IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 27, 2019, accepted March 4, 2019, date of publication March 7, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903533

Efficient Privacy-Preserving Multi-Dimensional Data Aggregation Scheme in Smart Grid

YANG MING[1], (Member, IEEE), XUANYI ZHANG[1], AND XIAOQIN SHEN[2]
[1]School of Information Engineering, Chang'an University, Xi'an 710064, China
[2]School of Sciences, Xi'an University of Technology, Xi'an 710054, China

Corresponding author: Yang Ming (yangming@chd.edu.cn)

This work was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2018JM6081, in part by the Project of Science and Technology of Xi'an City under Grant 2017088CG/RC051(CADX002), and in part by the Fundamental Research Funds for the Central Universities, CHD, under Grant 300102249204.

ABSTRACT Smart grid, characterized by high efficiency, security, and flexibility, is gradually replacing the traditional power grid. Data aggregation technology is frequently used to avoid user privacy disclosure as a result of power consumption data transmission in the smart grid. However, traditional one-dimensional data aggregation schemes fail to meet the demands of fine-grained analysis. Therefore, this paper proposes an efficient privacy-preserving multi-dimensional data aggregation (P^2MDA) scheme in smart grid by virtue of homomorphic encryption and superincreasing sequence. The security analysis indicates that the proposed scheme is proved to be secure in the random oracle model, satisfying all security and privacy requirements. The extensive performance analysis shows that in comparison to the related schemes, the proposed scheme achieves lowest computation and communication costs, thus appropriate for practical applications.

INDEX TERMS Smart grid, multidimensional data aggregation, homomorphic encryption, privacy preserving.

I. INTRODUCTION

While the popularity of electricity has greatly facilitated human life, it faces many challenges. For example, in North America, more than 5 million people plunged into darkness for 12 hours due to grid failures when 4.8 GW of electricity in the grid stopped working [1]. In Europe, the temporary closure of a transmission line paralyzes the entire grid, resulting in about 10 million people with no access to electricity [2]. Obviously, these grid accidents indicate the incompetence of the traditional power grid in terms of current social development. Therefore, smart grid (SG) has gradually appeared in the research field [3]–[7]. Figure 1 illustrates the smart grid model, which comprises the market, control center, service provider, energy generation, transmission, distribution, and customer [8], [9]. In the customer part, the smart meter (SM) is responsible for collecting user's power data in real time and transmitting it to control center and service provider through a secure two-way channel. Unlike the traditional power grid's one-way communication, the two-way communication of smart grid enables users to get their bills in real time and use electrical equipment reasonably [10]. The service provider is responsible for all the third-party services requirement in SG, such as collecting power consumption data for analysis and processing, and effectively predicting the peak of power consumption. In addition, the control center dynamically distributes power for the purpose of ensuring the safe operation of the grid based on the analysis results of the service provider [11].

Smart meter collects data at intervals of 10-15 minutes [12], leading to a large amount of data in the communication process. Therefore, the limited computing power of smart meter inevitably causes processing delay and inefficiency. This could provide potential criminals with opportunity to steal user privacy contained in the data collected by smart meter. In order to solve the above problems, researchers have employed cryptographic techniques to process electricity data so as to protect user privacy. Traditional schemes [13]–[20] are mostly based on homomorphic encryption technology [21], with the advantages of reducing the work load of smart meter by aggregating power consumption data, thereby protecting user privacy from being stolen during transmission. Nonetheless, a common drawback of these schemes are that only the overall power consumption data within

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Zhou.

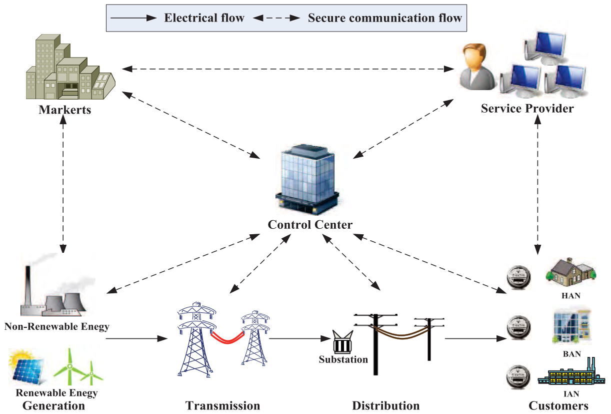


FIGURE 1. **Smart grid model.**

the aggregate range can be obtained. That means the specific power consumption of each device or each user cannot be calculated. Given such a defect, multi-dimensional data aggregation schemes in SG have been proposed [22]–[34], which could not only improve the smart grid data aggregation but also meet the requirements of fine-grained analysis.

In order to reduce computation and communication costs, an efficient multi-dimensional data aggregation scheme is proposed in this paper, referred to as P^2MDA. Our contributions are as follows:

- The P^2MDA scheme we presented employs the super-increasing sequence [22] and the ElGamal encryption technology [35], so SM could classify power consumption data based on powered devices and thus achieve multi-dimensional aggregation.
- The security analysis indicates that P^2MDA is provably secure in the random oracle model. In addition, the proposed scheme fulfills all security requirements.
- The performance in terms of communication and computation costs is evaluated through quantitative calculations. P^2MDA is more efficient than other schemes because it does not need bilinear pairing and map-to-point hash operation.

The organization of this paper is as follows. Section 2 discusses the related work. The background is introduced in Section 3 and P^2MDA is proposed in Section 4. In Section 5, the security of the proposed scheme is analyzed. Section 6 presents the performance comparisons. Finally, the paper is concluded in Section 7.

II. RELATED WORKS

In SG, the short cycle of users' power consumption data is to ensure its efficiency and stability. Based on these real-time data reflecting users' activities, malicious attackers can analyze users' private habits. Thus, it is of vital importance for researchers to explore effective measures to protect user privacy. Some researchers have initially proposed one-dimensional data aggregation schemes [13]–[20] and multi-dimensional data aggregation schemes [22]–[34] in SG. The advantage of multi-dimensional aggregation lies in its classification of powered devices for aggregation, thus facilitating the fine-grained data analysis of the control center in SG.

Multi-dimensional data aggregation can accomplish the aggregation of two or more types of data. In the traditional one-dimensional data aggregation scheme, only the total amount of user's power consumption data can be uploaded to the control center, so the control center can only obtain the total amount of power used by the user. The multi-dimensional data aggregation scheme classifies and uploads the electricity consumption of different types of electrical appliances in the user's home to the control center, and after the control center obtains the data, the data of the different electrical appliances of the user can be analyzed, thereby completing the power consumption fine-grained analysis. These data have realistic applications in power peak prediction and electricity price setting. At the same time, users can adjust their power consumption strategies in real time according to the power consumption of different appliances.

In 2012, Lu *et al.* [22] proposed the multi-dimensional data aggregation in SG, which required specific information about the total amount of electricity used and the consumption of a certain appliance at a certain time. However, thanks to the same ciphertext as that of Paillier encryption [36], users' power consumption data could be recovered as long as the decryption key is obtained. In addition, based on the Chinese remainder theorem, Jia *et al.* [23] put forward a multi-dimensional data aggregation in SG which could resist human-factor-aware attacks. Lu *et al.* [24] proposed a scheme to support two-dimensional data aggregation using the fractional-order group technology, but the scheme failed to consider the data integrity issue. Utilizing the Chinese remainder theorem and ElGamal homomorphic encryption [35], Sui *et al.* [25] suggested a multi-dimensional data aggregation scheme which adopted the HMAC technology to encrypt data hop-by-hop, so the efficiency is not high. Tahir *et al.* [26], claiming that the above scheme [24] failed to ensure data integrity, proposed an improved scheme which applied hash chain technology to realize data integrity protection. However, the costs of computation and communication rose linearly with the increase in the number of users. By virtue of the Paillier homomorphic encryption [36] and BLS short signature [37], Shen *et al.* [27] presented a multi-dimensional data aggregation scheme which make use of Horner's rule for the purpose of classifying different data. Based on Lagrangian interpolation technology, Bo *et al.* [28] proposed a two-dimensional data aggregation and fault-tolerant scheme in SG. Combining Chinese remainder theorem and Paillier homomorphic encryption [36], Bo *et al.* [29] proposed a multi-dimensional data aggregation scheme which required no trusted third party. Meanwhile, it allowed users to be divided into different groups, thus facilitating multi-dimensional data aggregation. Besides, Li *et al.* [30] proposed a privacy-preserving multi-subset data aggregation scheme. Though the scheme provided two payment modes for SG users, it is not qualified in their data protection. Later, based on BGN homomorphic encryption [38], Bo *et al*. [31] put forward a multi-dimensional data aggregation scheme that employed key-policy attribute-based encryption to enforce fine-grained access control at the dimension level. Furthermore, a fine-grained and fault-tolerant multi-dimensional data aggregation scheme is proposed by Ge *et al*. [32]. The decryption efficiency of this scheme is improved since no bilinear mapping and lambda method is used. Taking into consideration security and efficiency, Rafik *et al*. [33] used ElGamal homomorphic encryption [35] and elliptic curve cryptography [39] to achieve efficient multi-dimensional data aggregation scheme in SG. Using hash-then-homomorphic and Paillier homomorphic encryption, Zhang *et al*. [34] proposed a privacy-preserving communication scheme in 5G smart grid slice and vehicle network.

In summary, as could be seen from the above review, most of the existing data aggregation schemes in SG are based on Paillier homomorphic encryption [36] or ElGamal homomorphic encryption [35] with huge communication cost. By comparison, the proposed scheme in this paper is based on elliptic curve cryptography [39] which could effectively reduce computation and communication costs. In addition, it is also capable of multi-dimensional data aggregation and fine-grained analysis.

III. BACKGROUND

In this section, system model, security requirement, design goal, elliptic curve and security assumption are described respectively.

A. SYSTEM MODEL

As demonstrated in Figure 2, the system model consisted of third trust party (TTP), control center (CC), gateway (GW), and smart meter (SM). In our scheme, we mainly focuses on efficient privacy-preserving and multi-dimensional data aggregation, increasing the computational speed of smart meter and gateway, and saving communication bandwidth between smart meter and gateway, gateway and control center.

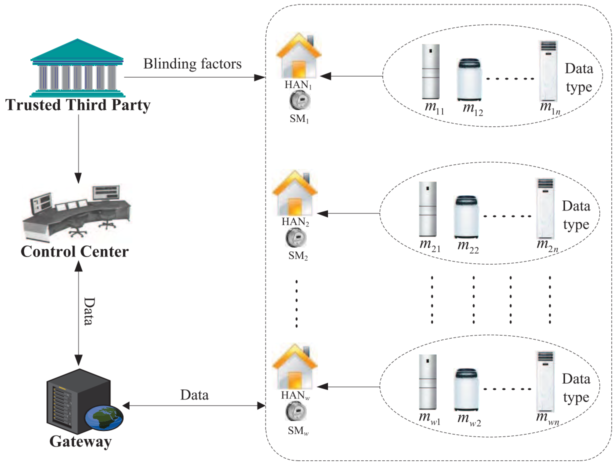


FIGURE 2. System model.

1) TRUSTED THIRD PARTY (TTP)

The trusted third party is a trusted entity, who is in charge of generating the blinding factor for the smart meter and transmitting the sum of the blinding factors to the CC.

2) CONTROL CENTER (CC)

The control center is responsible for generating system parameters, completing the registration of the gateway and smart meter. Meanwhile the CC analyzes the aggregated data transmitted by the GW, and acquires the power consumption of various type of electrical equipment in the smart grid.

