Hindawi
Security and Communication Networks
Volume 2022, Article ID 7359134, 12 pages
https://doi.org/10.1155/2022/7359134

*Research Article*

# EPPSA: Efficient Privacy-Preserving Statistical Aggregation Scheme for Edge Computing-Enhanced Wireless Sensor Networks

**Yunting Tao**,[1] **Fanyu Kong**,[1] **Jia Yu**,[2] **and Qiuliang Xu**[1]

[1]*School of Software, Shandong University, Jinan 250101, China*
[2]*College of Computer Science and Technology, Qingdao University, Qingdao 266071, China*

Correspondence should be addressed to Fanyu Kong; fanyukong@sdu.edu.cn

Received 29 November 2021; Accepted 1 April 2022; Published 2 May 2022

Academic Editor: Yuyu Yin



In edge computing-enhanced wireless sensor networks (WSNs), multidimensional data aggregation can optimize the utilization of computation resources for data collection. How to improve the efficiency of data aggregation has gained considerable attention in both academic and industrial fields. This article proposes a new efficient privacy-preserving statistical aggregation scheme (EPPSA) for WSNs, in which statistical data can be calculated without exposing the total number of sensor devices to control center. The EPPSA scheme supports multiple statistical aggregation functions, including arithmetic mean, quadratic mean, weighted mean, and variance. Furthermore, the EPPSA scheme adopts the modified Montgomery exponentiation algorithms to improve the aggregation efficiency in the edge aggregator. The performance evaluation shows that the EPPSA scheme gets higher aggregation efficiency and lower communication load than the existing statistical aggregation schemes.

## 1. Introduction

In recent years, wireless sensor networks (WSNs) have achieved an accelerated increase in deployment. WSNs are widely utilized in scenarios such as smart homes [1], vehicular ad hoc networks [2–4], industrial Internet of Things [5], and monitoring environments [6–8]. The sensor devices in WSNs are responsible for sensing real-time data and transmitting the sensed data to control center for data analysis and intelligent control. In a variety of WSN applications, some computations are too time-consuming for sensor devices. Edge computation is an effective solution for resource-limited sensor devices to gain edge devices' assistance, such as data aggregation and neural network models [9]. With the edge computation devices deployed near the target area, the computing load in WSN sensor devices could be distributed to the edge devices. With the help of edge computation devices, cloud data centers provide various services for numbers of applications [10–13].

To reduce data redundancy and communication delay, data aggregation has become one of the most practical techniques, which can be used in edge computing-enhanced WSNs. Usually, a gateway is an ideal edge device to perform data aggregation operations due to its high computational capability, and mobile edge computing (MEC) also provides an emergent paradigm that brings computation close to mobile sensors [14]. It is worth noting that data aggregation at edge gateways may suffer from some potential security risks [15]. Firstly, the data may be captured or falsified during the delivery process, considering WSNs are usually deployed in an unattended environment. Secondly, adversaries can invade the edge gateway for stealing users' private data. The traditional security approaches cannot be directly applied to edge computing-enhanced WSN data aggregation, since they may be conflicted with aggregation function [16]. Furthermore, due to the dynamic and heterogeneous characteristics of WSN devices, there exits difficulty for the sensed data to be collected, encrypted, used, and stored in accordance with the users' preferences [17, 18].

To solve the above problems, homomorphic encryption algorithms have been considered to construct privacy-preserving single-dimensional aggregation schemes [19–21]. Furthermore, researchers proposed several multidimensional privacy-preserving data aggregation schemes, the core

idea of which is to construct a conversion mechanism between multidimensional data and large integers [19, 20, 22–33]. These researches are centered on how to reduce computation costs and communication load while collecting and transmitting the data. Lu et al. [26] proposed an efficient privacy-preserving data aggregation (EPPA) scheme in smart grids. Merging multidimensional data by super-increasing sequence of large primes, Lu et al.'s scheme is more efficient than the one-dimensional data aggregation schemes. Using a polynomial method, Shen et al. [27] constructed a user-level polynomial to store multidimensional values in a single data space based on Horner's rule. Fault tolerance can be used to enhance the security and robustness of a data aggregation scheme. In [32], Mohammadali et al. presented a homomorphic privacy-preserving data aggregation scheme with the fault tolerance property, so it can keep data secure even if the aggregator is malicious or curious.

