Journal of Computer and Communications, 2025, 13(3), 156-175
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An Efficient and Anonymous Multidimensional Data Aggregation Scheme Based on Fog Computing for Smart Grids

Yu Chen, Jinmei Fan*, Yanhai Zhang

School of Mathematics and Statistics, Guilin University of Technology, Guilin, China
Email: unchey@163.com, *2007027@glut.edu.cn, zhang.yanhai@foxmail.com

How to cite this paper: Chen, Y., Fan, J.M. and Zhang, Y.H. (2025) An Efficient and Anonymous Multidimensional Data Aggregation Scheme Based on Fog Computing for Smart Grids. *Journal of Computer and Communications*, **13**, 156-175.
https://doi.org/10.4236/jcc.2025.133011

Received: February 26, 2025
Accepted: March 25, 2025
Published: March 28, 2025

Abstract

With the development of smart grid and the diversification of their applications, privacy-preserving multi-dimensional data aggregation has been widely studied because it can analyze users' electricity consumption more deeply. However, previous multidimensional data aggregation schemes suffer from heavy computational operations either to encrypt or decrypt the data. Additionally, existing fault-tolerant mechanisms require aggregation nodes to identify the IDs of smart meters, which leads to the risk of user privacy leakage. In this paper, we propose an efficient and anonymous multidimensional data aggregation scheme, called EAMA. In the proposed scheme, multidimensional data are encrypted by the Paillier encryption scheme of lower modular exponentiation, which reduces the computational cost. Moreover, to protect users' privacy, smart meters upload data using pseudonyms in a fog computing-based architecture, which prevents aggregation nodes from accessing smart meters' real IDs. Furthermore, a key challenge in the fault-tolerant phase is to eliminate blind factors, which EAMA effectively addresses without needing the real IDs of smart meters. Performance analysis demonstrates that EAMA achieves lower computational costs and satisfies security requirements.

Keywords

Data Aggregation, Privacy Protection, Fault Tolerance, Fog Computing, Smart Grid

1. Introduction

As an advanced power system, the smart grid (SG) makes use of contemporary information and communication technology [1] [2] to facilitate real-time data exchange and communication, increasing efficiency for better energy management.

By integrating traditional power grid infrastructures with advanced technologies such as cloud computing [3], fog computing [4] and mobile edge computing [5], the smart grid enables real-time data collection, transmission, and processing. This integration facilitates communication in both directions between companies and consumers, improving the efficiency, affordability, and stability of power management. A fundamental element of the SG is the smart meters (SMs) [6], which are positioned at the consumer's location to gather real-time electricity usage data and periodically report the data to the control center (CC). SMs are considered the most critical element on the consumer side, since they allow users to report their real-time energy usage actively. Based on the data, various applications are designed to enable the CC to predict power demand, adjust power generation, implement dynamic pricing strategies, and optimize overall grid management, etc. In order to give other organizations relevant data for commercial advertising or decision-making in the energy sector, CC could potentially carry out more statistical research [7]. Despite these advantages, the smart grid faces various security and privacy challenges. For example, by analyzing real-time power consumption data, malicious actors could potentially uncover users' habits and activities, leading to leak privacy. Besides, the aggregated data must be correct, even if certain SMs do not submit data of their own. Consequently, it is essential to adopt robust encryption technologies to safeguard user privacy.

Privacy-preserving data aggregation [8] has been extensively recognized as an effective approach for protecting data security in smart grids. To do this, SMs can employ homomorphic encryption (HE), enabling them to encrypt electrical usage data through Paillier Homomorphic Encryption [9]. This allows the aggregator gateway (AG) to safely aggregate the data that is encrypted while ensuring that the CC receives the total from SMs in a manner that preserves privacy [10]-[12]. Earlier aggregation methods primarily employed one-dimensional data aggregation, which was effective but lacked the fine-grained data required for a more detailed energy consumption analysis. Recent schemes have introduced multidimensional data aggregation techniques in smart grids, offering significant improvements over one-dimensional methods. Multidimensional aggregation enables the collection of electricity consumption data across various appliance categories, such as wall lamp, refrigerators, air conditioners, and dryers. The more fine-grained the data, the more effective the subsequent analysis becomes. However, multidimensional aggregation often involves complex cryptographic operations to ensure data encryption and decryption, adding to the computational complexity of the process. Traditional aggregation schemes rely on low-performance intermediate nodes that are often connected to multiple end nodes, limiting their ability to efficiently handle such cryptographic operations. To address this challenge, fog node (FN) [13] has been introduced in smart grids to enhance communication, computation, and storage capabilities, optimizing data aggregation. This transition to fog computing enhances the efficiency and scalability of data aggregation procedures.

As the data dimensions increase, the communication costs grow. A critical challenge is to minimize communication costs while maintaining user privacy. Previous aggregation schemes have employed masking technology [14] [15] to mask individual users' electricity consumption data, which reduces computational and communication overhead. However, these schemes fall short in terms of fault tolerance, as they require retransmission of billing reports in the event of node failures. Fault tolerance is essential for ensuring the robustness of data aggregation in smart grids. When SMs temporarily fail, preventing them from reporting data to FNs, fault tolerance mechanisms can help maintain data integrity. Some mechanisms have been suggested to achieve fault tolerance in response to the challenges presented by faulty SMs. User data is encrypted via a modified variant of the Paillier cryptosystem [16]. However, as the quantity of faulty SMs increases, the decryption expense raises substantially. Similarly, the framework suggested by Boudia *et al.* [17] incorporates a fault-tolerant method to guarantee the accuracy of the final aggregated data retrieved at the CC, even though the failure of certain nodes. This scheme employs a masking technique to protect individual user data, where the masks are designed to cancel each other out during aggregation. However, this approach requires the FN to know the real identities of the SMs, which introduces privacy concerns.

To solve the above problems, we propose EAMA, an efficiency and anonymous multidimensional data aggregation scheme based on fog computing for smart grids. This paper's primary achievements can be succinctly stated as below.

1) We have improved the Paillier cryptosystem by designing an encoding function that encodes multiple data items into a single ciphertext. Using this encoding function, the CC successfully obtains the aggregated data in the ciphertext. Additionally, the most computationally resource-intensive operation in Paillier encryption is the higher-order exponentiation of r^n, which we optimize to effectively reduce the computational overhead.

2) Since the data aggregator, the FN, is honest-and-curious, so we propose a bilinear pair-based batch anonymization authentication algorithm to efficiently verify users' anonymity and data integrity.

3) We offer a fault-tolerant approach for ensuring the accurate recovery of the final aggregated data at the CC. When an SM fails, our mechanism solves the problem of blind factor elimination without requiring the real ID of the faulty SM.

The subsequent sections of this article are structured as follows. Section II introduces related work. Section III introduces preliminary knowledge. Section IV introduces the system design of the EAMA, followed by Section V, which introduces the proposed scheme. Section VI introduces the system analysis of the EAMA, and Section VII discusses performance evaluation. Finally, Section VIII concludes the paper.

2. Related Work

In recent years, various data aggregation methods have been introduced for smart

grids. Among these, the work presented in [18] introduced a secure in-network aggregation approach specifically designed for smart grid environments. This approach leverages HE to ensure that users' private data remain undisclosed to intermediate aggregator nodes. As smart grids evolve, there is an increasing necessity for CC to obtain detailed data for enhanced services and to optimize demand response strategies. At the same time, it is crucial to preserve the privacy of individual users. HE emerged as a particularly advantageous approach for encrypted data aggregation owing to its homomorphic characteristics, which allow CC to execute statistical analyses effectively on encrypted data, thereby preserving the secrecy of user information.