3) GATEWAY (GW)

The gateway manages a large number of smart meters, authenticates the legitimacy of the data transmitted by the smart meter and aggregates the encrypted data. After that, GW sends the aggregated and encrypted power data to the CC through a secure channel.

4) SMART METER (SM)

The smart meter collects the power consumption data of each user's household electrical equipment, including refrigerator power data, washing machine data, air conditioner power data, and so on. Then, SM encrypts all kinds of collected data and uploads it to the GW after a short period of time.

B. SECURITY REQUIREMENTS

A secure data aggregation scheme should satisfy the following security requirements: integrity, privacy-preserving, confidentiality, authentication, and resistance against attacks. In our system model, TPP and SM are considered as trustable, CC and GW as ''honest-and-curious''. In smart grid, the user privacy and data integrity protection take the highest priority among all security requirements. In addition, attackers often launch various attacks such as modification attack, replay attack, impersonation attack, internal attack, and man-in-the-middle attack. Therefore, our system is supposed to fulfill the following security requirements.

1) INTEGRITY

Data integrity refers to the accuracy and reliability of data, ensuring that user data is not tampered with or corrupted by attackers. Given that all messages are transmitted in public channels, malicious attackers might utilize them to break regular transactions. A wrong message can not only be accepted and analyzed by other users, but also threaten the security of the entire smart grid. Therefore in order to guarantee data integrity, a secure data aggregation scheme needs to be capable of detecting any change in the data.

2) PRIVACY

The protection of user privacy concerning electricity usage information is crucial in smart grid. Smart meters frequently upload power consumption data, so there is a great risk of leaking users' private information during communication. Attackers can obtain users' private habits and the number of electrical devices after stealing their power usage information. Therefore, a secure data aggregation scheme should ensure the anonymity of users and the privacy of their data. It has to make sure that attackers could not match the consumption data with the specific user even if the data is leaked.

3) CONFIDENTIALITY

The electricity usage data belongs to user privacy, these data reflect the real-time power usage of the user's home. Once leaked during the communication process, the data would be used by malicious attackers to commit crimes. Undoubtedly, users' data confidentiality should be maintained by a secure data aggregation scheme to prevent attackers from exploiting any private information even if with a ciphertext.

4) AUTHENTICATION

Any electrical equipment should be verified for identity legality before joining the smart grid. In response to the verification of user identity, the control center in smart grid is responsible for authenticating the legitimacy of users and their information. Considering that attackers could paralyze the entire smart grid by sending illegal messages or disguising legitimate users, a secure data aggregation scheme must ensure that all users and their private information are legitimate.

5) RESISTANCE AGAINST ATTACKS

Being an open network, smart grid is vulnerable to various attacks, such as modification attack, replay attack, impersonation attack, internal attack, and man-in-the-middle attack. Hence, a secure data aggregation scheme must possess the ability to resist different attacks so as to ensure the security of communication.

6) MODIFICATION ATTACK

This type of attack occurs when an attacker illegally modifies messages to cause the malfunction of smart grid. The key to preventing such attack is to ensure the confidentiality of key and ciphertext.

7) REPLAY ATTACK

Under such attack, a legitimate message transmission is maliciously repeated or delayed. It is either carried out at the source of a legitimate message or retransmitted after the attacker intercepts the message. This attack can disrupt the authentication process without knowing the specific content of the message and effects transmission delay and communication bandwidth.

8) IMPERSONATION ATTACK

An attacker mounting this attack utilizes the data of legitimate users to communicate in smart grid. The attacker steals the identity information of legitimate users and delivers malicious messages as those legitimate users, thus affecting the normal communication in smart grid. Futhermore, the attacker can even remove data protection measures, and exposes the privacy of all users in smart grid.

9) INTERNAL ATTACK

This kind of attack happens when an internal attacker in smart grid collects users' electrical data regularly and legally, then analyzes these data to obtain the users' electricity usage habits and private information [40]–[42]. Because of the internal attackers are legitimate users, even if such attacks occur, they are not easily perceived by other users.

10) MAN-IN-THE-MIDDLE ATTACK

This is an attack where an attacker (man-in-the-middle) actively eavesdrops on users' legitimate communication and relay messages between two users to make them believe that they are connecting with each other when in fact they are communicating with the attacker, resulting in the disclosure of messages. In addition, the attacker can also copy messages during the user's communication process, and obtain the user's private information.

C. DESIGN GOAL

As indicates above through the security requirements and system model, the design goal of P^2MDA is to satisfy the following three parts:

1) EFFICIENCY

Limited computing and communication capacity of smart meter requires the proposed scheme add no significant communication and computation costs to it when implementing functions. Therefore, the goal is to design an efficiency data aggregation scheme avoiding bilinear pairing and map-to-point hash operations.

2) SECURITY

A secure data aggregation scheme should take into account various security threats. If the security of smart grid users is not guaranteed, their privacy will be leaked and the usage data be modified. Therefore, P^2MDA is designed to meet the demands of confidentiality, authentication, data integrity and resistance to various malicious attacks.

3) MULTI-DIMENSIONAL

For the purpose of conducting fine-grained analysis, P^2MDA enables the control center to know clearly the number of users and various types of power consumption data within the scope of its management.

D. ELLIPTIC CURVE

The concept of elliptic curve cryptography (ECC) is first proposed by Millier [39]. F_p is assumed as a finite field with a large prime q. The elliptic curve E over F_p is defined as $y^2 = x^3 + ax + b \ (mod\ p)$ where $a, b \in F_p$ and $\Delta = 4a^3+27b^2 \neq 0$. An additive group $\mathbb{G}$ is formed by the elliptic curve E under the operation of point addition $P + Q = R$. And the scalar multiplication operation over F_p is expressed as $kP = P + P + \cdots + P \ (k\ times)$.

E. SECURITY ASSUMPTION

1) ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (ECDLP)

Given two random point elements P,Q in $\mathbb{G}$ and an elliptic curve E, find an integer $a \in \mathbb{Z}_q^*$ such that $Q = aP$.

2) ELLIPTIC CURVE DISCRETE LOGARITHM ASSUMPTION (ECDLA)

No polynomial-time algorithm could solve the ECDL problem with non-negligible probability.

IV. THE PROPOSED SCHEME

In this section described is an efficient privacy-preserving multi-dimensional data aggregation scheme in smart grid, comprising five phases: initialization, registration, user data generation, data aggregation, and data reading. The notations are listed in Table 1.

TABLE 1. Notations.

Symbol	Definition
$\mathbb{G}$	An additive group
P	A generator of $\mathbb{G}$
H_1, H_2, H_3	Hash functions: $H_1, H_2, H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$
TTP	Trusted third party
CC	Control center
GW	Gateway
SM_i	Smart meter i
w	Number of users
n	Number of data types
d	Maximum value of data

A. INITIALIZATION

All system parameters are generated by CC, and the blinding factors are produced by TTP. Specifically w is assumed as the number of users in the system, n as the total types of electricity usage data for one user to be aggregated, and the value of each type is less than the constant d.

The following steps are executed by CC to generate the system parameters:

(1) CC generates a group $\mathbb{G}$ of the prime order q based on an elliptic curve E defined over a finite field F_p where $P \in \mathbb{G}$, serving as a generator.

(2) CC chooses $x \in \mathbb{Z}_q^*$ and computes $P_{pub} = xP$.

(3) CC chooses hash functions: $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.

(4) CC chooses a superincreasing sequence $\vec{a} = (a_1, a_2, \cdots, a_n)$, where $a_1, a_2, \cdots, a_n$ are big prime numbers such that $\sum_{j=1}^{i-1} a_j \cdot w \cdot d < a_i$ for $i = 1, 2, \cdots, n$ and $\sum_{i=1}^{n} a_i \cdot w \cdot d < q$.

(5) CC outputs the system parameters $\{p, q, \mathbb{G}, P, P_{pub}, H_1, H_2, H_3, \vec{a}\}$

TTP executes the following steps to produce the blinding factors:

(1) TTP randomly chooses a group of large numbers $k_1, k_2, \cdots, k_w \in \mathbb{Z}_q^*$ and calculates $k = \sum_{i=1}^{w} k_i$.

(2) TTP returns k to CC and k_i to each SM_i via secure channels, where $i = 1, 2, \cdots, w$.

B. REGISTRATION

In this phase, GW and all SM_i register with CC respectively. The detailed steps are as follows:

(1) SM_i randomly chooses $x_i \in \mathbb{Z}_q^*$ and $r_i \in \mathbb{Z}_q^*$, computes the public key $X_i = x_iP$ and a signature of knowledge signature $< R_i, s_i >$, where $R_i = r_iP$, $s_i = r_i + x_iH_1(ID_i, X_i, R_i)$. After that SM_i, returns $< ID_i, X_i, R_i, s_i >$ to CC.

(2) After receiving $< ID_i, X_i, R_i, s_i >$, CC checks if $R_i = s_iP - H_1(ID_i, X_i, R_i)X_i$ holds. Then, it publishes $< ID_i, X_i, R_i, s_i >$.

(3) GW chooses $x_{GW} \in \mathbb{Z}_q^*$ and $r_{GW} \in \mathbb{Z}_q^*$ at random, to compute the public key $X_{GW} = x_{GW}P$ and a signature of knowledge signature $< R_{GW}, s_{GW} >$, where

$$R_{GW} = r_{GW}P,$$
$$s_{GW} = r_{GW} + x_{GW}H_1(ID_{GW}, X_{GW}, R_{GW}).$$

Later, GW returns $< ID_{GW}, X_{GW}, R_{GW}, s_{GW} >$ to CC.

(4) After receiving $< ID_{GW}, X_{GW}, R_{GW}, s_{GW} >$, CC checks if

$$R_{GW} = s_{GW}P - H_1(ID_{GW}, X_{GW}, R_{GW})X_{GW}$$

holds. Finally, CC publishes $< ID_{GW}, X_{GW}, R_{GW}, s_{GW} >$.

C. USER DATA GENERATION

During this phase, each smart meter SM_i measures and generates n types of electricity consumption data $(m_{i1}, m_{i2}, \cdots, m_{in})$ and transmits them to GW. The following are specific steps

(1) SM_i randomly chooses $t_i \in \mathbb{Z}_q^*$ and computes the ciphertexts

$$C_{1,i} = t_iP,$$
$$C_{2,i} = t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in}) + H_2(T)k_i]P.$$

(2) SM_i randomly chooses $l_i \in \mathbb{Z}_q^*$ and calculates the signature

$$L_i = l_iP,$$
$$v_i = l_i + x_iH_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T).$$

where T denoted the current timestamp.

(3) Finally, SM_i returns $< C_{1,i}, C_{2,i}, ID_i, T, L_i, v_i >$ to GW.

D. DATA AGGREGATION

After receiving total w electricity consumption data $< C_{1,i}, C_{2,i}, ID_i, T, L_i, v_i >$ of w users from $SM_i(i = 1, 2, \cdots, w)$, GW performs the following steps

(1) GW first examines the timestamp T then computes and verifies if

$$v_iP = L_i + H_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)X_i$$

holds for each $i = 1, 2, \cdots, w$. Small exponent test technology [43] is employed by GW while performing the batch verification so as to increase speed. GW randomly selects a group of small numbers $\theta_1, \theta_2, \cdots, \theta_w \in [1, 2^w]$ to checks if

$$\begin{aligned}&(\sum\nolimits_{i=1}^{w} \theta_i v_i)P\\ &= \sum\nolimits_{i=1}^{w} \theta_i L_i + \sum\nolimits_{i=1}^{w} \theta_i H_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)X_i.\end{aligned}$$

(2) After successful verification of smart meters' signatures, GW computes the aggregation data

$$C_1 = \sum\nolimits_{i=1}^{w} C_{1,i}, C_2 = \sum\nolimits_{i=1}^{w} C_{2,i}.$$

(3) GW randomly chooses $l_{GW} \in \mathbb{Z}_q^*$ then calculates

$$\begin{aligned}L_{GW} &= l_{GW}P,\\ v_{GW} &= l_{GW} + x_{GW}H_3(ID_{GW}, X_{GW}, C_1, C_2, L_{GW}, T).\end{aligned}$$

Finally, GW returns the data $< C_1, C_2, ID_{GW}, T, L_{GW}, v_{GW} >$ to CC.