Most secure data aggregation schemes only consider summation-based aggregation since the underlying additive homomorphic encryption only supports the modular addition operations. In practice, various types of statistics (e.g., mean, variance and standard deviation) might often need to be supported for data application [34]. Therefore, it is necessary to design multifunctional secure data aggregation scheme supporting various data statistics. Zhang et al. [35] proposed a multifunctional secure data aggregation scheme (MODA). This scheme offers the building blocks for multifunctional aggregation by encoding raw data into well-defined vectors. Peng et al. [36] introduced a multifunctional aggregation scheme supporting diversified aggregation functions, including linear, polynomial, and continuous functions. Both of the above schemes implement the statistical functions computed by control center. For example, in [36], the ciphertext sum is generated in the edge device and the mean is calculated using the decrypted sum by control center. Thus, the total number of sensor devices is required to transmit to control center for calculating the mean by sum/total number.

In lots of WSN application scenarios (e.g., industrial monitoring), the total number of sensor devices represents industrial scale which should be kept secret. Smart factories use WSNs and edge computation to create new production forms with better efficiency and flexibility. The total number of sensor devices usually represents industrial production scale in a smart factory. Usually, control center is a third-party service from the cloud or a regulatory agency from the government side. Trade secrets can be learned and used by rivals if the scale of a factory's production is disclosed. Therefore, it is necessary to compute statistical aggregation functions without exposing the total number of WSN sensor devices. In such a scenario, the control center could use statistical data for scientific analysis and intelligent decision-making but would not have any data about the industrial production scale of the smart factory.

In this article, we propose the first privacy-preserving statistical aggregation scheme without revealing the total number of sensor devices to control center for edge computing-enhanced WSNs. The contributions of this article can be summarized as follows:

(i) We construct an efficient privacy-preserving statistical aggregation scheme based on the Paillier additive homomorphic encryption scheme and the ECDSA digital signature scheme, called EPPSA. The EPPSA scheme supports multiple statistical aggregation functions, including arithmetic mean, quadratic mean, weighted mean, and variance.

(ii) In the EPPSA scheme, the mean values can be calculated by the edge device and control center cooperatively, while control center does not know the total number of sensor devices. Firstly, the edge device computes the mean value in ciphertext since it has calculated the sum of the data in ciphertext.. Secondly, after receiving the mean in ciphertext, control center calculates the correct mean by using the modified extended Euclidean algorithm to process the decrypted mean. The EPPSA scheme avoids calculating sum/total number and the total number of WSN sensor devices can be kept secret to control center.

(iii) In the EPPSA scheme, we propose three modified Montgomery exponentiation algorithms to improve the aggregation efficiency in the edge device. Our idea is to avoid converting the data between the Montgomery domain and residue domain frequently during the whole process. The ciphertext data in the Montgomery domain can be aggregated by Montgomery multiplications, which are more efficient than ordinary modular multiplications.

(iv) We implement the EPPSA scheme and compare it with the existing schemes. Compared with [28], the EPPSA scheme gets 62.5% aggregation performance improvement for 1024 bits modulus. Compared with [36], the EPPSA scheme gets 50% and 33% communication load decrease on arithmetic mean and variance statistics, respectively.

The rest of this article is organized as follows: In Section 2, the problem formulation is presented. In Section 3, the related preliminaries are reviewed. In Section 4, the proposed EPPSA data aggregation scheme is given. In Section 5, the secure analysis is given. In Section 6, the performance evaluation and comparison are presented. Finally, Section 7 concludes this article.

## 2. Problem Formulation

In this section, the formalized system model, the security requirements, and design goals are presented.

*2.1. System Model.* In the EPPSA scheme, a WSN system is comprised of four parts, namely trusted authority (TA), control center (CC), edge aggregator (EA), and sensor device (SD). The system describes a three-level topological structure, as shown in Figure 1.