In 2012, Lu *et al.* [19] introduced a multidimensional data aggregation framework utilizing the Paillier HE encryption method. Their technique employed a super-increasing sequence to facilitate CC's computation of the sum of diverse power usage data kinds. However, super-increasing sequences become less efficient for data packing when the dimensionality is high. To tackle this issue, Li *et al.* [20] introduced a multi-subset data aggregation strategy utilizing the Paillier homomorphic encryption algorithm, incorporating two super-increasing sequences. This design allowed the aggregation of data from different ranges, enabling CC to obtain more granular insights, such as the sum of consumption within specific regions and the consumer's count in each range. Despite these advancements, as noted in [21], the scheme restricts each subset of SMs to predefined data ranges, limiting the flexibility and utility of the aggregated data. Boudia *et al.* [22] proposed a secure multidimensional data aggregation scheme based on elliptic curves and utilizing ElGamal homomorphic encryption (HE) with multiple public keys. Their approach avoided complex encryption operations during data transmission, reducing both computational and communication overhead. However, the scheme required additional elliptic curve scalar multiplication operations for each dimension via the security module, which imposed a significant computational burden. In contrast, Zuo *et al.* [23] employed a super-incremental sequence combined with the ElGamal HE technique for multidimensional data aggregation. Their technique devised two categories of super-increasing sequences: one for computing the aggregate electrical usage of identical types among all users, and the other for verifying the total amount of individuals whose electrical usage resides inside specified intervals.

The data aggregation schemes proposed in [14] [15], and [24] employ masking techniques to conceal individual users' electricity usage data. By aggregating all masks, the values cancel each other out, effectively preserving user privacy while providing the overall power usage data. These schemes are notable for their low computation and communication overhead. However, they lack fault tolerance, necessitating additional billing reports to be transmitted in the event of system failures. Zhang *et al.* [25] established a privacy-preserving billing mechanism that integrates ElGamal's multidimensional data aggregation algorithm with an anonymous user identity design. This method efficiently safeguards against collusion

attacks perpetrated by any two individuals within the system. Additionally, the ciphertexts generated by ElGamal encryption are typically smaller than those produced by Paillier encryption, making ElGamal more efficient in terms of ciphertext storage for multidimensional data. Boudia *et al.* [17] employed the HE approach into a fog computing architecture to encode multidimensional data. For efficient authentication, their scheme employed a batch authentication technique. Although their method, known as ESMA, is fault tolerant, it requires blind factorial updates for all functioning SMs.

To address these limitations, we introduced encoding functionality into EAMA, utilizing the Paillier homomorphic encryption (HE) technique to efficiently construct and secure multidimensional data. EAMA employs a pseudonymization technique, enabling the CC to read and process aggregated data without requiring blind factor updates for all functioning SMs, even in cases where some SMs are non-functional. This ensures that the CC can continue processing aggregated reports, maintaining fault tolerance and operational efficiency.

3. Preliminaries

3.1. Bilinear Pairing Maps

A bilinear pairing map $\tilde{e}:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$ is defined on the elliptic curve E on the finite field F_p, where p is a large prime. In this context, $\mathbb{G}_1$ denotes an additive cyclic group of order q derived from the elliptic curve E, while $\mathbb{G}_2$ represents a multiplicative cyclic group of identical order q. The bilinear pairing map exhibits the following properties:

1) *Bilinearity*: For every $P,V\in\mathbb{G}_1$ and $a,b\in\mathbb{Z}_q^*$, it holds that $e(aP,bV)=e(P,V)^{ab}$.

2) *Non-degeneracy*: There are two elements $P,V\in\mathbb{G}_1$ such that $e(P,V)\neq 1$, where 1 denotes the identity element of $\mathbb{G}_2$.

3) *Computability*: For any $P,V\in\mathbb{G}$, there exists an efficient algorithm to compute $e(P,V)$.

3.2. Optimization of the Paillier Cryptosystem for Higher Order Polynomial Operations

The Paillier cryptosystem [9], after optimizing the power operation g^m in its original encryption process, encounters its most resource-intensive computation in the evaluation of the higher-order power function r^n. In 2010, Ivan Damgård [26] proposed an optimization strategy to streamline the calculation of r^n, demonstrating that this enhancement preserves the security level of the original Paillier algorithm. Therefore, the security of the modified Paillier encryption algorithm will be analyzed by imitating the original Paillier encryption algorithm.

1) *Key Generation*: To ensure security, it is required that $p\equiv q\equiv 3 \bmod 4$, and $\gcd(p-1,q-1)=2$.

Compute $\lambda=\dfrac{(p-1)(q-1)}{2}$. Select a random number x such that $x\in\mathbb{Z}_n^*$,

and calculate $h = -x^2 \bmod n$. Choose a natural number s. In the original Paillier scheme, this corresponds to setting $s = 1$. Compute $h_s = h^{n^s} \bmod n^{s+1}$. The optimized public key is defined as: $\text{PublicKey} = (n, h_s)$ and the private key is: $\text{PrivateKey} = (\lambda, \mu)$.

2) *Encryption*: Generate a random number α, where $\alpha \in \mathbb{Z}_2[k/2]$, and k is the key length. The optimized encryption formula is: $c = (n \times m + 1) \times h_s^{\alpha} \bmod n^{s+1}$. By choosing $\alpha \ll n$, the computational efficiency of h_s^{α} is significantly improved compared to r^n in the original encryption process.

3) *Decryption*: To decrypt the ciphertext c, the plaintext m is recovered using the equation: $m = D(c) = L\left(c^{\lambda} \bmod n^2\right)\mu \bmod n$, where $L(u) = \dfrac{u-1}{n}$ is the decryption function used in the Paillier cryptosystem.

4. System Design

4.1. System Model

Our proposed method emphasizes the secure aggregation of sensor data within a Fog computing-based Smart Grid system while ensuring the privacy of sensitive information. Our system model involves four participating entities: a group of SMs, $SM_j = \{SM_{1j}, SM_{2j}, \cdots, SM_{nj}\}$ located at the network edge; FNs situated near smart devices; a Remote CC; and a Key Generation Center (KGC), as depicted in Figure 1.

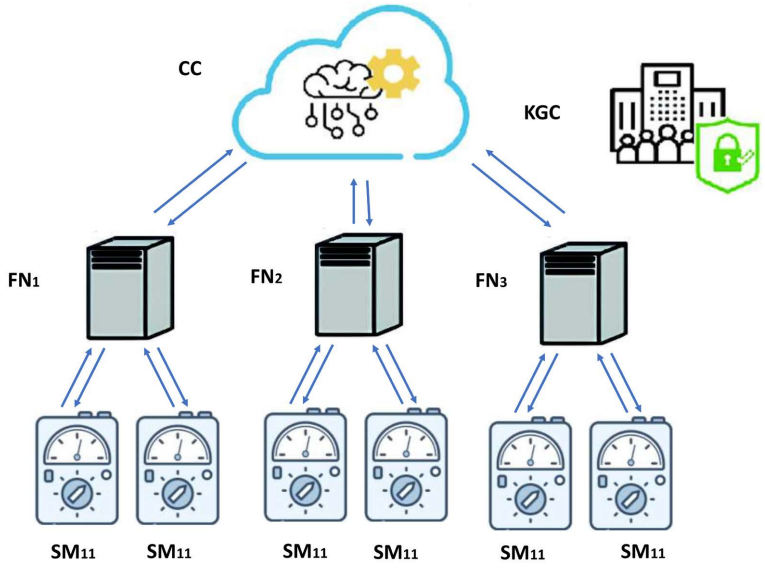

Figure 1. System model.

1) *SMs*: SMs, denoted as SM_{ij}, are terminal devices within the Internet of Things (IoT) network. The meters in question possess sensors, computational parts, to gather real-time data from their environment. The collected data is periodically reported to the nearest FN for further processing.

2) *FNs*: FNs are positioned nearby smart devices to enable communication be-

tween SM_{ij} and the CC. FN offers critical services, including data validation, the process, and storage for the CC. In our suggested method, the FN acquires data from SM_{ij} within its jurisdiction, authenticates and consolidates the data, and subsequently transmits the aggregated ciphertext to the CC for additional examination.

3) *CC*: The CC is tasked with producing system parameters during initialization and analyzing the aggregated data uploaded by the FN. The CC is crucial in overseeing and administering the system's overall functionality, utilizing the consolidated data obtained from the FNs.

4) *KGC*: The Key Generation Center (KGC) functions as a reliable intermediary in the cryptosystem, tasked with distributing public system parameters and private keys for SMs and CC, while also producing pseudonyms for each SM to safeguard user identity privacy.

