructions, the verification costs on the user are 58.7ms after data aggregation, which are quite light weight. On the other hand, without data aggregation, the verification costs will be linear to the number of smart meters (58.7t). When there are a lot of smart meters, the computation costs on the user will be quite high.

(4) The total computation cost of the Chinese-Remainder-Theorem-based construction with data aggregation is around 3.5% to that of construction without data aggregation. This is because $\lim_{t\longrightarrow\infty}(2.1t + 59.4/60.1t) = 3.5\%$. The total computation cost of the Lagrange-interpolation-based construction with data aggregation is around 5.8% to that of construction without data aggregation. This is because $\lim_{t\longrightarrow\infty}(3.5t + 58.7/60.1t) = 5.8\%$. The total computation cost of the Chinese-Remainder-Theorem-based construction with data aggregation is around 60% to that of the Lagrange-interpolation-based construction with data aggregation. This is because $\lim_{t\longrightarrow\infty}(2.1t + 59.4/3.5t + 58.7) = 60\%$.

In summary, the data aggregation process can reduce the computation cost on the user by adding additional computation cost on the gateway. And both the Chinese-Remainder-Theorem-based construction and the Lagrange-interpolation-based construction can reduce the total computation cost significantly. Moreover, the Chinese-Remainder-Theorem-based construction is more efficient than the Lagrange-interpolation-based construction. This is because the aggregation process on the gateway in the former scheme is much more efficient. Therefore, DASG can satisfy the computation cost requirement defined in Section 1.

5.2. Communication Costs.

The communication cost is evaluated by using the number of messages and lengths of messages transmitted in DASG.

Table 5: Number of messages transmitted in the three schemes (t: the number of smart meters.).

	Messages sent from smart meters to the gateway	Messages sent from the gateway to the data repository	Messages sent from the data repository to the user	Total
Construction based on Chinese remainder theorem (with data aggregation)	t	1	1	$t + 2$
Construction based on Lagrange interpolation function (with data aggregation)	t	1	1	$t + 2$
The benchmark construction (without data aggregation)	t	t	t	$3t$

Table 6: Lengths of messages transmitted in the three schemes (Unit: bit; t: the number of smart meters.).

	Messages sent from smart meters to the gateway	Messages sent from the gateway to the data repository	Messages sent from the data repository to the user	Total
Construction based on Chinese remainder theorem (with data aggregation)	$480t$	480	480	$480t + 960$
Construction based on Lagrange interpolation function (with data aggregation)	$480t$	480	480	$480t + 960$
The benchmark construction (without data aggregation)	$480t$	$480t$	$480t$	$1440t$

There are three kinds of messages, namely the messages sent from smart meters to the gateway, the messages sent from the gateway to the data repository, and the messages sent from the data repository to the user.

Similar to the evaluation of computation cost, we compare the communication costs of the three schemes. They are Chinese-Remainder-Theorem-based construction with data aggregation, Lagrange-interpolation-function-based construction with data aggregation, and the benchmark construction without data aggregation. Information transmitted in these messages includes m_i, τ_i, π, and Ω. In our experiment, we used the 160-bit elliptic curve. In this case, $m_i \in Z_p$ and $\pi \in Z_p$ are integers whose lengths are $160 - \text{bit}$, while τ_i and Ω are points on the curve whose length are $320 - \text{bit}$. Assuming there are t smart meters, we can get the number of messages and lengths of messages transmitted in the three schemes as shown in Tables 5 and 6, respectively.

From Tables 5 and 6, we can draw the following conclusions:

(1) The number of messages and lengths of messages sent from smart meters to the gateway are linear to the number of smart meters, and the data aggregation process does not reduce the communication costs between smart meters and the gateway.

(2) With data aggregation, the number of messages and the lengths of messages will be independent of the number of smart meters. On the other hand, without data aggregation, they will be linear to the number of smart meters.

(3) When there are a lot of smart meters, the total number of messages with data aggregation is around 1/3 to that without data aggregation. This is because $\lim_{t\longrightarrow\infty}(t + 2/3t) = (1/3)$.