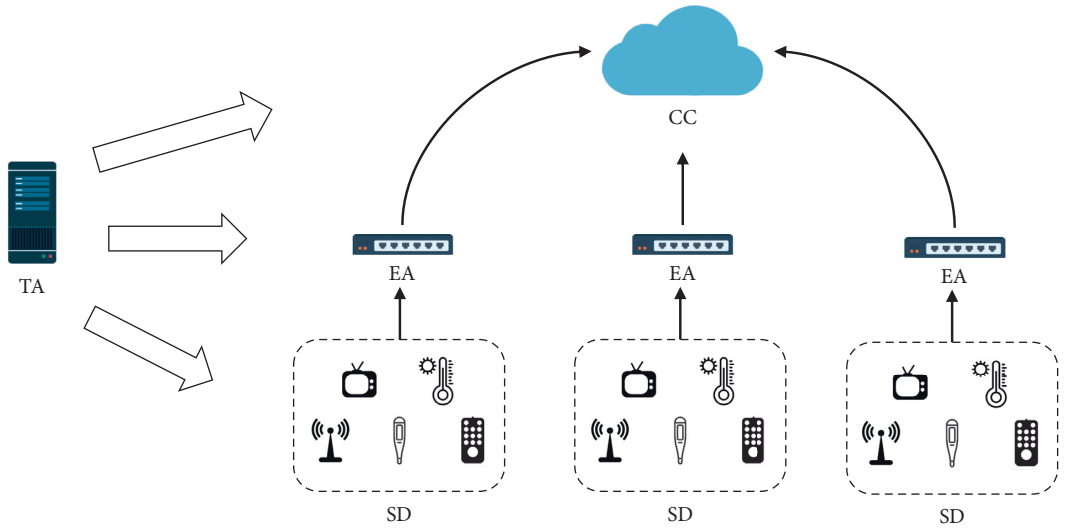


Figure 1: System model.

(i) TA is a trusted third party, which is responsible for generating and distributing the secret keys to all the system participants. In the phase of system initialization, TA sets the ECDSA key pairs into the sensor devices, edge devices, and control center. TA distributes the Paillier public key to the sensor devices, edge devices, and the Paillier private key to control center separately by sending digital envelopes over the Internet.

(ii) CC is a powerful service controller of a WSN sensing system. According to special application requirements, CC is responsible for analyzing the data statistics, for example, data mining. CC is assumed to be honest-but-curious. It means that CC attempts to mine valuable information while performing its specified tasks.

(iii) EA is a wireless receiving equipment that is deployed at the edge of the WSN. EA is responsible for collection, aggregation, and transmission of sensor data. EA collects encrypted data from sensor devices, aggregates the data, and transmits the aggregated data to CC. EA is a high-performance computing device so that it can perform computationally expensive processes.

(iv) SD is deployed at the intended area and is responsible for sensing and communication. SDs automatically sense and encrypt the particular data before sending them to EA. For example, ambient temperature sensors record the real-time temperature in an intelligent agricultural system and report the encrypted data to CC via EA.

*2.2. Security Requirements.* In our system model, EA and CC are curious about SD's privacy data, but they cannot collude with each other. Moreover, there is an adversary $\alpha$ assumed to have the capability to eavesdrop on data during their transit. To protect data against internal and external attacks, the following security requirements should be fulfilled:

(i) *Data confidentiality*. Even though data from SDs or EA is eavesdropped on by $\alpha$ during their transit, they cannot be identified. EA cannot infer the privacy information of SDs while aggregating statistic data. When CC receives the statistics data, for example, mean, variance, it cannot identify the individual data or number of SDs.

(ii) *Authentication*. It should be guaranteed that the data are generated by legitimate SD entities. Otherwise, malicious operations from $\alpha$, for example, replay attack, may undermine the accuracy of the statistics. Similarly, the aggregate data should be guaranteed to be generated by a legitimate EA.

(iii) *Data integrity*. Accuracy and completeness of data in transmission should be guaranteed. When an adversary $\alpha$ forges or modifies the data, the malicious operations should be detected by the receiver.

*2.3. Design Goal.* Our design goal is to design an efficient privacy-preserving statistical aggregation scheme. The following design goals should be achieved:

(i) *Security*. The proposed scheme should satisfy the secure requirements mentioned above. The security goal is to prevent individual data and statistical data from being stolen by the adversary. In order to achieve this security goal, both internal and external behavior should be detected.