4.2. Design Goals

Our objective is to introduce an effective and anonymous privacy-preserving multidimensional data aggregation scheme for SG based on Fog Computing. The objective is to safeguard critical information transmitted by smart terminal devices while maintaining user privacy. The subsequent objectives must be accomplished:

1) *Security*: The proposal must ensure the secrecy, integrity, and authenticity of data transition among system entities. Outside adversaries ($\mathbb{A}$) must be denied access to decrypted data. Any modifications to messages should be detectable, and the authentication of transmitting data, along with the verification of legal entity identities, should be performed by FNs and the CC.

2) *Privacy Protection*: User privacy is paramount in SG. Decryption keys are known only to the CC. Ciphertext computations are performed exclusively by the FN. KGC is aware of users' real identities, ensuring isolation between keys, ciphertext, and identity. This isolation guarantees that total power consumption can be obtained without revealing user privacy. Individual privacy in IoT applications is crucial. In this proposal, individual privacy is an unrevealed secret to $\mathbb{A}$. Even if there is collusion between the FN and CC, individual privacy is protected when user SM_{ij} employs anonymous privacy encryption.

3) *Fault Tolerance*: In Smart Grids, SM_{ij} may experience communication failures. The proposal must include fault tolerance mechanisms to ensure correct aggregation and decryption in the event of certain SM_{ij} failures.

4) *Performance*: Performance is crucial for both SMs and FNs to meet the practical demands of data aggregation in Smart Grids. This requires keeping computation and communication costs as low as possible to ensure system efficiency. Performance is essential for data aggregation in extensive SG, facilitating the prompt and efficient processing of significant data volumes.

5. Proposed Scheme

In this section, we present EAMA for Multidimensional Data Aggregation. **Table**

1 presents a compilation of acronyms and symbols utilized in this article, accompanied by their definitions. The scheme consists of five main components:

Table 1. Notations.

Notation	Definition
CC	The CC
FN	The FN
SM	The SM
n	The modulus defined as $n = p_0 q_0$
g	The generator of the group $\mathbb{Z}_{n^2}$
$\mathbb{G}_1$	An additive group
$\mathbb{G}_2$	A multiplicative group
(n, h_1)	The public key pair
(λ, μ)	The private key pair
e	A bilinear pairing
H	A secure hash function, $H: \{0,1\}^* \to \mathbb{G}_1$
H_1	A secure hash function $H_1: \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{G}_1$
w_j	The number of SM covered by FN_j
a_{ij}	The secret share of SM_{ij}
sk_{ij}	The secret key of SM_{ij}
pk_{ij}	The public key of SM_{ij}
$m_{ij,l}$	The data type l of SM_{ij}
$b_{ij,l}$	The encoded representation of $m_{ij,l}$
l	The total count of data categories
2^z	The upper limit for a data category's value

5.1. System Initialization

Based on the security parameter κ, KGC is required to generate two κ-bit prime numbers, p_0 and q_0, which satisfy the conditions: $p_0 \equiv q_0 \equiv 3 \pmod 4$, $\gcd(p_0 - 1, q_0 - 1) = 2$. The KGC subsequently chooses two large κ-bit primes, p_0 and q_0, and calculates the public key of the Paillier encryption system as $n = p_0 q_0$, $g = 1 + n$. The associated private key is expressed as. Subsequently, the $\lambda = \frac{(p_0 - 1)(q_0 - 1)}{2}$ KGC constructs a function $L(u) = \frac{u - 1}{n}$ and computes μ as $\mu = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n$ A random number x is then chosen, where $x \in \mathbb{Z}_n^*$, and $h = -x^2 \bmod n$. A natural number s is chosen, and for the original Paillier system, $s = 1$. For $h_s = h^{n^s} \bmod n^2$, we have $h_1 = h^n \bmod n^2$. The public key is denoted as the tuple (n, h_1), whereas the associated private key is represented by the tuple (λ, μ). The KGC defines a bilinear pairing map $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative cyclic groups of

identical order p, and P serves as the generator of $\mathbb{G}_1$. The KGC further establishes four collision-resistant hash functions: $H:\{0,1\}^* \to \mathbb{G}_1$ and $H_1:\{0,1\}^* \times \mathbb{G}_1 \to \mathbb{G}_1$.

The KGC establishes the upper limit of FNs at J, and defines $SM_{ij} \subseteq \{1j, 2j, \cdots, Nj\}$ as the indexed collection of operational SMs. A pseudo-random number producer generates $N \times J$ secret shares $a_{ij} \in \mathbb{Z}_n^*$, where $i = 1, \cdots, N$, and $j = 1, \cdots, J$. The KGC then computes a_{0j}:

$$a_{0j} = \sum_{ij \in SM_{ij}} a_{ij} \bmod n, \text{ where } j = 1, \cdots, J \tag{1}$$

5.2. Registration

1) SM_{ij} *Registration and Pseudonym Selection*: Each SM SM_{ij} produces its unique identity ID_{ij}, randomly chooses a private key $sk_{ij} \in \mathbb{Z}_q^*$, and derives its public key as $pk_{ij} = sk_{ij} \cdot P$, with P representing the group's generator. The pseudonym PID_{ij} is computed as: $PID_{ij} = H_1\left(RID_{ij} \| sk_{ij}\right)$ The signature $\bar{\sigma}_{ij}$ is then calculated as: $\bar{\sigma}_{ij} = H_1\left(PID_{ij} \| pk_{ij} \| TS\right) \cdot sk_{ij}$ The SM SM_{ij} sends $\{RID_{ij}, PID_{ij}, pk_{ij}\}$ to KGC along with the timestamp TS.

Upon receiving the registration request, the KGC first verifies the signature:

$$\bar{\sigma}_{ij} \cdot P = H_1\left(PID_{ij} \| pk_{ij} \| TS\right) \cdot pk_{ij}$$

If the signature is valid, the KGC generates a list of pseudonyms L_{ij} based on the real identity RID_{ij}. The pseudonym list corresponding to PID_{ij} of SM_{ij} is as follows: $L_{ij} = \{PID_{ij1}, PID_{ij2}, \cdots, PID_{ijl}, \cdots\}$.

During the l-th anonymous update cycle, the KGC selects a random number $z_l \in \mathbb{Z}_q^*$ (where $l = 1, 2, \cdots, N$) to modify the pseudo-identity of SM_{ij}. The updated pseudonym is computed as: $PID_{ijl} = PID_{ij} \oplus H_1\left(PID_{ij} \| z_l \cdot P\right)$ The new signature σ_{ijl} is calculated as:

$$\sigma_{ijl} = H1\left(PID_{ijl} \| pk_{ij} \| TS\right) \cdot sk_{ij} \tag{2}$$

The KGC verifies the following equation to ensure consistency:

$$H1\left(PID_{ij} \| pk_{ij} \| TS\right) \cdot pk_{ij} = \sigma_{ijl} \cdot P \tag{3}$$

If the formula is satisfied, the tuple $\{PID_{ijl}, pk_{ij}, \sigma_{ijl}\}$ is recorded. After successful verification and registration, the KGC sends $\{PID_{ijl}, pk_{ij}, \sigma_{ijl}\}$ to the FN and CC for further processing.

2) *FN Registration*: FN is registered, randomly selected $sk_j \in Z_q^*$ as the private key, and computed the public key $pk_j = sk_j P$, where $\{ID_j, pk_j\}$ sent to KGC.

3) *CC Registration*: CC randomly selects $sk_{CC} \in Z_q^*$ as the private key and computes the public key $pk_{CC} = sk_{CC} P$. Then, CC sends $\{ID_{CC}, pk_{CC}\}$ to KGC.

At the end of this phase, KGC publishes the system parameters as

$$\Gamma = \left(e, \mathbb{G}_1, \mathbb{G}_2, p, P, n, g, h_1, z, H, H_1, pk_{CC}, \{pk_j\}_{1 \le j \le J}, \{pk_{ij}\}_{1 \le i \le N, 1 \le j \le J}\right)$$

5.3. Report Generation

During the time period T, each SM_{ij} encrypts the l-dimensional data $\left(m_{ij,1}, m_{ij,2}, \cdots, m_{ij,l}\right)$ and simultaneously creates the corresponding signature in Figure 2.