E. DATA READING

Upon receiving $< C_1, C_2, ID_{GW}, T, L_{GW}, v_{GW} >$, CC performs the following procedures to recover the aggregated data

(1) CC checks the timestamp T and computes

$$v_{GW}P = L_{GW} + H_3(ID_{GW}, X_{GW}, C_1, C_2, L_{GW}, T)X_{GW}.$$

(2) CC utilizes the private key x and the blinding factor k to compute $\Phi = C_2 - xC_1 - H_2(T)kP$.

(3) CC computes $M = a_1 \sum_{i=1}^{w} m_{i1} + a_2 \sum_{i=1}^{w} m_{i2} + \cdots + a_n \sum_{i=1}^{w} m_{in}$ by solving the discrete log of Φ with the base P using the Pollard's lambda algorithm [40] with the time complexity $O(\sqrt{w \cdot n \cdot d})$.

By invoking the **Algorithm 1**, CC can recovers the aggregated data $(D_1, D_2, \cdots, D_n)$, where each $D_j = \sum_{i=1}^{w} m_{ij}$.

Correctness

$$\begin{aligned}\Phi &= C_2 - xC_1 - H_2(T)kP\\ &= \sum\nolimits_{i=1}^{w} C_{2,i} - x\sum\nolimits_{i=1}^{w} C_{1,i} - H_2(T)kP\\ &= \sum\nolimits_{i=1}^{w} (t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in})\\ &\quad + H_2(T)k_i]P) - x\sum\nolimits_{i=1}^{w} t_iP - H_2(T)kP\\ &= \sum\nolimits_{i=1}^{w} t_iP_{pub} + (a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_1\sum\nolimits_{i=1}^{w} m_{i1} +\\ &\quad \cdots + a_n\sum\nolimits_{i=1}^{w} m_{in})P + \sum\nolimits_{i=1}^{w} H_2(T)k_iP\\ &\quad - \sum\nolimits_{i=1}^{w} t_ixP - H_2(T)kP\\ &= \sum\nolimits_{i=1}^{w} t_iP_{pub} + (a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_1\sum\nolimits_{i=1}^{w} m_{i1} +\\ &\quad \cdots + a_n\sum\nolimits_{i=1}^{w} m_{in})P + H_2(T)kP - \sum\nolimits_{i=1}^{w} t_iP_{pub}\\ &\quad - H_2(T)kP\\ &= \left(a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_2\sum\nolimits_{i=1}^{w} m_{i2} + \cdots + a_n\sum\nolimits_{i=1}^{w} m_{in}\right)P\end{aligned}$$

Algorithm 1 Recovery All Aggregated Data

```
Input: superincreasing sequence $\vec{a} = (a_1, a_2, \cdots, a_n)$ and $M$
Output: $D_j$ for $j = 1, 2, \cdots, n$
begin:
    set $X = M$
    for $j = n$ to 1 do
        $D_j = \frac{X - X \bmod a_j}{a_j}$
    end
    return $(D_1, D_2, \cdots, D_n)$
end
```

From the **Algorithm 1**, we obtain

$$\begin{aligned}X = M = a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_2\sum\nolimits_{i=1}^{w} m_{i2} + \cdots\\ + a_{n-1}\sum\nolimits_{i=1}^{w} m_{i(n-1)} + a_n\sum\nolimits_{i=1}^{w} m_{in}.\end{aligned}$$

Since any type of data is less than a constant n, we have

$$\begin{aligned}&a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_2\sum\nolimits_{i=1}^{w} m_{i2} + \cdots + a_{n-1}\sum\nolimits_{i=1}^{w} m_{i(n-1)}\\ &< a_1\sum\nolimits_{i=1}^{w} d + a_2\sum\nolimits_{i=1}^{w} d + \cdots + a_{n-1}\sum\nolimits_{i=1}^{w} m_{i(n-1)}\\ &= \sum\nolimits_{j=1}^{n-1} a_jwd\\ &< a_n.\end{aligned}$$

Therefore,

$$\begin{aligned}X \bmod a_n = a_1\sum\nolimits_{i=1}^{w} m_{i1} + a_2\sum\nolimits_{i=1}^{w} m_{i2} + \cdots\\ + a_{n-1}\sum\nolimits_{i=1}^{w} m_{i(n-1)},\end{aligned}$$

$$D_n = \frac{X - X \bmod a_n}{a_n} = \sum\nolimits_{i=1}^{w} m_{in}.$$

With the similar procedure, we can obtain $D_j = \sum_{i=1}^{w} m_{ij}$ for $j = 1, 2, \cdots, n - 1$.

V. SECURITY

The security model of P^2MDA is introduced in this section and adequately the proposed scheme is proved to be secure in the random oracle model. Finally, comparisons are made between the security of P^2MDA and that of other schemes.

A. SECURITY MODEL

A secure multi-dimensional data aggregation scheme in SG must satisfy the requirements of confidentiality and unforgeability, formally defined by two games executed by an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$. The following queries can be made by the attacker $\mathcal{A}$.

- **Hash** H_1, H_2, H_3 **query**: A query is given, and a random value is returned.
- **Create** SM_i **query**: $\mathcal{A}$ makes a query on the identity ID_i of SM_i; $\mathcal{C}$ produces the corresponding blinding factor, public key and private key for SM_i.
- **Corrupt** SM_i **query**: $\mathcal{A}$ makes a query on the identity ID_i of SM_i; $\mathcal{C}$ sends the corresponding private key to $\mathcal{A}$.
- **Signctypt query**: $\mathcal{A}$ makes a query on the message m_i under the identity ID_i of SM_i; $\mathcal{C}$ returns the corresponding ciphertext C to $\mathcal{A}$.
- **Unsignctypt query**: $\mathcal{A}$ makes a query on the ciphertext C under the identity ID_i of SM_i; $\mathcal{C}$ decrypts the ciphertext and returns the corresponding message m_i to $\mathcal{A}$.

The confidentiality (indistinguishability under the chosen plaintext attack (IND-CPA)) is defined through the following game played between an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$.

- **Initialization**:$\mathcal{A}$ chooses a challenging identity ID_i^*, and returns ID_i^* to $\mathcal{C}$.
- **Setup**: $\mathcal{C}$ returns the system parameters to $\mathcal{A}$ after generating the master key and system parameters.
- **Phase 1**: $\mathcal{A}$ properly makes the Hash H_1, H_2, H_3 queries, Create SM_i queries, Corrupt SM_i queries, and Signcrypt queries for polynomial bounded times.
- **Challenge**: After finishing Phase 1, $\mathcal{A}$ chooses two messages $m_0^* = (m_{01}^*, m_{02}^*, \cdots, m_{0n}^*)$ and $m_1^* = (m_{11}^*, m_{12}^*, \cdot\cdot\cdot, m_{1n}^*)$ with the same length, and then calculates $M_0^* = a_1m_{01}^* + a_2m_{02}^* + \cdots + a_nm_{0n}^*, M_1^* = a_1m_{11}^* + a_2m_{12}^* + \cdots + a_nm_{1n}^*$. $\mathcal{A}$ sends (M_0^*, M_1^*) to $\mathcal{C}$ who randomly selects $b \in \{0, 1\}$, generates the C_b^* of the message M_b^* and returns C_b^* to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ properly makes the queries as in Phase 1 except the Corrupt query on ID_i^*
- **Guess**: $\mathcal{A}$ outputs the guess $b' \in \{0, 1\}$. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{IND-CPA} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 1 (Confidentiality): The proposed scheme is IND-CPA security if there is no polynomial-time attacker who could win the aforementioned game with a non-negligible advantage.

The unforgeability (existential unforgeability against adaptive chosen message attacks (EUF-CMA)) is defined through the following game played between an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$.

- **Initialization** $\mathcal{A}$ chooses a challenging identity ID_i^*, and returns ID_i^* to $\mathcal{C}$.
- **Setup**: $\mathcal{C}$ returns the system parameters to $\mathcal{A}$ after generating the master key and system parameters.
- **Queries**: $\mathcal{A}$ is allowed to make Hash H_1, H_2, H_3 queries, Create SM_i queries, Corrupt SM_i queries, Signcrypt queries, and Unsigncrypt queries but not Corrupt SM_i query and Unsigncrypt query on the challenging identity ID_i^*.
- **Forgery**: $\mathcal{A}$ outputs a ciphertext $(C_{1,i}, C_{2,i})$ on m_i under ID_i^*, such that
  - $(C_{1,i}, C_{2,i})$ is a valid ciphertext on m_i under ID_i^*.
  - ID_i^* has never been requested in the Corrupt SM_i queries.

Definition 2 (Unforgeability): The proposed scheme is EUF-CMA security if there is no polynomial-time attacker who could win the aforementioned game with a non-negligible probability.

B. SECURITY PROOF

Theorem 1: The proposed scheme is secure against IND-CPA if ElGamal encryption is secure against the indistinguishability under the chosen plaintext attack.

Proof: Suppose a polynomial-time adversary $\mathcal{A}$ wins the game in Definition 1 with a non-negligible advantage ε, then there is an algorithm $\mathcal{B}$ that can break the indistinguishability of ElGamal encryption under chosen plaintext attack.

Initialization: A simulator $\mathcal{S}$ of ElGamal encryption generates the system parameters $\{p, q, \mathbb{G}, P, P_{pub}\}$ and sends them to $\mathcal{B}$. Then $\mathcal{A}$ selects an identity ID_i^* as the challenging identity and returns it to $\mathcal{B}$.

Setup: $\mathcal{B}$ chooses $H_1, H_2, H_3, \vec{a}$ and returns the parameters $\{p, q, \mathbb{G}, P, P_{pub}, H_1, H_2, H_3, \vec{a}\}$ to $\mathcal{A}$. Here, hash functions H_1, H_2, H_3 are considered as random oracles in the proof.

$\mathcal{B}$ maintains the initially empty list as follows to keep the consistency and response:

- H_1 list L_{H_1}: It consists of tuples $(ID_i, X_i, R_i, h_{1,i})$.
- H_2 list L_{H_2}: It consists of tuples (T, h_2).
- H_3 list L_{H_3}: It consists of tuples $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$.
- L_{SM_i}: It consists of tuples $(ID_i, x_i, X_i, s_i, R_i, k_i)$.

Phase 1: The following polynomial bounded times queries are made by $\mathcal{A}$ adaptively.

H_1 **query**: $\mathcal{A}$ makes a query on (ID_i, X_i, R_i), and then $\mathcal{B}$ checks L_{H_1} and executes the next step:

- If L_{H_1} contains $(ID_i, X_i, R_i, h_{1,i})$, $\mathcal{B}$ would extract the values $h_{1,i} = H_1(ID_i, X_i, R_i)$ from L_{H_1} and return it to $\mathcal{A}$.
- If L_{H_1} does not contain $(ID_i, X_i, R_i, h_{1,i})$, $\mathcal{B}$ would randomly choose a number $h_{1,i} \in \mathbb{Z}_q^*$, add $(ID_i, X_i, R_i, h_{1,i})$ in L_{H_1} and return $h_{1,i}$ to $\mathcal{A}$.