(4) When there are a lot of smart meters, the total lengths of messages with data aggregation is around 1/3 to that without data aggregation. This is because $\lim_{t\longrightarrow\infty}(480t + 960/1440t) = (1/3)$.

In summary, by data aggregating, the communication costs can be reduced significantly. Therefore, DASG can satisfy the communication cost requirement defined in Section 1.

6. Conclusion

In this paper, we have designed a data aggregation protocol for a smart grid called DASG. In DASG, by using the Chinese Remainder theorem and the Lagrange interpolation function, the gateway can aggregate multiple messages signed by different smart meters with different private keys into one short message, and the user can check the integrity of the aggregated short message. By doing so, the computation and communication costs can be reduced significantly. Experimental results show that DASG is feasible for real world applications.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the Science and Technology Project of State Grid Corporation of China (Grant no. 5100-202118566A-0-5-SF).

Supplementary Materials

The online supplementary material file named “security analysis” is the proof for Theorem 1. (*Supplementary Materials*)

References

[1] National Institute of Standards and Technology, *NIST Framework and Roadmap for Smart Grid Interoperability Standards*, 2010, http://www.nist.gov/smartgrid.

[2] Us Department of Energy, *What the Smart Grid Means to Americans*, 2008, https://www.doe.gov/sites/prod/files/oeprod/%20DocumentsandMedia/ConsumerAdvocates.pdf.

[3] V. Tudor, M. Almgren, and M. Papatriantafilou, “Analysis of the impact of data granularity on privacy for the smart grid,” in *Proceedings of the 12th ACM Workshop Workshop Privacy Electron. Soc*, p. 6170, Berlin, Germany, November 2013.

[4] U. Das and V. Namboodiri, “A quality-aware multi-level data aggregation approach to manage smart grid AMI traffic,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 2, pp. 245–256, 2019.

[5] D. Bernaudo, “SmartGrid/AEIC AMI Interoperability Standard Guidelines for ANSI C12.19/IEEE 1377/MC12.19 End Device Communications and Supporting Enterprise Devices, Networks and Related Accessories, the Association of Edison Illuminating Companies,” *Meter and Service Technical Committee report ver. 2*, vol. 1, 2010, https://www.smartgrid.gov/files/AEIC_Guidelines_v20_SmartGridAEIC_AMI_Interoperability_Stdar.pdf.

[6] S.-H. Wang, K. Muhammad, J. Hong, A. K. Sangaiah, and Y.-D. Zhang, “Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization,” *Neural Computing & Applications*, vol. 32, no. 3, pp. 665–680, 2020.

[7] W. Luan, D. Sharp, and S. Lancashire, “Smart grid communication network capacity planning for power utilities,” in *Proceedings of the Transmiss. Distrib. Conf. Expo*, vol. 1, IEEE, New Orleans, LA, USA, April 2010.

[8] Yu-D. Zhang, S. C. Satapathy, D. S. Guttery, J. M. Górriz, and S.-H. Wang, “Improved breast cancer classification through combining graph convolutional network and convolutional neural network,” *Information Processing & Management*, vol. 58, no. 2, Article ID 102439, 2021.

[9] S. McHann, “Grid analytics: how much data do you really need?” in *Proceedings of the Rural Electric Power Conf*, p. C31C34, IEEE, Stone Mountain, GA, USA, April 2013.

[10] T. He, L. Gu, L. Luo, T. Yan, J. Stankovic, and S. Son, “An overview of data aggregation architecture for real-time tracking with sensor networks,” in *Proceedings of the 20th Int. Parallel Distrib. Process. Symp*, vol. 1, IEEE, Rhodes, Greece, April 2006.

[11] S. Ruj and A. Nayak, “A decentralized security framework for data aggregation and access control in smart grids,” *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 196–205, 2013.

[12] F. B. Hildebrand, *Introduction to Numerical Analysis*, Dover, New York, 1974.

[13] J. Alfred, P. C. Menezes, V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, vol. 1, CRC Press, Boca Raton, Florida, 2000.

[14] K. Nahrstedt and T. T. Abdelzaher, “privacy-preserving data aggregation in wireless sensor networks,” in *Proceedings of the Infocom IEEE International Conference on Computer Communications*, vol. 5, no. 6, pp. 2045–2053, IEEE, Valencia, Spain, September 2007.