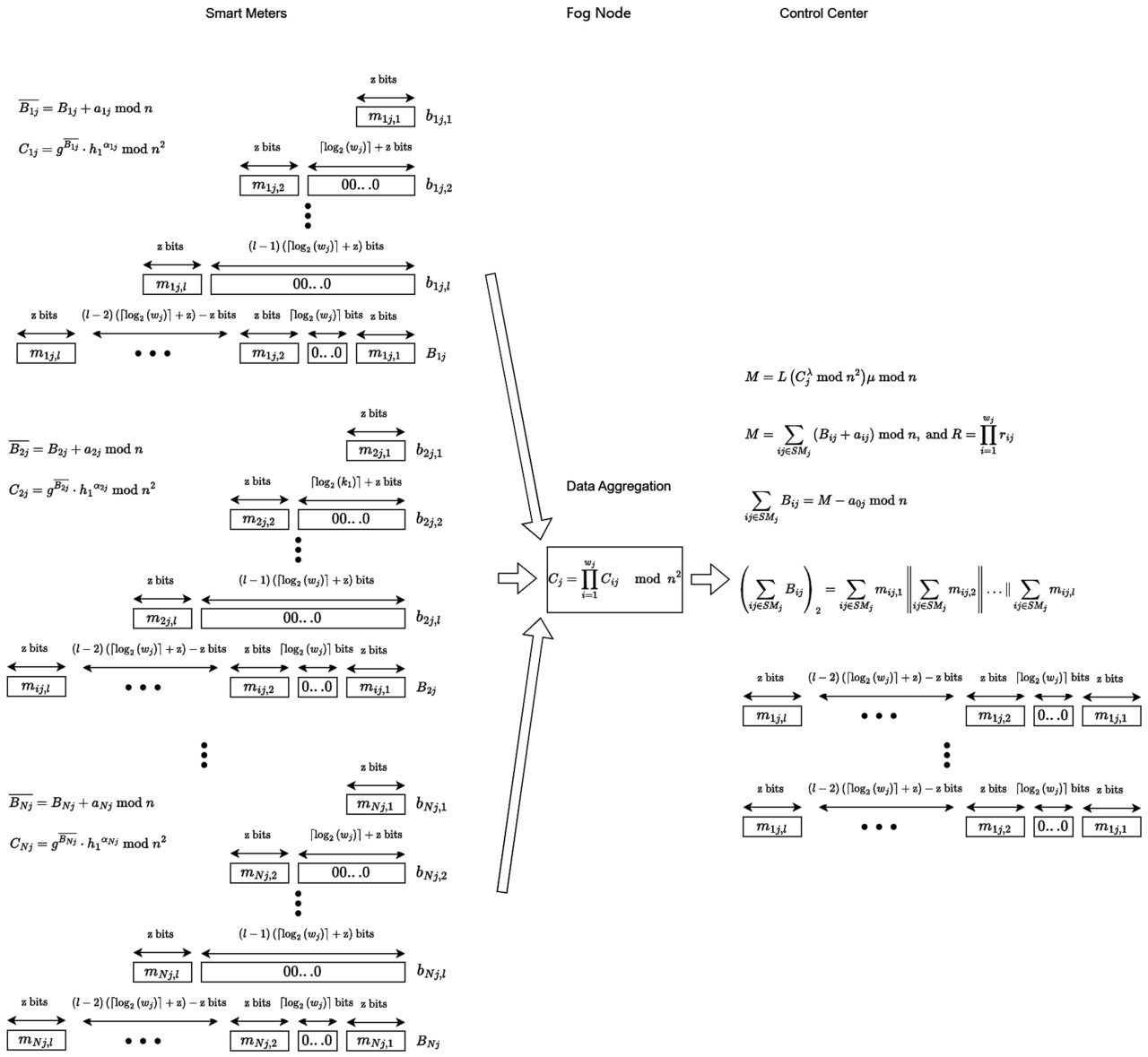

Figure 2. Data operation flowchart.

Step 1: SM_{ij} encodes $\left(m_{ij,1}, m_{ij,2}, \cdots, m_{ij,l}\right)$ into the binary string $\left(b_{ij,1}, b_{ij,2}, \cdots, b_{ij,l}\right)$ where each of the $b_{ij,d} = \left(m_{ij,d}\right)_2 \| 0^{z^*(d-1)}, d = 1, 2, \cdots, l$ and assigns its private information as $m_{ij} = \sum_{d=1}^{l} b_{ij,d}$. $B_{ij} = b_{ij,1} + b_{ij,2} + \cdots + b_{ij,l}$. Then encode, SM_{ij} to compute $\overline{B_{ij}}$.

$$\overline{B_{ij}} = B_{ij} + a_{ij} \bmod n \tag{4}$$

Step 2: In the preceding Paillier encryption method, SM_{ij} selects a random

number $r_{ij} \in \mathbb{Z}_n^*$ and calculates the encrypted content as

$$C_{ij} = g^{\overline{B_{ij}}} \cdot r_{ij}^n \bmod n^2 \tag{5}$$

In our Paillier cryptosystem, randomly generate $\alpha_{ij}, \alpha_{ij} \in Z_2\left[k/2\right]$, where k is the key length

$$\begin{aligned} C_{ij} &= g^{\overline{B_{ij}}} \cdot g^{\alpha_{ij}} \bmod n^2 \\ &= (n+1)^{\overline{B_{ij}}} \cdot h_1^{\alpha_{ij}} \bmod n^2 \\ &= \left(n * \overline{B_{ij}} + 1\right) h_1^{\alpha_{ij}} \bmod n^2 \\ &= \left(n * \overline{B_{ij}} + 1\right) h_1^{\alpha_{ij}} \bmod n^2 \\ &= \left(n * \left(B_{ij} + a_{ij} \bmod n\right) + 1\right) h_1^{\alpha_{ij}} \bmod n^2 \end{aligned} \tag{6}$$

Step 3: SM_{ij} Utilize the private key sk_{ij} to generate the signature as described below. $\sigma_{ij} = sk_{ij} H\left(C_{ij} \| TS \| ID_j \| PID_{ijl}\right)$ where TS means the present timestamp.

Step 4: SM_{ij} transmits the verified data report information $\left(C_{ij}, TS, \sigma_{ij}, PID_{ijl}\right)$ to the corresponding F_j.

5.4. Data Aggregation

1) *All-Inclusive Data Aggregation*: During the period T, we posit the occurrence of certain SM failures within the SG, and use $SM_j \subseteq \left\{PID_{1jl}, PID_{2jl}, \cdots, PID_{Njl}\right\}$ to denote the index set of working SMs. Let $w_j = \left|SM_j\right|$ describe the quantity of operational SMs under F_j, and let t signify the minimum threshold of the effective sample size. When $w_j \geq t$, verified data report messages (e.g., $\left(C_{ij}, TS, \sigma_{ij}, PID_{ijl}\right)$) are received from legitimate SMs, F_j conducts a batch validation by verifying the subsequent calculation:

$$\left(e\left(\sum_{ij \in SM_j} \sigma_{ij}, P\right) = \prod_{ij \in SM_j} e\left(H\left(C_{ij} \| TS \| ID_j \| PID_{ijl}\right), pk_{ij}\right)\right.$$

The FN_j will initially conduct bulk verification to authenticate the received signature by determining if the subsequent equation is satisfied:

$$\begin{aligned} e\left(\sum_{ij \in SM_j} \sigma_{ij}, P\right) &= \prod_{ij \in SM_j} e\left(H\left(C_{ij} \| TS \| ID_j \| PID_{ijl}\right), pk_{ij}\right) \\ &= \prod_{ij \in SM_j} e\left(H\left(C_{ij} \| TS \| ID_j \| PID_{ijl}\right), sk_{ij} P\right) \\ &= \prod_{ij \in SM_j} e\left(sk_{ij} H\left(C_{ij} \| TS \| ID_j \| PID_{ijl}\right), P\right) \\ &= \prod_{ij \in SM_j} e\left(\sigma_{ij}, P\right) \end{aligned} \tag{7}$$

$$e\left(P, \sum_{i=1}^{w_j} \sigma_{ij}\right) = \prod_{i=1}^{w_j} e\left(p_{k_{ij}}, H\left(C_{ij} \| ID_{ij} \| TS\right)\right). \tag{8}$$

Batch verification decreases the quantity of pairing processes from w_j to $w_j + 1$. Upon validation, FN_j aggregates all encrypted data and transmits the aggregated data to CC. Execute the subsequent steps.