(ii) *Efficiency*. The proposed scheme should consider computation cost and communication load. On one hand, it is necessary to use lightweight encryption and signing primitives. On the other hand, methods should be adopted to reduce the consumption of aggregate computation.

(iii) *Statistical aggregation*. A series of data statistical functions should be supported by the proposed scheme. In an actual scenario, statistics of measurement indicators, such as mean, weighted mean, and variance, are essential for analysis. Meanwhile, except for statistics, the CC should not get any other information.

## 3. Preliminaries

*3.1. The Paillier Cryptosystem.* The Paillier cryptosystem is a widely used public key encryption scheme with additive homomorphic property [37] and is standardized by International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) in 2019 [38]. The Paillier cryptosystem consists of three parts, namely key generation, encryption, and decryption, which are described in Scheme 1.

The security of the Paillier encryption algorithm is based on the integer factoring problem. When choosing the parameter $g$, it is necessary to judge whether $n$ is divisible by the order of $g$. This can be efficiently checked by testing whether $gcd(L(g^{\lambda} \bmod n^2), n) = 1$, where function $gcd(.)$ is the greatest common divisor function.

The Paillier cryptosystem has several interesting homomorphic properties, which are associated with the statistics given below:

$$\mathrm{Dec}\left(Enc(d_1)Enc(d_2) \bmod n^2\right) = d_1 + d_2 \bmod n. \quad (1)$$

*3.2. Mean Value Computation on Ciphertext of Paillier Cryptosystem.* Shah et al. [39] proposed a solution for noninteger mean value computation in the homomorphic encrypted domain. This method can be adopted by statistical aggregation scheme in WSNs. Let $(d_1, d_2, \ldots, d_m)$ be a set of numbers. The mean value, denoted by $d_{\mathrm{mean}}$, is the sum of the values divided by the total number of elements, $\mathrm{d}_{\mathrm{mean}} = \sum_{i=1}^{i=m} \mathrm{d}_i/m$. In practice, the mean $d_{\mathrm{mean}}$ may result in integer or float value. Using the homomorphic property of the Paillier cryptosystem given in (2), the mean can be calculated in the encrypted domain.

$$Enc(d_{\mathrm{mean}}) = \left(\prod_{i=1}^{i=m} Enc(d_i)\right)^{m^{-1} \bmod n} \bmod n^2. \quad (2)$$

If the plain domain mean $d_{\mathrm{mean}}$ is an integer, the encrypted domain mean $Enc(d_{\mathrm{mean}})$ calculated by (2) results in the correct mean $d_{\mathrm{mean}}$ after decryption. However, if the plain domain mean $d_{\mathrm{mean}}$ is a decimal, the encrypted domain mean $Enc(d_{\mathrm{mean}})$ calculated by (2) results in a large integer after decryption. For example, $d_{mean} = \alpha/\beta$, where $\alpha$ is not divisible by $\beta$. After decryption, $Enc(d_{mean})$ will result in $\alpha\beta^{-1} \bmod n^2$, which is a large integer. Reducing the large integer to the correct mean value is a two-dimensional lattice reduction problem and can be solved by the Lagrange-Gauss lattice reduction algorithm. Shah et al. proposed an efficient method to reduce the large integer called the modified extended Euclidean algorithm ($MME$). The method is shown in Algorithm 1.

| Scheme | Computation complexity |
|---|---|
| EPPSA | $(m-1) \cdot T_{MM}$ |
| [30] | $(m-1) \cdot T_{OMM}$ |
| [33] | $(m-1) \cdot T_{OMM}$ |
| [35] | $(m-1) \cdot T_{MM} + T_{ME}$ |

Scheme 1: The Paillier cryptosystem.

The modulus $n$ of Paillier cryptosystem and large integer value $w$ can be considered as independent points in a two-dimensional lattice space ✓. These two basis vectors, $(0, n)$ and $(1, w)$, can be reduced for optimal values. Algorithm 1 computes the reduced value of $w$ using adapted extended Euclidean algorithm, which is the correct mean value.