H_2 **query**: $\mathcal{A}$ makes a query on T, and then $\mathcal{B}$ checks L_{H_2} and executes the next step:

- If L_{H_2} contains $(T, h_{2,i})$, $\mathcal{B}$ would extract the values $h_{2,i} = H_2(T)$ from L_{H_2} and return it to $\mathcal{A}$.
- If L_{H_2} does not contain $(T, h_{2,i})$, $\mathcal{B}$ would randomly choose a number $h_{2,i} \in \mathbb{Z}_q^*$, add $(T, h_{2,i})$ in L_{H_2} and return $h_{2,i}$ to $\mathcal{A}$.

H_3 **query**: $\mathcal{A}$ makes a query on $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)$, and then $\mathcal{B}$ checks L_{H_3} and takes the next step:

- If L_{H_3} contains $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$, then $\mathcal{B}$ would extract the values $h_{3,i} = (ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)$ from L_{H_3} and return it to $\mathcal{A}$.
- If L_{H_3} does not contain $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$, then $\mathcal{B}$ would randomly choose a number $h_{3,i} \in \mathbb{Z}_q^*$,

add $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$ in L_{H_3} and return $h_{3,i}$ to $\mathcal{A}$.

Create SM_i **query**: $\mathcal{A}$ makes a query on the identity ID_i of SM_i, and then $\mathcal{B}$ checks L_{SM_i} and performs the next step:

- If L_{SM_i} contains $(ID_i, x_i, X_i, s_i, R_i, k_i)$, then $\mathcal{B}$ would extract the values (X_i, R_i) from L_{SM_i} and return them to $\mathcal{A}$.
- If L_{SM_i} does not contain $(ID_i, x_i, X_i, s_i, R_i, k_i)$, then $\mathcal{B}$ would randomly choose $x_i, r_i, h_{1,i}, k_i \in \mathbb{Z}_q^*$, calculate $X_i{=}x_iP$, $R_i = r_iP$, $s_i = r_i + x_ih_{1,i}$, and store $(ID_i, X_i, R_i, h_{1,i})$ and $(ID_i, x_i, X_i, s_i, R_i, k_i)$ into $L_{H_{1,i}}$ and L_{SM_i}, respectively. Finally, $\mathcal{B}$ returns (X_i, R_i) to $\mathcal{A}$.

Corrupt SM_i **query**: $\mathcal{A}$ makes a query on identity ID_i of SM_i, and then $\mathcal{B}$ checks L_{SM_i} and executes the next step:

- If $ID_i = ID_i^*$, $\mathcal{B}$ would abort the game.
- If $ID_i \neq ID_i^*$, $\mathcal{B}$ would check L_{SM_i} for the tuple $(ID_i, x_i, X_i, s_i, R_i, k_i)$ and return (x_i, s_i, k_i) to $\mathcal{A}$.

Signcrypt query: $\mathcal{A}$ makes a query on the message m_i under the identity ID_i of SM_i, and then $\mathcal{B}$ extracts x_i from L_{SM_i} and randomly selects $l_i, h_{3,i}, t_i, k_i \in \mathbb{Z}_q^*$. After that, $\mathcal{B}$ calculates $C_{1,i} = t_iP$, $C_{2,i} = t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in}) + H_2(T)k_i]P$, $L_i = l_iP$, and $v_i = l_i + x_ih_{3,i}$. Finally, $\mathcal{B}$ stores $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$ in $L_{H_{3,i}}$ and returns $(C_{1,i}, C_{2,i}, v_i, L_i, T)$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ randomly picks two same length messages $m_0^* = (m_{01}^*, m_{02}^*, \cdot\cdot\cdot, m_{0n}^*)$, $m_1^* = (m_{11}^*, m_{12}^*, \cdot\cdot\cdot, m_{1n}^*)$ and calculates $M_0^* = a_1m_{01}^* + a_2m_{02}^* + \cdots + a_nm_{0n}^*$, $M_1^* = a_1m_{11}^* + a_2m_{12}^* + \cdots + a_nm_{1n}^*$, and sends (M_0^*, M_1^*) to $\mathcal{B}$. Then $\mathcal{B}$ returns (M_0^*, M_1^*) to the simulator $\mathcal{S}$ of ElGamal encryption. Next, $\mathcal{S}$ randomly selects $b \in \{0, 1\}$ and $t_i \in \mathbb{Z}_q^*$, computes the ciphertext $C_{1,M_b^*} = t_i^*P$, $C_{2,M_b^*} = t_i^*P_{pub} + M_b^*P$, and returns $(C_{1,M_b^*}, C_{2,M_b^*})$ to $\mathcal{B}$. After checking the tuples $(ID_i^*, x_i^*, X_i^*, s_i^*, R_i^*, k_i^*)$ in L_{SM_i} and randomly choosing $l_i^*, T^* \in \mathbb{Z}_q^*$, $\mathcal{B}$ calculates $C_{1,i}^* = C_{1,M_b^*}$, $C_{2,i}^* = C_{2,M_b^*} + H_2(T^*)k_i^*P$, $L_i^* = l_i^*P$, and $v_i^* = l_i^* + H_3(ID_i^*, X_i^*, C_{1,i}^*, C_{2,i}^*, L_i^*, T^*)X_i^*$. At last, $\mathcal{B}$ returns $(C_{1,i}^*, C_{2,i}^*, v_i^*, L_i^*, T^*)$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively makes the queries as in Phase 1 except the Corrupt query on the challenging identity ID_i^*.

Guess: $\mathcal{B}$ outputs b' as the guess against the semantic secure under the chosen plaintext attack against ElGamal encryption.

Probability analysis: For the purpose of evaluating the advantage of $\mathcal{B}$ breaking the indistinguishability, the following three events are defined:

- E_1: $\mathcal{B}$ never abort the game in all Corrupt queries.
- E_2: $\mathcal{B}$'s guess about the value of b is completely correct.
- E_3: $\mathcal{B}$ outputs the message $(C_{1,i}^*, C_{2,i}^*, v_i^*, L_i^*, T^*)$ such that $ID_i^* = ID_i$.

According to the above simulation, $\Pr[E_1] \geq (1 - \frac{1}{q_{H_1}})^{q_{Cor}}$, $\Pr[E_2|E_1] \geq \varepsilon$, and $\Pr[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{H_1}}$ could be obtained, where q_{H_1} and q_{Cor} represent the number of H_1 queries and Corrupt queries respectively. Thus, the advantage of $\mathcal{B}$ breaking the indistinguishability of ElGamal encryption is described as

$$\begin{aligned}\Pr[E_1 \wedge E_2 \wedge E_3] &\geq \Pr[E_3|E_1 \wedge E_2]Pr[E_2|E_1]Pr[E_1]\\ &\geq \frac{1}{q_{H_1}}\varepsilon(1 - \frac{1}{q_{H_1}})^{q_{Cor}}.\end{aligned}$$

Because of the non-negligibility of ε, we know that the $\Pr[E_1 \wedge E_2 \wedge E_3]$ is non-negligible. Above analyses show that $\mathcal{B}$ could break the indistinguishability of ElGamal encryption with a non-negligible advantage, thus implying that P^2MDA is indistinguishable under the chosen plaintext attack.

Theorem 2: The proposed scheme is semantic secure against ciphertext unforgeability under the ECDL assumption.

Proof: Assume an attacker $\mathcal{A}$ can break ciphertext unforgeability of P^2MDA with a non-negligible advantage ε, then an algorithm $\mathcal{B}$ can be constructed to solve the ECDL problem. Given an instance of the ECDL problem as $(P, aP = Q)$, the task of $\mathcal{B}$ is to find an element $a \in \mathbb{Z}_q^*$.

Initialization: $\mathcal{A}$ selects an identity ID_i^* as the challenging identity and returns it to $\mathcal{B}$.

Setup: $\mathcal{B}$ selects $x \in \mathbb{Z}_q^*$ at random, calculates $xP = P_{pub}$, and then returns the system parameters $\{p, q, \mathbb{G}, P, P_{pub}, H_1, H_2, H_3, \vec{a}\}$ to $\mathcal{A}$.

H_1, H_2, H_3 **query**: The same as in Theorem 1.

Create SM_i **query**: $\mathcal{A}$ makes a query on the identity ID_i of SM_i, and then $\mathcal{B}$ checks L_{SM_i} and executes the next step:

- If L_{SM_i} contains $(ID_i, x_i, X_i, s_i, R_i, k_i)$, $\mathcal{B}$ would return (X_i, R_i) to $\mathcal{A}$.
- If L_{SM_i} does not contain $(ID_i, x_i, X_i, s_i, R_i, k_i)$, $\mathcal{B}$ would carry out the next step:
  - If $ID_i{=}ID_i^*$, $\mathcal{B}$ would select $s_i, h_{1,i}, k_i \in \mathbb{Z}_q^*$ randomly, set $X_i{=}aP$, $R_i = s_iP - h_{1,i}X_i$ and then store $(ID_i, X_i, R_i, h_{1,i})$ and $(ID_i, \perp, X_i, s_i, R_i, k_i)$ into $L_{H_{1,i}}$ and L_{SM_i}, respectively. Finally, $\mathcal{B}$ would return (X_i, R_i) to $\mathcal{A}$.
  - If $ID_i \neq ID_i^*$, $\mathcal{B}$ would select $x_i, r_i, h_{1,i}, k_i \in \mathbb{Z}_q^*$ randomly, calculate $X_i{=}x_iP$, $R_i = r_iP$, $s_i = r_i + x_ih_{1,i}$, and then store $(ID_i, X_i, R_i, h_{1,i})$ and $(ID_i, x_i, X_i, s_i, R_i, k_i)$ into $L_{H_{1,i}}$ and L_{SM_i}, respectively. Finally, $\mathcal{B}$ would return (X_i, R_i) to $\mathcal{A}$.

Corrupt SM_i **query**: $\mathcal{A}$ makes a query on the identity ID_i of SM_i, and then $\mathcal{B}$ checks L_{SM_i} and executes the next step:

- If $ID_i = ID_i^*$, $\mathcal{B}$ would abort the game.
- If $ID_i \neq ID_i^*$, $\mathcal{B}$ would check L_{SM_i} for the tuple $(ID_i, x_i, X_i, s_i, R_i, k_i)$.
  - If L_{SM_i} contains $(ID_i, x_i, X_i, s_i, R_i, k_i)$, $\mathcal{B}$ would return (x_i, s_i, k_i) to $\mathcal{A}$.
  - If L_{SM_i} does not contain $(ID_i, x_i, X_i, s_i, R_i, k_i)$, $\mathcal{B}$ would make the Create query on ID_i. After that, $\mathcal{B}$ would return (x_i, s_i, k_i) to $\mathcal{A}$.

Signcrypt query: $\mathcal{A}$ makes a query on the message m_i under the identity ID_i of SM_i, and then $\mathcal{B}$ executes the next step:

- If $ID_i = ID_i^*$, $\mathcal{B}$ would select $v_i, t_i, T \in \mathbb{Z}_q^*$ randomly before calculating $C_{1,i} = t_iP$, $C_{2,i} = t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in}) + H_2(T)k_i]P$, $L_i = v_iP - h_{3,i}X_i$, and

$h_{3,i} = H_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)$. Finally, $\mathcal{B}$ would store $(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T, h_{3,i})$ in $L_{H_{3,i}}$, and return $(C_{1,i}, C_{2,i}, v_i, L_i, T)$ to $\mathcal{A}$.