Step 1: FN_j aggregates w_j encrypted ciphertexts into

$$\begin{aligned} C_j &= \prod_{i=1}^{w_j} C_{ij} \bmod n^2 \\ &= \prod_{i=1}^{w_j} g^{\overline{B_{ij}}} \cdot h_1^{\alpha_{ij}} \bmod n^2 \\ &= g^{\sum_{i=1}^{w_j} \overline{B_i}} \cdot \left(\prod_{i=1}^{w_j} h_1^{\alpha_{ij}} \right) \bmod n^2 \\ &= g^{\sum_{i=1}^{w_j} \left(B_{ij} + a_{ij} \right) \bmod n} \cdot \left(h_1^{\sum_{i=1}^{w_j} \alpha_{ij}} \right) \bmod n^2 \\ &= \left(1 + n \cdot \left(\sum_{i=1}^{w_j} \left(B_{ij} + a_{ij} \right) \bmod n \right) \right) \cdot \left(h_1^{\sum_{i=1}^{w_j} \alpha_{ij}} \right) \bmod n^2 \end{aligned} \tag{9}$$

Step 2: The function FN_j generates the signature utilizing its private key sk_j in the following manner.

$$\sigma_j = sk_j H\left(C_j \| TS \| ID_j \right) \tag{10}$$

where TS denotes the recent timestamp.

Step 3: FN_j sends a full report to CC containing

$$\left\{ C_j, TS, \sigma_j, ID_j \right\} \tag{11}$$

If some SMs break down, FN_j will not receive the corresponding packets.

2) *Fault-Tolerant Data Aggregation*: When certain SMs are unable to transmitting data, FN_j will not obtain the relevant packets. Let $SM_j \subseteq \left\{ PID_{1jl}, PID_{2jl}, \cdots, PID_{Njl} \right\}$ denote the collection of all operational SM units in FN_j, and let SM_j' represent the subset of defective SM devices ($SM_j' \in SM_j$). Consequently, we get

$$a_{0j} \neq \sum_{ij \in SM_j / SM_j'} a_{ij} \bmod n. \tag{12}$$

Consequently, this phenomenon will directly influence the accuracy of the ultimate decryption result. The entity FN_j must transmit the set SM_j' to the KGC. Subsequently, the KGC generates a new a_{0j}' based on SM_j' and transmits it to the CC.

5.5. Data Reading

1) All-Inclusive Data reading: Upon obtaining a whole report from FN_j, CC initially authenticates the signature in accordance with the subsequent equation:

$$e\left(P, \sigma_j \right) = e\left(pk_j, H\left(C_j \| TS \| ID_j \right) \right) \tag{13}$$

When the equation is satisfied, it indicates that the signature is legitimate. Upon verifying the authenticity, CC decrypts the aggregated ciphertext C_j and extracts the aggregated data by executing the subsequent phases:

Step 1: CC decrypts the aggregated ciphertext by first retrieving the encrypted data C_j. From Equation (8), take:

$$M = \sum_{ij \in SM_j} \left(B_{ij} + a_{ij} \right) \bmod n \tag{14}$$

The report $\left(1 + M \cdot n\right) \cdot \left(h_1\right)^{\sum_{i=1}^{w_j} \alpha_{ij}} \bmod n^2$ is still the ciphertext of the Paillier encryption system. It can be equated to $g^M \cdot R^n \bmod n^2$ with $R = \prod_{i=1}^{w_j} r_{ij}$ and CC uses the tuple $\left(\lambda, \mu\right)$ to recover M : as

$$M = L\left(C_j^{\lambda} \bmod n^2\right) \mu \bmod n \tag{15}$$

Step 2: After decryption, CC uses a_{0j} to obtain $\sum_{ij \in SM_j} B_{ij}$ as follows:

$$\sum_{ij \in SM_j} B_{ij} = M - a_{0j} \bmod n \tag{16}$$

Step 3: CC Retrieve each aggregated data $\sum_{ij \in SM_j} b_{ij,d}$, $d = 1, 2, \cdots, l$ using the decoding function.

The CC divides the binary representation of $\sum_{ij \in SM_j} B_{ij}$ into l bit chunks of maximal length less than z , so that the aggregated data $M_{ij} = \sum_{ij \in SM_j, d=1}^{l} m_{ij,d}$ can be written as

$$\left(\sum_{ij \in SM_j} B_{ij} \right)_2 = \sum_{ij \in SM_j} m_{ij,l} \| \cdots \| \sum_{ij \in SM_j} m_{ij,2} \| \sum_{ij \in SM_j} m_{ij,1} \tag{17}$$

2) *Fault-Tolerant Data Reading*: Upon receipt of the fault-tolerant report from FN_j , CC initially authenticates the signature in accordance with the subsequent equation:

$$e\left(P, \sigma_j\right) = e\left(pk_j, H\left(C_j \| TS \| ID_j\right)\right) \tag{13}$$

However, the released M ($M = \sum_{ij \in SM_j'} \left(B_{ij} + a_{ij} \right) \bmod n$) contains only a subset of the a_{ij} values, and thus we cannot use a_{0j} to eliminate the blind factor. Consequently, we utilize the a_{0j}' generated by the KGC to obtain $\sum_{ij \in SM_j'} B_{ij}$.

$$\sum_{ij \in SM_j'} B_{ij} = M - a_{0j} \bmod n \tag{18}$$

CC Retrieve each aggregated data $\sum_{ij \in SM_j'} b_{ij,d}$, $d = 1, 2, \cdots, l$ using the decoding function.

The CC divides the binary representation of $\sum_{ij \in SM_j'} B_{ij}$ into l bit chunks of maximal length less than z , so that the aggregated data $M_{ij} = \sum_{ij \in SM_j', d=1}^{l} m_{ij,d}$ can be written as

$$\left(\sum_{ij \in SM_j'} B_{ij} \right)_2 = \sum_{ij \in SM_j'} m_{ij,l} \| \cdots \| \sum_{ij \in SM_j'} m_{ij,2} \| \sum_{ij \in SM_j'} m_{ij,1} \tag{19}$$

Thus, we can convert the aggregated data into a binary bit string, and then separate out the substrings of length z bits starting from the lower bit to retrieve the aggregated data for each dimension.

6. System Analysis

In EAMA, we presume that FNs and CC are honest and curious. Each FN precisely compiles encrypted multidimensional electrical usage data from the identical grid

zone, while the CC effectively checks and decrypts the data that is provided. Both entities are interested in users' power consumption trends and may seek to extract useful information. Furthermore, an internal attacker (e.g., a curious user) may attempt to exploit the secret key of the SM and other essential factors to get the original data from SMs located in other users' residences.

1) *Anonymity*: During data transmission, the SM communicates with the FN via a pseudonym. During the information transmission session, SMs establish communication links with the FN utilizing temporary identity. As the parameter z_l employs a randomization technique in the identity generation process, there exists no discernible association between several temporary identifications produced by the same device. Only KGC has the capability to restore the user's true identity by analyzing the temporary IDs. To augment security and resist attackers from correlating high-precision electricity consumption data with user identities, the system periodically updates and manages the temporary identifiers of all SMs. This technique efficiently safeguards user identification information, hence achieving the anonymity objective of the scheme design.