[15] M. M. Groat, H. E. Wen-bo, and S. Forrest, “KIPDA,” *K-indistinguishable privacy-preserving data aggregation in wireless sensor networks*, vol. 28, no. 6, pp. 2024–2032, 2011.

[16] M. Joye and B. Libert, “A scalable scheme for privacy-preserving aggregation of time-series data,” in *Proceedings of the Financial Cryptography and Data Security - 17th International Conference*, Okinawa, Japan, April 2013.

[17] 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, 2017.

[18] S. H. Wang, D. R. Nayak, D. S. Guttery, X. Zhang, and Y. D. Zhang, “COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis,” *Information Fusion*, vol. 68, pp. 131–148, 2021.

[19] D. T. T. Chan, J. Lee, Y. R. Jong, and Q. S. Tony, “Quek. Privacy preservation with channel-based jamming for data aggregation in Smart Grids,” in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)IEEE ICC*, IEEE, Paris, France, May 2017.

[20] R. Bista, Y. K. Kim, M. S. Song, and J. W. Chang, “Improving data confidentiality and integrity for data aggregation in wireless sensor networks,” *IEICE - Transactions on Info and Systems*, vol. E95-D, no. 1, pp. 67–77, 2012.

[21] D. George, C. Fournet, M. Kohlweiss, and S. Zanella-bguelin, “Smart meter aggregation via secret-sharing,” *Acm Workshop on Smart Energy Grid Security*, vol. 1, pp. 75–80, 2013.

[22] Y. A. Wael, H. Wu, and S. Salil, “Kanhere. Reliable and Secure End-To-End Data Aggregation Using Secret Sharing in WSNs,” in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, San Francisco, CA, USA, March 2017.

[23] C. Li, R. Lu, H. Li, L. Chen, and J. Chen, “PDA: a privacy-preserving dual-functional aggregation scheme for smart grid communications,” *Security and Communication Networks*, vol. 8, no. 15, pp. 2494–2506, 2015.

[24] A. Abdallah and X. S. Shen, “A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid,” *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396–405, 2018.

[25] L. Chen, R. Lu, and Z. Cao, “PDAFT: a privacy-preserving data aggregation scheme with fault tolerance for smart grid communications,” *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1122–1132, 2015.

[26] D. Westhoff, J. Girao, and M. Acharya, “Concealed data aggregation for reverse multicast traffic in sensor networks: encryption, key distribution, and routing adaptation,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1417–1431, 2006.

[27] S. Rahnama, J. D. Bendtsen, J. Stoustrup, and H. Rasmussen, “Robust aggregator design for industrial thermal energy

storages in smart grid," *IEEE Transactions on Smart Grid*, vol. 99, pp. 1–15, 2015.

[28] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, "The role of aggregators in smart grid demand response markets," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1247–1257, 2013.

[29] Z. Guan, G. Si, X. Zhang et al., "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[30] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Advances in Cryptology-CRYPTO 2001, LNCS*, vol. 2139, pp. 213–229, 2001.

[31] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions Fundamentals*, vol. 5, pp. 1234–1243, 2001.

[32] B. Dan, B. Lynn, and H. Shacham, *Short Signatures from the Weil Pairing, International Conference on the Theory and Application of Cryptology and Information Security*, Springer, Berlin, Heidelberg, 2001.

[33] M. Bellare and P. Rogaway, *ACM Conference on Computer and Communications Security*, pp. 62–73, Springer, Berlin, Heidelberg, 1993.

[34] Openssl org, *openssl-1.0.1e.tar.gz*, 2013, http://www.openssl.org/source/.

[35] B. Lynn: 2006, http://crypto.stanford.edu/pbc/manual/ PBC Library Manual 0.5.11.

[36] Standards for Efficient Cryptography Group (Secg), *SEC 1: Elliptic Curve Cryptography*, vol. 20, 2000.

[37] J. Zhang, C. Wan, C. Zhang, X. Guo, and T. Lu, "Auditing images collected by sensors in ambient intelligence systems with privacy and high efficiency," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 12771–12789, 2021.