- If $ID_i \neq ID_i^*$, $\mathcal{B}$ would execute the user data generation algorithm, generate $(C_{1,i}, C_{2,i}, ID_i, T, L_i, v_i)$ and send $(C_{1,i}, C_{2,i})$ to $\mathcal{A}$.

Unsigncrypt query: $\mathcal{A}$ makes a query on the ciphertext $(C_{1,i}, C_{2,i})$ under the identity ID_i of SM_i, and then $\mathcal{B}$ checks L_{SM_i} and performs the next step:

- If $ID_i = ID_i^*$, $\mathcal{B}$ would abort the game.
- If $ID_i \neq ID_i^*$, $\mathcal{B}$ would decrypt the ciphertext to get the message through the master key x.

Forgery: Finally, $\mathcal{A}$ outputs a forged ciphertext $(C_{1,i}, C_{2,i}, v_i, L_i, T)$ under the identity ID_i of SM_i.

- If $ID_i \neq ID_i^*$, $\mathcal{B}$ would abort the game.
- If $ID_i = ID_i^*$, based on the forking lemma [44], $\mathcal{B}$ would output another valid ciphertext $(ID_i, C_{1,i}, C_{2,i}, v_i', L_i, T)$ with a different choice of H_3. Since both ciphertexts are valid, two equations could be devised as follows:

$$v_iP = L_i + X_ih_{3,i},$$
$$v_i'P = L_i + X_ih_{3,i}'.$$

Furthermore,

$$\begin{aligned}(v_i - v'_i)P &= v_iP - v'_iP \\ &= (h_{3,i} - h'_{3,i})X_i \\ &= (h_{3,i} - h'_{3,i})aP.\end{aligned}$$

Finally, $\mathcal{B}$ outputs $a = (v_i - v_i')(h_{3,i} - h_{3,i}')^{-1}$ as the solution to the given ECDL problem.

Probability analysis: After completing the above simulation, the probability of $\mathcal{B}$ solving the ECDL problem is analyzed. Three related events are defined as follows:

- E_1: $\mathcal{B}$ never abort the game in all Unsigncrypt queries and Corrupt queries.
- E_2: $\mathcal{B}$ outputs a valid ciphertext $(C_{1,i}, C_{2,i}, v_i, L_i, T)$ under ID_i.
- E_3: $ID_i^* = ID_i$.

According to the above simulation, $\Pr[E_1] \geq (1 - \frac{1}{q_{H_1}})^{q_{Cor}+q_{Uns}}$, $\Pr[E_2|E_1] \geq \varepsilon$, $\Pr[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{H_1}}$ could be obtained, where q_{H_1}, q_{Cor} and q_{Uns} denote the number of H_1 queries, Create queries and Unsigncrypt queries. Accordingly, the probability of $\mathcal{B}$ solving the ECDL problem could be described as:

$$\begin{aligned}\Pr[E_1 \wedge E_2 \wedge E_3] &\geq \Pr[E_3|E_1 \wedge E_2]Pr[E_2|E_1]Pr[E_1] \\ &\geq \frac{1}{q_{H_1}}\varepsilon(1 - \frac{1}{q_{H_1}})^{q_{Cor}+q_{Uns}}.\end{aligned}$$

Because of the non-negligibility of ε, we know that the $\Pr[E_1 \wedge E_2 \wedge E_3]$ is non-negligible. Based on the above analysis, we conclude that $\mathcal{B}$ could solve the ECDL problem with a non-negligible probability. This is in contradiction with the hardness of ECDL problem. Thus P^2MDA is able to provide unforgeability.

C. ANALYSIS OF SECURITY REQUIREMENT

Here, the security of P^2MDA is analyzed and then compares with other related schemes.

Confidentiality: Theorem 1 implies that CC must calculate equation $\Phi = C_2 - xC_1 - H_2(T)kP$ to obtain users' data. However, with no specific information about the private key x and the blinding factor k, it is impossible for any attacker to obtain users' data. Therefore, P^2MDA would meet the demand of confidentiality.

Authentication: Theorem 2 suggests that no one could generate a correct ciphertext $< C_{1,i}, C_{2,i}, ID_i, T, L_i, v_i >$ without the private key. Beside, the users' identity could be authenticated by GW via verifying whether the equation $(\sum_{i=1}^{w} \theta_iv_i)P = \sum_{i=1}^{w} \theta_iL_i + \sum_{i=1}^{w} \theta_iH_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)X_i$ holds. Consequently, P^2MDA would satisfy the requirement of authentication.

Integrity: In the proposed scheme, the ciphertext $(C_{1,i}, C_{2,i})$ is signed to generate the signature (L_i, v_i). And Theorem 2 indicates that no attacker could generate a valid signature. Thus, any change in the ciphertext $(C_{1,i}, C_{2,i})$ could be detected by checking the signature, suggesting that P^2MDA could ensure the integrity of the ciphertext.

Privacy: In the proposed scheme, SM transports the electricity data to GW which then aggregates and returns the data to CC. Through Algorithm 1, CC decrypts the aggregated data, and recovers the sum of some types of electricity data. During this process, attackers have no information about the (t_i, l_i) and could not extract any specific user's data. Therefore, P^2MDA could protect user privacy.

Resistance against attacks: The proposed scheme is proved to possess the ability to resist modification attack, replay attack, impersonation attack, internal attack, and man-in-the-middle attack.

- Modification attack. Theorem 2 proves that no attacker could forge a legal ciphertext. GW could detect any change in ciphertext based on the equations $(\sum_{i=1}^{w} \theta_iv_i)P = \sum_{i=1}^{w} \theta_iL_i + \sum_{i=1}^{w} \theta_ih_iX_i$ and $vP = R + H_3(ID_{GW}, X_{GW}, C_1, C_2, T)X_{GW}$. Therefore, P^2MDA could resist any modification attack.
- Replay attack. The time stamp T is used in the message $< C_{1,i}, C_{2,i}, ID_i, T, L_i, v_i >$ and $< C_{GW}, C_{GW}, ID_{GW}, T, L_{GW}, v_{GW} >$, where $C_{2,i} = t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in}) + H_2(T)k_i]P$, $v_i = l_i + x_iH_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)$, and $C_2 = \sum_{i=1}^{w} C_{2,i}v_{GW} = l_{GW} + x_{GW}H_3(ID_{GW}, X_{GW}, C_1, C_2, L_{GW}, T)$. GW and CC could detect replay attack by checking T. Therefore, P^2MDA could resist any replay attack.
- Impersonation attack. Theorem 1 shows that no attacker could produce a legal ciphertext without users' private key. GW and CC could also detect impersonation attacks by verifying the legitimacy of the ciphertext. Therefore, P^2MDA is able to resist any impersonation attack.
- Internal attack. Each SM_i embeds the blinding factor k_i in $C_{2,i} = t_iP_{pub} + [(a_1m_{i1} + a_2m_{i2} + \cdots + a_nm_{in}) + H_2(T)k_i]P$, and CC could not recover the specific power

consumption of each user due to the inaccessibility of k_i in the data reading stage. Therefore, P^2MDA is capable of resisting any internal attack.

- Man-in-the-middle attack. Above analyses suggest that in P^2MDA, GW could authenticate the SM_i through checking the equation $v_iP = L_i + H_3(ID_i, X_i, C_{1,i}, C_{2,i}, L_i, T)X_i$. Similarly, GW could be verified by CC via checking the equation $vP = R + H_3(ID_{GW}, X_{GW}, C_1, C_2, L_{GW}, T)X_{GW}$ holds. Therefore, P^2MDA is able to resist any man-in-the-middle attack.

Additionally, the security of P^2MDA in smart grid is compared with that of schemes [22], [24], [26], [28], [29], [33], [34], as shown in Table 2, where ✓ denotes "satisfy" and ✗ "not satisfy". S1, S2, S3, S4, S5, S6, S7, and S8 are used to represent confidentiality, authentication, integrity, modification attack, replay attack, impersonation attack, internal attack, and man-in-the-middle attack, respectively.

According to Table 2, P^2MDA could satisfy all security requirements that are described in Section III.

TABLE 2. Security comparisons.

Scheme	S1	S2	S3	S4	S5	S6	S7	S8
[22]	✓	✓	✓	✓	✓	✓	✗	✓
[24]	✓	✗	✗	✗	✓	✓	✗	✗
[26]	✓	✓	✓	✗	✓	✓	✗	✗
[28]	✓	✓	✓	✗	✓	✗	✓	✗
[29]	✓	✓	✗	✗	✓	✓	✓	✗
[33]	✓	✓	✓	✗	✓	✓	✗	✓
[34]	✓	✓	✓	✓	✓	✓	✓	✓
P^2MDA	✓	✓	✓	✓	✓	✓	✓	✓

VI. PERFORMANCE EVALUATION

In this section, the performance of P^2MDA is evaluated in comparison with schemes [22], [24], [26], [28], [29], [33], [34], from the perspectives of the computation cost of SM, GW and CC, and the communication cost between SM and GW, GW and CC, respectively.

For the sake of fairness, the multi-dimensional data aggregation schemes in smart grid are compared under the same security level of 80 bits. With regards the Paillier encryption-based schemes [22], [28], and [29], two prime numbers of 512 bits a, b are chosen. In terms of the pairing-based schemes [24] and [26], we choose a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is an additive group formed by a generator P with the order q on a super singular elliptic curve $E : y^2 = x^3 + x \bmod p$ with embedding degree 2. In addition, p, q are prime numbers of 512-bits and 160-bits respectively and satisfies the equation $q \cdot 12 \cdot r = p + 1$. As to the ECC-based scheme [33], we choose an elliptic curve $E : y^2 = x^3 + ax + b \bmod p$ with a prime order q, where p, q are 160 bits prime numbers and $a = -3$, b is a random 160 bits prime number.

A. COMPUTATION COST

The performance evaluations in terms of computation cost of P^2MDA and the related schemes [22], [24], [26], [28], [29], [33], and [34] are provided. Yet, some lightweight operations (hash function, hash chain, and point addition) are not taken into account. The well-known MIRACL Crypto SDK [45] is used to quantify the running time of the cryptographic operations. The evaluations are conducted using a laptop computer with 2.53GHz i5 CPU, 4 GB memory and 64-bit windows 10 operating system. The data in Table 3 was simulated for 10000 runs and their average is taken for consideration.

TABLE 3. Time cost of operation (millisecond).

Notations	Descriptions	Execution time
T_{m-ecc}	Scale multiplication operation in ECC	0.38
$T_{\log}$	Solving the DL operation mod p	0.64
T_{mtp}	Map to Point hash function operation	3.58
T_{n^2}	Exponentiation operation in $\mathbb{Z}_{n^2}$	2.02
T_p	Bilinear pairing operation	10.31
T_{p-D}	Paillier public key decryption operation	11.82
T_{p-E}	Paillier public key encryption operation	9.89
$T_{\exp-p}$	Exponentiation operation in p	0.13
T_m	Scale multiplication operation in bilinear pairing	1.42
T_n	Exponentiation operation in $\mathbb{Z}_n$	0.58

Based on the experiment results, the respective computation costs of [22], [24], [26], [28], [29], [33], and [34] and P^2MDA are summarized in Table 4.