2) *Confidentiality*: An outside attacker is presumed to possess the capability to eavesdrop on the public communication channel between the SM and the FN, thereby intercepting the encrypted message C_{ij}. Nonetheless, due to the semantic security afforded by the Paillier encryption method, an adversary is unable to extract any meaningful information from the ciphertext, even if it is acquired. During the encrypted data reporting phase, each SM_{ij} first encodes the l-dimensional data $\left(m_{ij,1}, m_{ij,2}, \cdots, m_{ij,l}\right)$ into a binary string $\left(b_{ij,1}, b_{ij,2}, \cdots, b_{ij,l}\right)$, where each $b_{ij,d} = \left(m_{ij,d}\right)_2 \| 0^{z*(d-1)}$, for $d = 1, 2, \cdots, l$. The private data is then set as $m_{ij} = \sum_{d=1}^{l} b_{ij,d}$. Under the modified Paillier encryption scheme, the corresponding ciphertext is generated as: $C_{ij} = \left(n \cdot \overline{B_{ij}} + 1\right) \cdot h_1^{\alpha_{ij}} \bmod n^2$ where $\overline{B_{ij}} = B_{ij} + a_{ij} \bmod n$. Each SM_{ij} holds a unique secret parameter a_{ij}, which varies for each instance. Consequently, EAMA is resistant to selective plaintext attacks.

3) *Privacy*: In this framework, FN retains only the encrypted data of anonymous users without possessing the decryption key. CC possesses the decryption key but cannot access the fine encrypted information or the users' true identities; and KGC, while aware of the users' true identities, cannot acquire their detailed data. Even if FN and CC collude, CC can decrypt the encrypted data of individual users acquired from FN. Nonetheless, it is challenging to erase the secret parameter a_{ij}, preventing the identification of the actual users behind the electricity consumption data. Consequently, despite the potential for cooperation between two parties, the method can nevertheless proficiently obscure real-time electricity usage data from being linked to the user's true identity, so safeguarding personal privacy.

4) *Leak Toughness*: EAMA ensures that a group of collaborating users does not threaten the privacy of other users. When an attacker $\mathbb{A}$ seeks to compromise a user's privacy, they must have access to personal data and the associated secret share x_{ij}. In EAMA, the secret shares $a_{ij} \in \mathbb{Z}_n^*$ produced by the KGC are allo-

cated independently, indicating that the compromise of secret shares from a subset of users does not disclose those of other users. Assume that $\mathbb{A}$ successfully compromises $w_j - 1$ users associated with a certain FN_j and acquires their secret shares $a_{1j}, a_{2j}, \cdots, a_{(w_j-1)j}$. For these $w_j - 1$ users, Equation (1) can be reformulated as:

$$a_{0j} = \sum_{i=1}^{w_j-1} a_{ij} + a_{(w_j-1)j} \bmod n \tag{20}$$

Despite this, the data privacy of the remaining users remains intact, as $\mathbb{A}$ does not possess the CC's secret share of the sum a_{0j} nor the Paillier private key. Therefore, $\mathbb{A}$ can't compromise $a_{w_j j}$. Consequently, with EAMA, $\mathbb{A}$ cannot access the private data of other users, irrespective of the number of hacked users. Even if the Paillier private key were to be obtained through logarithmic computation, $\mathbb{A}$ would still be unable to retrieve individual data contents due to the secure embedding of the secret shared value a_{ij} within each ciphertext. Thus, in EAMA, encrypting power consumption data not only ensures leakage resilience but also preserves data confidentiality and the privacy of all SMs.

7. Performance Evaluation

Utilizing established cryptographic libraries, MIRACL [27] and PBC [28], we evaluate the expense of the cryptographic operations employed in EAMA and prior technologies. Our technique is predicated on the modified Paillier cryptosystem utilizing a 1024-bit key and a pairing operation within a base field of 160 bits. We utilize the standard curve secp160r1 for ECC.

Table 2 presents the findings acquired on a computer equipped with an Intel i7-8750H CPU operating at 2.2 GHz and 8 GB of RAM. It is important to recognize that multiplication in Z_n^2, modular addition, and hash operations are insignificant in comparison to exponentiation in Z_n^2 and pairing operations.

Table 2. Time consumption for related operations.

Symbol	Definition	Time (ms)
T_h	Time for general hash function	0.01
T_H	Time for hashing to ECC point multiplication	0.02
T_{ex}	Time for modular exponentiation	7.13
T_{mun}	Time for regular multiplication	1.58
T_{pmn}	Time for ECC point multiplication	0.32
T_{pa}	Time for ECC point addition	0.34
T_{inv}	Time for modular inversion	8.72
T_{bp}	Time for bilinear pairing operation	16.81

7.1. Computation Cost

In EAMA, when SM_{ij} produces its output, it necessitates $T_{mu} + T_{ex}$ to generate C_{ij} and $T_H + T_{pm}$ to produce σ_{ij}. Consequently, the whole computational ex-

pense on the SM side is represented by $T_{mu}+T_{ex}+T_H+T_{pm}$. Our approach considerably decreases the computing expense for the user. (l represents multidimensional data.)

In data aggregation, upon receiving $\{C_{ij},\sigma_{ij},TS,PID_{ijl}\},i=1,2,\cdots,N$ from all the SMs, FN_j performs $n+1$ bilinear pairwise operations and one ECC hash-to-dot for data verification. Subsequently, it consolidates the encrypted data and produces a signature, requiring $3n-3$ conventional multiplication operations to form the aggregated ciphertext, one ECC hash-to-point operation, and one ECC pointwise multiply operation to create a new signature on the aggregated the encrypted data. The overall computational expense of FN is expressed as $(n+1)\cdot T_{bp}+(3n-3)\cdot T_{mu}+2T_H+T_{pm}$. Within the Fog computing paradigm, Fog Nodes has greater processing capabilities than traditional nodes. Consequently, the aggregate procedure may be executed efficiently. Subsequently, we evaluate the computational expense of the current schemes with regard to each SM and AG individually. In the approach proposed by Zhang *et al.* [25], offer a data reporting method where in the computation is expressed as $\omega_1 m_{i1}+\omega_{k+1}m_{i1}^2+\omega_2 m_{i2}+\omega_{k+2}m_{i2}^2+\cdots+\omega_k m_{ik}+\omega_{2k}m_{ik}^2$, each SM requires $3k$ multiplication operations, in addition to two exponentiation operations, one hash operation, and one multiplication to generate CT_i. Generating the authenticator σ_i requires two exponentiation operations, two hash operations, and one multiplication. Thus, the total computation costs on the SM side total $4T_{ex}+3T_h+(3k+2)\cdot T_{mu}$. In Data Aggregation, after AG receives $\{CT_i,\sigma_i,T\},i=1,2,\cdots,n$, from all SMs, it requires $n-1$ multiplication operations to generate the aggregated ciphertext CT. It requires one exponentiation, one hash operation to compute τ_n, n multiplication operations to generate ξ, and n exponentiation operations coupled with $2n-1$ multiplication operations to get σ. The total computational costs of AG are represented as $(n+1)\cdot T_{ex}+(4n-2)\cdot T_{mu}+T_h$. Using a comparable analytical method, Zuo *et al.*'s framework [23] indicates that the overall computational expense for each SM component is $k\cdot T_{mu}+T_h+5T_{ex}$, while the total computational expense for the AG component is $(n+1)\cdot T_{bp}+(4n-4)\cdot T_{mu}+(n+1)\cdot T_h+T_{ex}$. Ultimately, the overall computational expense for each SM and aggregator in Boudia *et al.*'s framework [17] is $2T_{ex}+1T_H+1T_{pm}$ and $(n+1)\cdot T_{bp}+(3n-3)\cdot T_{mu}+2T_H+1T_{pm}$, respectively.

A comparison of the computational overhead in terms of SMs is presented in **Figure 3**.

7.2. Communication Cost

Given that SMs are equipped with resource-constrained storage and computational devices, each SM encrypts l categories of data and transmits them to the respective aggregator. The communication cost can be categorized into two components: communication from the SM to the FN and communication from the FN to the CC. Given that SMs are equipped with resource-constrained storage and computational equipment, we concentrate on assessing the communication

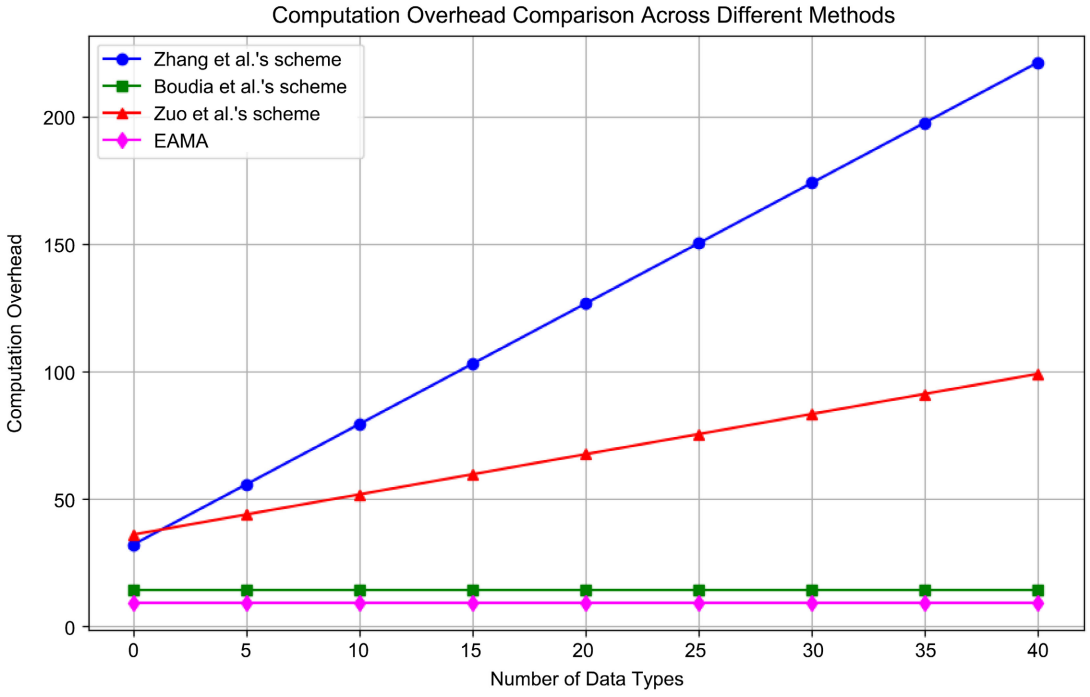

Figure 3. Computational overhead on the SM side.

expenses from SMs to FN. In EAMA, the variables, $\{C_{ij}, TS, \sigma_{ij}, PID_{ijl}\}$ are sent from the SM to FN_j, where $C_{ij} \in \mathbb{Z}_{n^2}$ and $\sigma_{ij} \in \mathbb{G}_1$. To provide uniformity, we designate the size of the PID_{ijl} and timestamp as 160 bits and 64 bits, respectively, across all methods. Consequently, the transmission expense from SM to FN amounts to $2048+64+160+160=2432$ bits. Subsequently, we examine the correspondence from FN to CC. In EAMA, the elements, $\{C_j, TS, \sigma_j, ID_j\}$ are transmitted from FN_j to CC, where $C_{ij} \in \mathbb{Z}_{n^2}$ and $\sigma_{ij} \in \mathbb{G}_1$. The communication cost from FN to CC is calculated as $2048+64+160+64=2336$ bits. In the system proposed by Zhang *et al.* [25], each SM_i transmits $\{CT_i, \sigma_i, T\}$ to AG. Consequently, the transmission expense from n SMs to AG amounts to $(2048+512+64)\cdot n = 2624n$ bits. In the approach proposed by Zuo *et al.* [23], each SM_i transmits $\{ID_i, C_i^a, C_i^b, T_i, \sigma_i, pk_i\}$ to AG. The transmission expense from n SMs to AG is $(64+512+512+64+1024+512)\cdot n = 2688n$ bits. In the framework suggested by Boudia *et al.* [17], $\{C_{ij}, ID_{ij}, TS, \sigma_{ij}\}$ is transmitted from SM_{ij} to the respective FN, where C_{ij} represents the ciphertext of the Paillier encryption system and σ_{ij} denotes a signature, resulting in a total communication cost of $(2048+64+64+160)\cdot n = 2336n$ bits. Ultimately, the comparison of communication costs is illustrated in **Figure 4**.

To evaluate our enhanced scheme against current ones, we established $n=400$ as the real value and performed a comprehensive comparison of the communication costs from n SMs to the FN, as illustrated in **Figure 4**. The figure indicates that the communication cost of our design is inferior to that of other schemes, with the exception of the scheme proposed by Boudia *et al.* [17]. While the transmission cost from SM to FN in the strategy proposed by Boudia *et al.* is somewhat

cheaper than ours, their scheme lacks pseudonyms, potentially compromising its security.

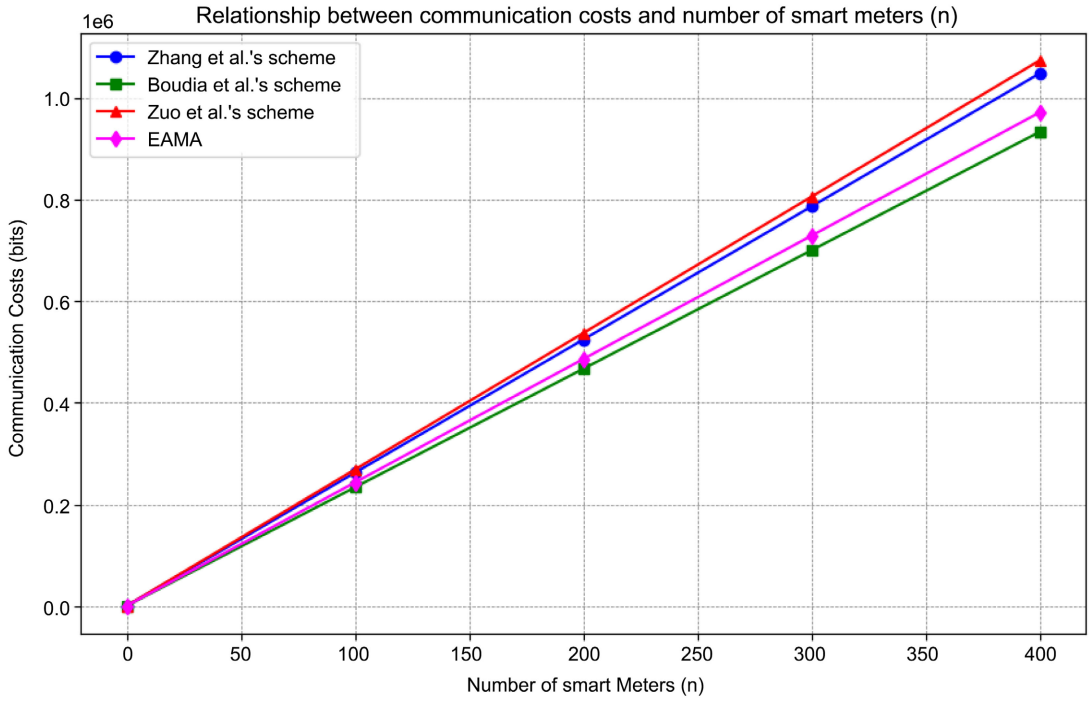

Figure 4. Communication overhead on the SM side.

8. Conclusion

In this paper, we propose an efficient and anonymized multidimensional data aggregation scheme for fog computing smart grids, named EAMA. By leveraging an improved Paillier encryption scheme, our approach enhances encryption performance. Furthermore, the scheme incorporates a pseudonym mechanism for SMs, enabling FNs to aggregate data based on pseudonyms before retrieving the final aggregated results. The security analysis concludes that the approach guarantees data privacy, confidentiality, integrity, and authentication. The performance analysis underscores the scalability benefits of EAMA, the efficacy of its fault-tolerance mechanism, and its cost-efficiency in both computing and communication. Furthermore, EAMA supports inquiries beyond mere summing, rendering it appropriate for the application needs of smart city smart grids.