Firstly, the computation costs of SM in different schemes are calculated. In scheme [22], SM requires $n+1$ exponentiation operations in $\mathbb{Z}_{n^2}$, one scale multiplication operation in bilinear pairing and one map-to-point hash operation. Hence the SM's computation cost is $1T_m + (n+1)T_{n^2} + 1T_{mtp} = 2.02n + 7.02$ ms. In schemes [24] and [26], SM requires three exponentiation operations in $\mathbb{G}_1$ and three scale multiplication operations in bilinear pairing. As a result, its computation cost is $3T_{\exp-p} + 3T_m = 4.65$ ms. In scheme [28], SM requires $4n+2$ scale multiplication operations in bilinear pairing, a Paillier public key encryption operation and $3n$ exponentiation operations in $\mathbb{Z}_n$. Accordingly, the SM's computation cost is $(4n+2)T_m + 1T_{p-E} + 3nT_n = 7.42n + 14.15$ ms. Moreover, in scheme [29], SM requires $3n$ multiplication operations and n exponentiation operations in $\mathbb{Z}_n$. Thus, the SM's computation cost is $3nT_m + nT_n = 4.84n$ ms. In scheme [33], SM requires $2n+2$ scale multiplication operations in ECC, so the computation cost is $(2n+2)T_{m-ecc} = 0.76n + 0.76$ ms. In scheme [34], SM requires $(n+8)$ exponentiation operations in $\mathbb{Z}_{n^2}$, hence the SM's computation cost is $(n+8)T_{n^2} = 2.02n + 16.16$ ms. Finally in P^2MDA, SM requires four scale multiplication operations in ECC. Consequently its computation cost is $4T_{m-ecc} = 1.52$ ms.

Secondly, the computation cost of GW in each scheme is obtained. In scheme [22], GW requires $w+1$ bilinear pairing operations, one scale multiplication operation in bilinear pairing and $w+1$ map-to-point hash operations. Hence the GW's computation cost is $(w+1)T_p + 1T_m + (w+1)T_{mtp} = 13.89w + 15.7$ ms. In schemes [24] and [26], GW requires $3w+1$ exponentiation operations in $\mathbb{G}_1$ and 3 scale multiplication operations in bilinear pairing. Thus, its computation cost is $(3w+1)T_{\exp-p} + 3T_m = 0.39w + 4.39$ ms.

TABLE 4. Computation cost (millisecond).

Scheme	SM	GW	CC	Total time
[22]	$1T_m$+$(n$+$1)T_{n^2}$+$1T_{mtp}$ = 2.02n+7.02 ms	$(w$+$1)T_p$+$1T_m$+$(w$+$1)T_{mtp}$ = 13.89w+ 15.70 ms	$2T_p$+$1T_{p-D}$+$1T_{mtp}$ = 36.04 ms	13.89w+2.02n+58.74 ms
[24]	$3T_{\exp-p}$+$3T_m$ = 4.65 ms	$(3w$+$1)T_{\exp-p}$+$3T_m$ = 0.39w+4.39 ms	$4T_{\exp-p}$+$3T_m$ = 4.78 ms	0.39w+13.82 ms
[26]	$3T_{\exp-p}$+$3T_m$ = 4.65 ms	$(3w$+$1)T_{\exp-p}$+$3T_m$ = 0.39w+4.39 ms	$4T_{\exp-p}$+$3T_m$ = 4.78 ms	0.39w+13.82 ms
[28]	$(4n$+$2)T_m$+$1T_{p-E}$+$3nT_n$ = 7.42n+14.15 ms	$(2w$+$1)T_m$+wT_n = 3.42w+1.42 ms	$1T_{p-D}$+$2T_m$ = 14.66 ms	7.42n+3.42w+28.81 ms
[29]	$3nT_m$+nT_n = 4.84n ms	$2wT_m$ = 2.84w ms	$2wT_m$+wT_n = 3.40w ms	6.24w+4.84n ms
[33]	$(2n$+$2)T_{m-ecc}$ = 0.76n+0.76 ms	$(w$+$2)T_{m-ecc}$ = 0.38w+0.76 ms	$(w$+$2)T_{m-ecc}$+$nT_{\log}$ = 0.38w+0.64n+0.76 ms	0.76w+1.40n+2.28 ms
[34]	$(n$+$8)T_{n^2}$ = 2.02n+16.16 ms	$(5w$+$5)T_{n^2}$ = 10.10w+10.10 ms	$(3w$+$7)T_{n^2}$ = 6.06w+14.14 ms	16.16w+2.02n+40.4 ms
P^2MDA	$4T_{m-ecc}$ = 1.52 ms	$(2w$+$2)T_{m-ecc}$ = 0.76w+0.76 ms	$4T_{m-ecc}$+$1T_{\log}$ = 2.16 ms	0.76w+4.42 ms

Furthermore, in scheme [28], GW requires $2w + 1$ scale multiplication operations in bilinear pairing and w exponentiation operations in $\mathbb{Z}_n$. Consequently, the computation cost is $(2w + 1)T_m + wT_n = 3.42w + 1.42$ ms. In scheme [29], GW requires $2w$ scale multiplication operations in bilinear pairing, and the corresponding computation cost is $2wT_m = 2.84w$ ms. In scheme [33], GW requires $w + 2$ scale multiplication operations in ECC. So its computation cost is $(w + 2)T_{m-ecc} = 0.38w + 0.76$ ms. In scheme [34], GW requires $(5w + 5)$ exponentiation operations in $\mathbb{Z}_{n^2}$, hence the GW's computation cost is $(5w+5)T_{n^2} = 10.10w+10.10$ ms. And in P^2MDA, GW requires $(2w+2)$ scale multiplication operations in ECC, thus with a computation cost as $(2w + 2)T_{m-ecc} = 0.76w + 0.76$ ms.

Thirdly, the computation cost of CC in each scheme is calculated successively. In scheme [22], CC requires two bilinear pairing operations, one Paillier public key decryption operation and one map-to-point hash operation. Hence, the CC's computation cost is $2T_p + 1T_{p-D} + 1T_{mtp} = 36.04$ ms. In scheme [24] and [26], CC requires four exponentiation operations in $\mathbb{G}_1$ and three scale multiplication operations in bilinear pairing, so the computation cost is $4T_{\exp-p} + 3T_m = 4.78$ ms. In scheme [28], CC requires a Paillier public key decryption operation and two scale multiplication operations in bilinear pairing. Hence the CC's computation cost is $1T_{p-D}+2T_m = 14.66$ms. In scheme [29], CC requires $2w$ scale multiplication operations in bilinear pairing and w exponentiation operations in $\mathbb{Z}_n$. Therefore, its computation cost is $2wT_m+wT_n = 3.40w$ ms. In scheme [33], CC requires $w + 2$ scale multiplication operations in ECC and n solving the DL operations mod p, so the computation cost is $(w + 2)T_{m-ecc}+nT_{\log} = 0.38w+0.64n+0.76$ ms. In scheme [34], CC requires $(3w+7)$ exponentiation operations in $\mathbb{Z}_{n^2}$, hence the CC's computation cost is $(3w + 7)T_{n^2} = 6.06w + 14.14$ ms. Lastly, in P^2MDA, CC requires four scale multiplication operations in ECC and one solving the DL operation mod p, thus with a computation cost as $4T_{m-ecc} + 1T_{\log} = 2.16$ ms.

Figure 3 and Figure 4 show the comparison results with regards computation cost between schemes [22], [24], [26], [28], [29], [33], [34] and P^2MDA.

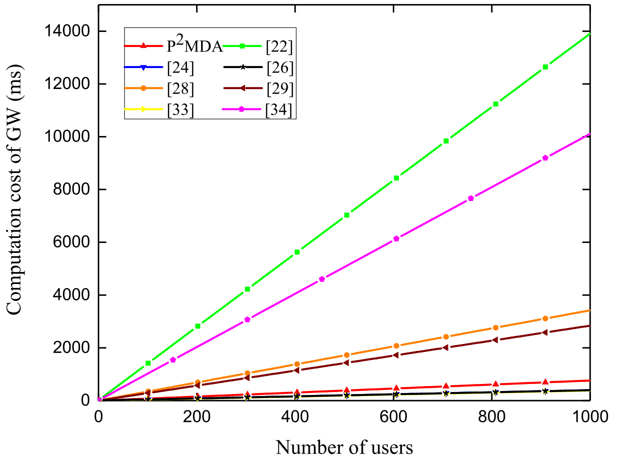


FIGURE 3. Computation costs of SM vs. Number of data types.

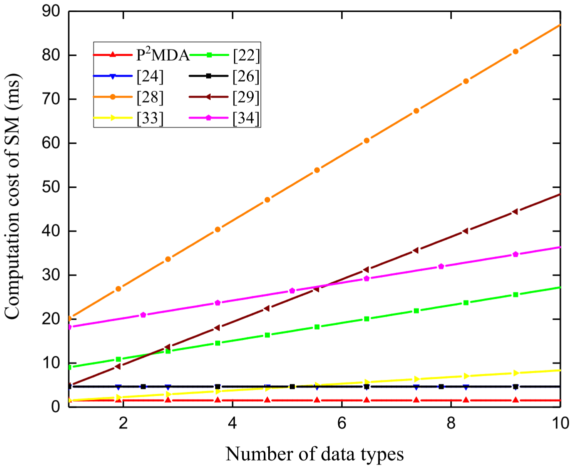


FIGURE 4. Computation costs of GW vs. Number of users.

Figure 3 illustrates the correlation between the computation cost of SM and the number of data types. Clearly, the computation cost of SM in our scheme is the smallest compared with [24] and [26]. Unlike that in schemes [22], [28], [29], [33] and [34], the computation cost

TABLE 5. Communication cost comparisons (bit).

Scheme	Communication cost SM-GW	Communication cost GW-CC
[22]	2659 bits	2659 bits
[24]	1024 bits	1024 bits
[26]	1184 bits	1184 bits
[28]	$2048n$ bits	$2048w$+$2048n$ bits
[29]	$2048n$ bits	$2048w$ bits
[33]	$160n$+544 bits	$160n$+544 bits
[34]	8448 bits	$2048w$+8448 bits
P^2MDA	704 bits	704 bits

of SM in our scheme remains the same with the increase in n. On the other hand, the correlation between the computation cost of GW and the number of users is demonstrated in Figure 4, which displays a lower slope of our computation cost curve in comparison to that of other schemes. Compare with scheme [33], although P^2MDA has a slightly larger computation cost of GW, its total computation cost is smaller.

We assume there are 1000 users and 10 types of data, as shown in Figure 5. Then, the total computation cost is further calculated in terms of number of users and data types. The computation cost of schemes [22], [24], [26], [28], [29], [33], [34] and P^2MDA are $13.89w+2.02n+57.45$ ms, $0.39w+13.82$ ms, $0.39w+13.82$ ms, $3.42w+17.42n+28.81$ ms, $6.24w+4.84n$ ms,$0.76w+1.40n+2.28$ ms, $16.16w+2.02n+40.4$ ms and $0.76w+4.42$ ms, respectively.

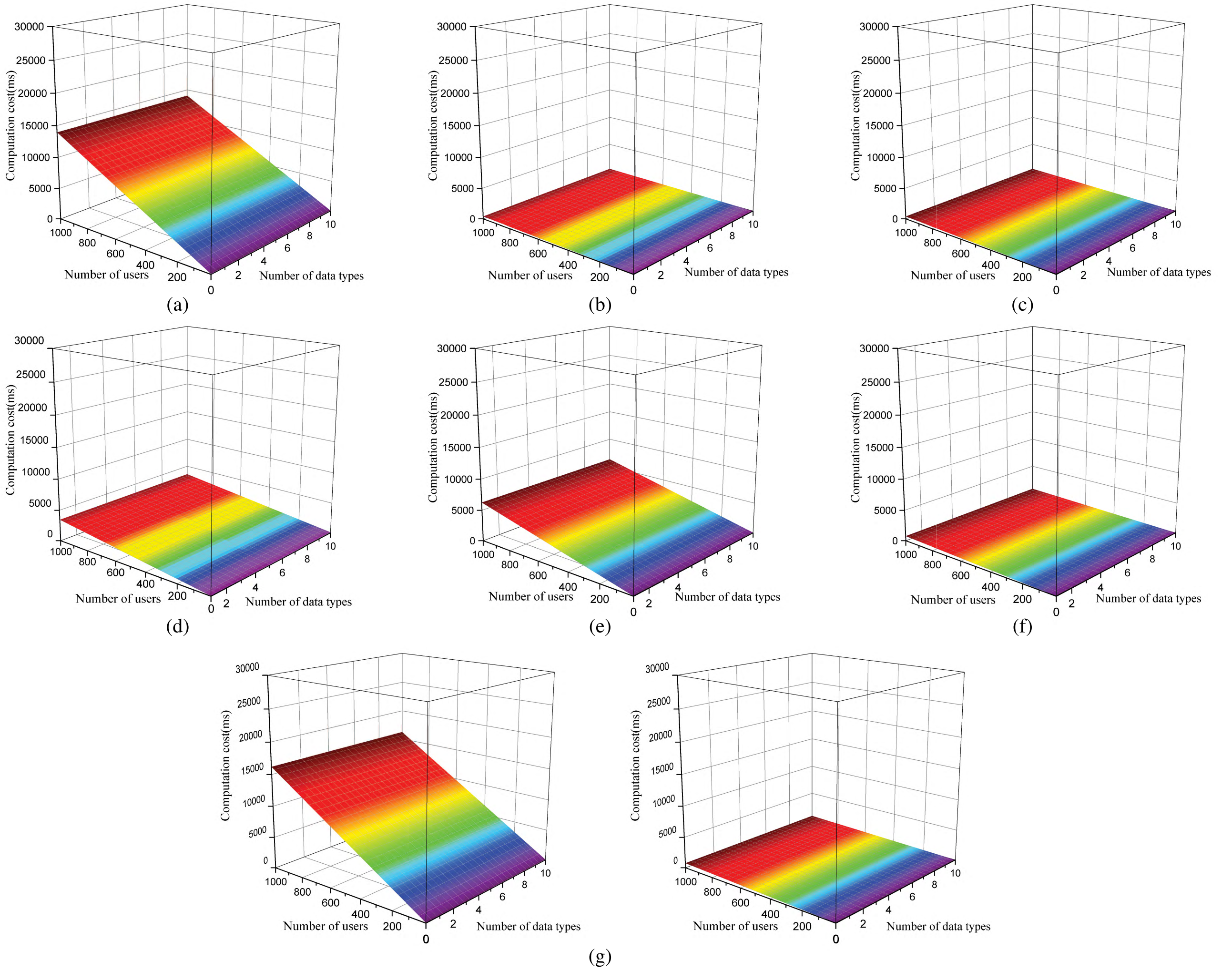
In summary, Figure 5 clearly shows that compared with schemes [22], [24], [26], [28], [29], [33], and [34] P^2MDA has the smallest total computation cost. Generally, the limited computing power and frequent data upload mode of SM in SG will bring about data delay and other failures if the computation cost is too high. Therefore, P^2MDA is undoubtedly more suitable for data aggregation in SG than other schemes.

B. COMMUNICATION COST

The performance evaluations in terms of communication cost of P^2MDA and schemes [22], [24], [26], [28], [29], [33], [34] are provided below. In smart grid, SM first transmits electricity data to GW which then aggregates and transmits these data to CC for analysing and processing. Therefore, the communication cost is generated as a result of the communication between SM and GW and between GW and CC. As is mentioned above, the length of elements in $\mathbb{G}_1$, $\mathbb{G}_T$, $\mathbb{G}$, $\mathbb{Z}_q^*$, $\mathbb{Z}_n$, $\mathbb{Z}_{n^2}$ respectively, is 512 bits, 1024bits, 160 bits, 160 bits, 1024bits and 2048bits. The length of one-way hash function is assumed as 160 bits, and that of identity and timestamp as 32 bits. The comparison of communication cost is shown in Table 5.

On the one hand, the communication between SM and GW is analyzed. In scheme [22], $\{C_i, \sigma_i, RA, U_i, TS\}$ is sent from SM to GW, where $C_i \in \mathbb{Z}_{n^2}$, $\sigma_i \in \mathbb{G}_1$, RA and U_i are 32-bit identities, TS is a 32-bit timestamp. Therefore, the communication cost is calculated as $|C_i + |\sigma_i| + |RA| + |U_i| + |TS| = 2048 + 512 + 32 + 32 + 32 = 2659$ bits. In scheme [24], C_i is sent from SM to GW, where $C_i \in \mathbb{G}_T$. Therefore, the communication cost is 1024 bits as $|C_i| = 1024$ bits. In scheme [26], $\{C_i, H_i\}$ is sent from SM to GW, where $C_i \in \mathbb{G}_T$, H_i being a one-way hash function. Hence, its communication cost is $|C_i| + |H_i| = 1024 + 160 = 1184$ bits. Furthermore, in scheme [28], $\{c_{i1}, c_{i2}, \cdots, c_{in}\}$ is sent from SM to GW, where $c_{ij} \in \mathbb{G}_T, j = 1, 2, \cdots, n$. As a result, the communication cost is $|c_{ij}| = 2048n$ bits. In scheme [29], $\{C_{i,1,r}, C_{i,2,r}, \cdots, C_{i,n,r}\}$ is sent from SM to GW, where $C_{i,j,r} \in Z_{n^2}, j = 1, 2, \cdot \cdot \cdot, n$. Thus, the communication cost is $|C_{i,j,r}| = 2048n$ bits. In scheme [33], $\{C_{ij}, ID_{ij}, S_{ij}, T\}$ is sent from SM to GW, where $C_{ij} \in \mathbb{G}$, $S_{ij} \in \mathbb{G}$. ID_{ij} is a 32-bit identity and T is a 32-bit timestamp. So the communication cost is calculated as $|C_{ij}| + |ID_{ij}| + |S_{ij}| + |T| = (n + 1) \times 160 + 32 + 160 \times 2 + 32 = 160n + 544$ bits. In scheme [34], $\{ID_g, ID_u, TS, R_i, \theta_i, T_i, B_i, MAC(B_i)\}$ is send from SM to GW, where ID_g and ID_u are 32-bit identities, TS is a 32-bit timestamp, $R_i \in \mathbb{Z}_{n^2}, \theta_i \in \mathbb{G}, T_i \in \mathbb{Z}_{n^2}, B_i \in \mathbb{Z}_{n^2}, MAC(B_i) \in \mathbb{Z}_{n^2}$. Therefore, the communication cost is $|ID_g|+|ID_u|+|TS|+|R_i|+|\theta_i|+|T_i|+|B_i|+|MAC(B_i)| = 32+32+32+2048+160+2048+2048+2048 = 8448$ bits. At last, in P^2MDA, $\{C_{1,i}, C_{2,i}, ID_i, L_i, v_i, T\}$ is sent from SM to GW, where $C_{1,i} \in \mathbb{G}$, $C_{2,i} \in \mathbb{G}$, $L_i \in \mathbb{G}$, $v_i \in \mathbb{Z}_q^*$. ID_i is a 32-bit identity and T a 32-bit timestamp. Therefore, the communication cost is $|C_{1,i}|+|C_{2,i}|+|ID_i|+|L_i|+|v_i|+|T| = 160 + 160 + 32 + 160 + 160 + 32 = 704$ bits.

On the other hand, the communication between GW and CC is analyzed. In scheme [22], $\{C, \sigma_g, RA, GW, TS\}$ is sent from GW to CC, where $C \in \mathbb{Z}_{n^2}$, $\sigma_g \in \mathbb{G}_1$, RA and GW are 32-bit identities TS is a 32-bit timestamp. As a result, the communication cost is calculated as $|C| + |\sigma_g| + |RA| + |GW| + |TS| = 2048 + 512 + 32 + 32 + 32 = 2659$ bits. In scheme [24], C is sent from GW to CC, where $C \in \mathbb{G}_T$. Therefore, the communication cost is 1024 bits as $|C| = 1024$ bits. In scheme [26], $\{C, H\}$ is sent from GW to CC, where $C \in \mathbb{G}_T$, H is a 160 bits one-way hash function. Hence, the communication cost is $|C| + |H| = 1024 + 160 = 1184$ bits. Moreover, in scheme [28], $\{R(i), C(j)\}$ is sent from GW to CC, where $R(i) \in \mathbb{Z}_{n^2}$, $i = 1, 2, \cdot \cdot \cdot, w$ and $C(j) \in \mathbb{Z}_{n^2}$, $j = 1, 2, \cdot \cdot \cdot, n$. Consequently, the communication cost is $2048w + 2048n$ bits as $|R(i)| + |C(j)| = 2048w + 2048n$ bits. In scheme [29], $C(i)$ is sent from GW to CC, where the communication cost is $2048w$ bits as $C(i) \in \mathbb{Z}_{n^2}$, $i = 1, 2, \cdot \cdot \cdot, w$. Therefore, $|C(i)| = 2048w$ bits. In scheme [33], $\{C_j, ID_j, T, S_j\}$ is sent from GW to CC, where $C_j \in \mathbb{G}$, $S_j \in \mathbb{G}$, ID_j is a 32-bit identity and T a 32-bit timestamp. So, the communication cost is $|C_j| + |ID_j| + |T| + |S_j| =(n+1)\times160+32+32+160\times2 = 160n+544$ bits. In scheme [34], $\{ID_u, ID_g, T(i), TS, R, \theta, B, MAC_1, MAC_2\}$ is sent from GW to CC, where ID_u and ID_g are 32-bit identities, TS is a 32-bit timestamp, $R \in \mathbb{Z}_{n^2}$, $\theta \in \mathbb{G}$, $T(i) \in \mathbb{Z}_{n^2}$, $i = 1, 2, \cdot \cdot \cdot, w$, $B \in \mathbb{Z}_{n^2}$, $MAC_1 \in \mathbb{Z}_{n^2}$, $MAC_2 \in \mathbb{Z}_{n^2}$. Therefore, the communication cost is bits as $|ID_u| + |ID_g| + |T(i)| + |TS| + |R| + |\theta|+|B|+|MAC_1|+|MAC_2| = 32+32+2048w+32+2048+160 + 2048 + 2048 + 2048 = 2048w + 8448$ bits. Finally, in P^2MDA, $\{C_1, C_2, ID_{GW}, L_{GW}, v_{GW}, T\}$ is sent from GW



(a) (b) (c) (d) (e) (f) (g)

FIGURE 5. **Computation cost. (a) Overall computation cost of scheme [22]. (b) Overall computation cost of scheme [24]. (c) Overall computation cost of scheme [26]. (d) Overall computation cost of scheme [28]. (e) Overall computation cost of scheme [29]. (f) Overall computation cost of scheme [33]. (g) Overall computation cost of scheme [34]. (h) Overall computation cost of scheme P^2MDA.**

to CC, where $C_1 \in \mathbb{G}$, $C_2 \in \mathbb{G}$, $L_{GW} \in \mathbb{G}$, $v_{GW} \in \mathbb{Z}_q^*$, ID_{GW} is a 32 bits identity and T a 32-bit timestamp. Therefore, the communication cost is $|C_1| + |C_2| + |ID_{GW}| + |L_{GW}| + |v_{GW}| + |T| = 160 + 160 + 32 + 160 + 160 + 32 = 704$ bits.