Funding

The work was supported by the National Natural Science Foundation of China under Grant 12061027, by the Natural Science Foundation of Guangxi of China under Grant 2018GXNSFBA281019, by the Doctoral Research Foundation of Guilin University of Technology under Grant GUTQDJJ2018033, and by the Opening Fund of Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education under Grant CRKL210206.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhang, Z., Deng, R., Yau, D.K.Y., Cheng, P. and Chen, J. (2020) Analysis of Moving Target Defense against False Data Injection Attacks on Power Grid. *IEEE Transactions on Information Forensics and Security*, **15**, 2320-2335. https://doi.org/10.1109/tifs.2019.2928624

[2] Chouikhi, S., Esseghir, M. and Merghem-Boulahia, L. (2023) Energy Consumption Scheduling as a Fog Computing Service in Smart Grid. *IEEE Transactions on Services Computing*, **16**, 1144-1157. https://doi.org/10.1109/tsc.2022.3174698

[3] Yigit, M., Gungor, V.C. and Baktir, S. (2014) Cloud Computing for Smart Grid Applications. *Computer Networks*, **70**, 312-329. https://doi.org/10.1016/j.comnet.2014.06.007

[4] Liu, J., Weng, J., Yang, A., Chen, Y. and Lin, X. (2020) Enabling Efficient and Privacy-Preserving Aggregation Communication and Function Query for Fog Computing-Based Smart Grid. *IEEE Transactions on Smart Grid*, **11**, 247-257. https://doi.org/10.1109/tsg.2019.2920836

[5] Ni, J., Zhang, K. and Vasilakos, A.V. (2021) Security and Privacy for Mobile Edge Caching: Challenges and Solutions. *IEEE Wireless Communications*, **28**, 77-83. https://doi.org/10.1109/mwc.001.2000329

[6] Kabalci, Y. (2016) A Survey on Smart Metering and Smart Grid Communication. *Renewable and Sustainable Energy Reviews*, **57**, 302-318. https://doi.org/10.1016/j.rser.2015.12.114

[7] Rossi, B. and Chren, S. (2020) Smart Grids Data Analysis: A Systematic Mapping Study. *IEEE Transactions on Industrial Informatics*, **16**, 3619-3639. https://doi.org/10.1109/tii.2019.2954098

[8] Gong, Y., Cai, Y., Guo, Y. and Fang, Y. (2016) A Privacy-Preserving Scheme for Incentive-Based Demand Response in the Smart Grid. *IEEE Transactions on Smart Grid*, **7**, 1304-1313. https://doi.org/10.1109/tsg.2015.2412091

[9] Paillier, P. (1999) Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J., Ed., *Advances in Cryptology—EUROCRYPT'99*. *Lecture Notes in Computer Science*, Springer, 223-238. https://doi.org/10.1007/3-540-48910-x_16

[10] Kataray, T., Nitesh, B., Yarram, B., Sinha, S., Cuce, E., Shaik, S., *et al.* (2023) Integration of Smart Grid with Renewable Energy Sources: Opportunities and Challenges—A Comprehensive Review. *Sustainable Energy Technologies and Assessments*, **58**, Article 103363. https://doi.org/10.1016/j.seta.2023.103363

[11] Zhang, K., Lu, L., Zhao, J., Wei, L. and Ning, J. (2024) Secure Multi-Asks/bids with Verifiable Equality Retrieval for Double Auction in Smart Grid. *Peer-to-Peer Networking and Applications*, **17**, 3255-3268. https://doi.org/10.1007/s12083-024-01744-5

[12] Ayub Khan, A., Ali Laghari, A., Rashid, M., Li, H., Rehman Javed, A. and Reddy Gadekallu, T. (2023) Artificial Intelligence and Blockchain Technology for Secure Smart Grid and Power Distribution Automation: A State-of-the-Art Review. *Sustainable Energy Technologies and Assessments*, **57**, Article 103282. https://doi.org/10.1016/j.seta.2023.103282

[13] Lyu, L., Nandakumar, K., Rubinstein, B., Jin, J., Bedo, J. and Palaniswami, M. (2018) PPFA: Privacy Preserving Fog-Enabled Aggregation in Smart Grid. *IEEE Transac-*

tions on Industrial Informatics, **14**, 3733-3744.
https://doi.org/10.1109/tii.2018.2803782

[14] Gope, P. and Sikdar, B. (2020) An Efficient Privacy-Friendly Hop-by-Hop Data Aggregation Scheme for Smart Grids. *IEEE Systems Journal*, **14**, 343-352. https://doi.org/10.1109/jsyst.2019.2899986

[15] Gope, P. and Sikdar, B. (2019) Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart Grids. *IEEE Transactions on Information Forensics and Security*, **14**, 1554-1566. https://doi.org/10.1109/tifs.2018.2881730

[16] Saleem, A., Khan, A., Malik, S.U.R., Pervaiz, H., Malik, H., Alam, M., *et al.* (2020) FESDA: Fog-Enabled Secure Data Aggregation in Smart Grid IoT Network. *IEEE Internet of Things Journal*, **7**, 6132-6142. https://doi.org/10.1109/jiot.2019.2957314

[17] Merad-Boudia, O.R. and Senouci, S.M. (2021) An Efficient and Secure Multidimensional Data Aggregation for Fog-Computing-Based Smart Grid. *IEEE Internet of Things Journal*, **8**, 6143-6153. https://doi.org/10.1109/jiot.2020.3040982

[18] Li, F., Luo, B. and Liu, P. (2010). Secure Information Aggregation for Smart Grids Using Homomorphic Encryption. 2010 1*st IEEE International Conference on Smart Grid Communications*, Gaithersburg, 4-6 October 2010, 327-332. https://doi.org/10.1109/smartgrid.2010.5622064

[19] Lu, R.X., Liang, X.H., Li, X., Lin, X.D. and Shen, X.M. (2012) EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications. *IEEE Transactions on Parallel and Distributed Systems*, **23**, 1621-1631. https://doi.org/10.1109/tpds.2012.86

[20] Li, S., Xue, K., Yang, Q. and Hong, P. (2018) PPMA: Privacy-Preserving Multisubset Data Aggregation in Smart Grid. *IEEE Transactions on Industrial Informatics*, **14**, 462-471. https://doi.org/10.1109/tii.2017.2721542

[21] Mustafa, M.A., Cleemput, S., Aly, A. and Abidin, A. (2019) A Secure and Privacy-Preserving Protocol for Smart Metering Operational Data Collection. *IEEE Transactions on Smart Grid*, **10**, 6481-6490. https://doi.org/10.1109/tsg.2019.2906016

[22] Merad Boudia, O.R., Senouci, S.M. and Feham, M. (2017) Elliptic Curve-Based Secure Multidimensional Aggregation for Smart Grid Communications. *IEEE Sensors Journal*, **17**, 7750-7757. https://doi.org/10.1109/jsen.2017.2720458

[23] Zuo, X., Li, L., Peng, H., Luo, S. and Yang, Y. (2021) Privacy-Preserving Multidimensional Data Aggregation Scheme without Trusted Authority in Smart Grid. *IEEE Systems Journal*, **15**, 395-406. https://doi.org/10.1109/jsyst.2020.2994363

[24] Alsharif, A., Nabil, M., Sherif, A., Mahmoud, M. and Song, M. (2019) MDMS: Efficient and Privacy-Preserving Multidimension and Multisubset Data Collection for AMI Networks. *IEEE Internet of Things Journal*, **6**, 10363-10374. https://doi.org/10.1109/jiot.2019.2938776

[25] Zhang, S., Chang, J. and Wang, B. (2023) A Multidimensional Data Aggregation Scheme of Smart Home in Microgrid with Fault Tolerance and Billing for Demand Response. *IEEE Systems Journal*, **17**, 4639-4649. https://doi.org/10.1109/jsyst.2023.3286468

[26] Damgård, I., Jurik, M. and Nielsen, J.B. (2010) A Generalization of Paillier's Public-Key System with Applications to Electronic Voting. *International Journal of Information Security*, **9**, 371-385. https://doi.org/10.1007/s10207-010-0119-9

[27] Shamus (2014) Multiprecision Integer and Rational Arithmetic C/C++ Library (Miracl).

[28] Lynn PBC Library. https://crypto.stanford.edu/pbc/