The cost generates by communication between SM and GW and between GW and CC is displayed in Figure 6 and Figure 7, respectively.

When the number of data types is assumed as 10 in Figure 6, P^2MDA saves 244.375 bytes, 40 bytes, 59.5 bytes, 2472 bytes, 2472 bytes, 180 bytes and 968 bytes of bandwidth successively during the communication between SM and GW, significantly decreases by 73.5%, 31.3%, 40.2%, 68.2%, 96.6%, 96.6%, 67.2% and 91.6% in comparison to schemes [22], [24], [26], [28], [29], [33] and [34].

Similarly, when both the number of users and data types are assumed as 10 in Figure 7, P^2MDA saves 244.375 bytes, 40 bytes, 59.5 bytes, 5032 bytes, 2472 bytes, 180 bytes and 3578 bytes of bandwidth respectively during the communication between GW and CC, significantly decreases by 73.5%, 31.3%, 40.2%, 68.2%, 98.3%, 96.6%, 67.2% and 97.6% compared with the schemes [22], [24], [26], [28], [29], [33], [34] respectively.

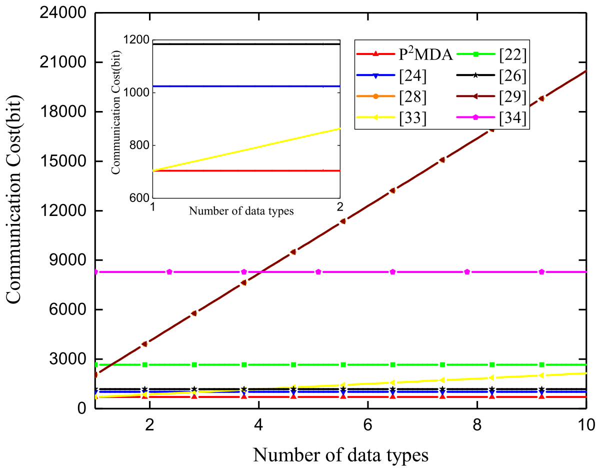


FIGURE 6. **Communication cost SM-GW.**

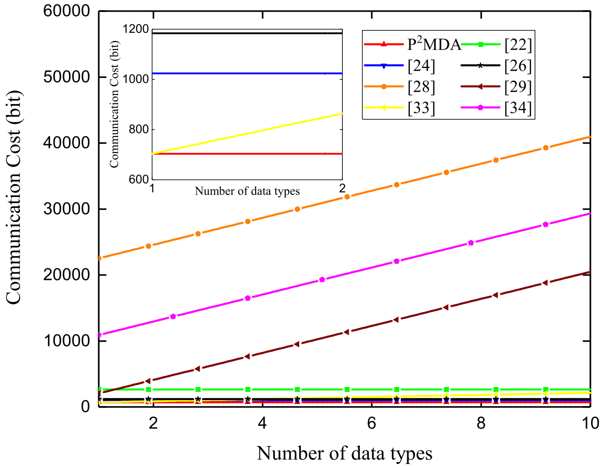


FIGURE 7. Communication cost GW-CC.

Judging from the above analyses, the costs of P^2MDA are considerably reduced during both communication phases. Compare with schemes [22], [24], [26] and [34], P^2MDA achieves the lowest communication cost. In addition, unlike that of schemes [28], [29], and [33], the communication cost of P^2MDA remains the same with the increase of n and w. Effectively reductions in communication cost, communication bandwidth, and latency are achieved by the proposed scheme. In summary, P^2MDA is capable of improving communication efficiency and saving communication bandwidth.

VII. CONCLUSION

In this paper, an efficient privacy-preserving multi-dimensional data aggregation (P^2MDA) scheme in SG has been proposed. It has achieved the multi-dimensional data aggregation based on homomorphic encryption and super-increasing sequence. The security analysis has indicated that the proposed scheme fulfills all security requirements. Moreover, the performance evaluations have demonstrated that P^2MDA is more efficient and low-cost in terms of computation and communication, for no bilinear pairing and map-to-point hash operations are used. Therefore, P^2MDA is more suitable for applications in smart grid.

REFERENCES

[1] A. Mao, G. Zhang, and Y. Lv, "Analysis on large-scale blackout occurred in South America and North Mexico interconnected power grid on Sept. 8, 2011 and lessons for electric power dispatching in China," *Power Syst. Technol.*, vol. 36, no. 4, pp. 74–78, Apr. 2012.

[2] E. Van der Vleuten and V. Lagendijk, "Transnational infrastructure vulnerability: The historical shaping of the 2006 European 'Blackout,'" *Energy Policy*, vol. 38, no. 4, pp. 2042–2052, Apr. 2010.

[3] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: Power delivery for the 21st century," *IEEE Power Energy Mag.*, vol. 3, no. 5, pp. 34–41, Sep./Oct. 2005.

[4] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 57–64, Jun. 2010.

[5] G. T. Heydt, "The next generation of power distribution systems," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 225–235, Dec. 2010.

[6] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 60–65, Apr. 2011.

[7] H. Liang, B. J. Choi, W. Zhuang, and X. Shen, "Towards optimal energy store-carry-and-deliver for PHEVs via V2G system," in *Proc. INFOCOM*, 2012, pp. 1674–1682.

[8] C. Greer, D. A. Wollman, D. E. Prochaska, and P. A. Boynton, "NIST framework and roadmap for smart grid interoperability standards, release 3.0," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Special Publication (NIST SP)-1108r3, Oct. 2014.

[9] J. Won, C. Y. Ma, D. K. Yau, and N. S. Rao, "Proactive fault-tolerant aggregation protocol for privacy-assured smart metering," in *Proc. INFOCOM*, 2014, pp. 2804–2812.

[10] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 82–88, Jun. 2010.

[11] W. Meng, R. Ma, and H.-H. Chen, "Smart grid neighborhood area networks: A survey," *IEEE Netw.*, vol. 28, no. 1, pp. 24–32, Jan./Feb. 2014.

[12] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 5–20, Feb. 2013.

[13] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 327–332.

[14] K. Alharbi and X. Lin, "LPDA: A lightweight privacy-preserving data aggregation scheme for smart grid," in *Proc. IEEE Int. Conf. Wireless Commun.*, Oct. 2012, pp. 1–6.

[15] X. Huang and S. Wang, "Aggregation points planning in smart grid communication system," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1315–1318, Aug. 2015.

[16] C. Li, R. Lu, H. Li, L. Chen, and J. Chen, "PDA: A privacy-preserving dual-functional aggregation scheme for smart grid communications," *Secur. Commun. Netw.*, vol. 8, no. 15, pp. 2494–2506, Oct. 2015.

[17] H. Bao and L. Chen, "A lightweight privacy-preserving scheme with data integrity for smart grid communications," *Concurrency Comput.; Pract. Exper.*, vol. 28, no. 4, pp. 1094–1110, Mar. 2016.

[18] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 248–258, Jun. 2015.

[19] H. Bao and R. Lu, "A lightweight data aggregation scheme achieving privacy preservation and data integrity with differential privacy and fault tolerance," *Peer Peer Netw. Appl.*, vol. 10, no. 1, pp. 106–121, Jan. 2017.

[20] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1767–1774, Mar. 2019. doi: 10.1109/TII.2018.2809672.

[21] X. Yi, R. Paulet, and E. Bertino, *Homomorphic Encryption and Applications*. Cham, Switzerland: Springer, 2014.

[22] 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 Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[23] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 598–607, Jun. 2014.

[24] R. Lu, K. Alharbi, X. Lin, and C. Huang, "A novel privacy-preserving set aggregation scheme for smart grid communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[25] Z. Sui, M. Niedermeier, and H. de Meer, "RESA: A robust and efficient secure aggregation scheme in smart grids," in *Proc. Int. Conf. Critical Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2015, pp. 171–182.

[26] M. Tahir, A. Khan, A. Hameed, M. K. Khan, and F. Jabeen, "Towards a set aggregation-based data integrity scheme for smart grids," *Ann. Telecommun.*, vol. 72, nos. 9–10, pp. 551–561, Oct. 2017.

[27] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1381–1396, Jun. 2017.

[28] B. Pan, P. Zeng, and K.-K. R. Choo, "A new multidimensional and fault-tolerable data aggregation scheme for privacy-preserving smart grid communications," in *Proc. Int. Conf. Appl. Techn. Cyber Secur. Intell.* Cham, Switzerland: Springer, 2018, pp. 206–219.

[29] B. Pan, P. Zeng, and K.-K. R. Choo, "An efficient data aggregation scheme in privacy-preserving smart grid communications with a high practicability," in *Proc. Int. Conf. Complex, Intell., Softw. Intensive Syst.* Cham, Switzerland: Springer, 2018, pp. 677–688.

[30] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multi-subset data aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 462–471, Feb. 2018.

[31] B. Lang, J. Wang, and Z. Cao, "Multidimensional data tight aggregation and fine-grained access control in smart grid," *J. Inf. Secur. Appl.*, vol. 40, pp. 156–165, Jun. 2018.

[32] S. Ge, P. Zeng, R. Lu, and K.-K. R. Choo, "FGDA: Fine-grained data analysis in privacy-preserving smart grid communications," *Peer Peer Netw. Appl.*, vol. 11, no. 5, pp. 966–978, Sep. 2018.

[33] O. R. M. Boudia, S. M. Senouci, and M. Feham, "Elliptic curve-based secure multidimensional aggregation for smart grid communications," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7750–7757, Dec. 2017.

[34] Y. Zhang, J. Li, D. Zheng, P. Li, and Y. Tian, "Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice," *J. Netw. Comput. Appl*, vol. 122, pp. 50–60, Nov. 2018.

[35] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 4, no. 31, pp. 469–472, Jul. 1985.

[36] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT* Berlin, Germany: Springer, 1999, pp. 223–238.

[37] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, 2004.

[38] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. Crypto*. Berlin, Germany: Springer, 2005, pp. 325–341.

[39] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Crypto*. Berlin, Germany: Springer, 1985, pp. 417–426.

[40] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 666–675, Feb. 2014.

[41] H. Bao and R. Lu, "Comment on 'privacy-enhanced data aggregation scheme against internal attackers in smart grid,"' *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 2–5, Feb. 2016.

[42] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2411–2419, Sep. 2017.

[43] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 2559–2564, Apr. 2014.

[44] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–369, Jun. 2000.

[45] Shamus Software. *Multi Precision Integer and Rational Arithmetic Cryptographic Library (MIRACL)*. Accessed: Dec. 1, 2018. [Online]. Available: http://www.certivox.com/miracl/

YANG MING (M'18) received the B.S. and M.S. degrees in mathematics from the Xi'an University of Technology, in 2002 and 2005, respectively, and the Ph.D. degree in cryptography from Xidian University, in 2008. He is currently a Professor with Chang'an University. His main research interests include cryptography and wireless network security.

XUANYI ZHANG received the B.S. degree from Xi'an Technological University, in 2016. He is currently pursuing the master's degree with Chang'an University, Xi'an. His main research interests include cryptography and homomorphic encryption.

XIAOQIN SHEN received the B.S. degree in mathematics from the Xi'an University of Technology, in 2002, and the Ph.D. degree in mathematics from Xi'an Jiaotong University, in 2007. She is currently a Professor with the Xi'an University of Technology. Her main research interests include cryptography and wireless network security.