*3.3. Montgomery Multiplication.* Montgomery multiplication (MM) is an efficient technique for computing modular multiplications [40]. Assuming an odd modulus $n$ is a $t$-bit number, let $r = 2^t$. For integers $0 < a, b < k$, the Montgomery multiplication is $MM(a, b) = ab2^{-t} \bmod n$. By taking $r$ as a power of 2, the division becomes simple shifting. The process of MM is presented in Algorithm 2.

Utilizing the MM algorithm, the Montgomery exponentiation is present in Algorithm 3. For a number $\alpha$, the corresponding number in the Montgomery domain is denoted by $\overline{\alpha}$.

## 4. The Proposed EPPSA Scheme

In this section, we propose the first privacy-preserving statistical aggregation scheme without revealing the total number of sensor devices to control center. In order to achieve the security goals, the edge device and control center calculates the statistics cooperatively, while control center does not know the total number of sensor devices. The Paillier cryptosystem is used as the encryption scheme and the ECDSA algorithm [41] is used as the signature scheme.

The EPPSA scheme consists of four phases including system initialization, data encryption, secure statistical aggregation, and secure statistics reading. In the system initialization phase, TA initializes the WSN system by generating and distributing the secret keys of the Paillier and ECDSA algorithms. In the data encryption phase, sensor device $SD_i$ collects raw data and encrypts these data to generate a data report. Then sensor device sends the encrypted data report to EA via wireless networks. In the secure statistical aggregation phase, EA calculates sum and mean value in the encrypted domain and sends the statistical report to CC. In this phase, EA does not reveal the total

Input: $n$, $w$, where $n$ is the modulus and $w$ is the large number.
Output: $R_w = \mathrm{MEE}(n, w)$.
(1) $(x_1, x_2) = (0, n)$
(2) $(y_1, y_2) = (1, w)$
(3) $Q = [x_2/y_2]$
(4) $(tmp_1, tmp_2) = (x_1, x_2) - Q(y_1, y_2)$
(5) $(x_1, x_2) = (y_1, y_2)$
(6) $(y_1, y_2) = (tmp_1, tmp_2)$
(7) while $w > \sqrt{n}$ do
(8) $Q = [x_2/y_2]$
(9) $(tmp_1, tmp_2) = (x_1, x_2) - Q(y_1, y_2)$
(10) $(x_1, x_2) = (y_1, y_2)$
(11) $(y_1, y_2) = (tmp_1, tmp_2)$
(12) end while
(13) return $R_w = x_2/x_1$

Algorithm 1: The reduction based on modified extended Euclidean algorithm

(i) Input: $a$, $b$, $n'$, where $n'$ is computed by the extended Euclidean algorithm.
Output: $v = MM(a, b)$.
(1) $s = ab$
(2) $m = sn' \bmod r$
(3) $u = (s + mk)/r$
(4) if $v \geq n$, then return $v - n$
(ii) else return $v$

Algorithm 2: The Montgomery multiplication

(i) Input: $\gamma$, $e$, $n$, where $n'$ is computed by the extended Euclidean algorithm.
Output: $\theta = \gamma^e \bmod n$.
(1) $\overline{\gamma} = \gamma.r \bmod n$
(2) $\overline{\theta} = 1.r \bmod n$
(3) For $i = t - 1$ down to 0
(4) $\overline{\theta} = MM(\overline{\theta}, \overline{\theta})$
(5) if $e_i = 1$, then $\overline{\theta} = MM(\overline{\gamma}, \overline{\theta})$
(6) $\theta = MM(\overline{\theta}, 1)$
(7) return $\theta$

Algorithm 3: The Montgomery exponentiation

number of sensor devices to CC. In the secure statistics reading phase, CC decrypts the statistical report and calculates the quadratic mean and variance of each dimension. Finally, CC gets all the arithmetic mean, quadratic mean, weighted mean, and variance without knowing the total number of sensor devices. Furthermore, to achieve the improvement in aggregation performance, we present three modified Montgomery exponentiation algorithms. Using these algorithms, EPPSA avoids frequent conversion of exponentiation results between the Montgomery domain and residue domain.

*4.1. Modified Montgomery Exponentiation Algorithms.* We modified Algorithm 3 to improve the aggregation performance. Three modified algorithms below map the result of modular exponentiation into the Montgomery domain.

*4.1.1. Modified Montgomery exponentiation 1.* The modified Montgomery exponentiation method 1 (MME1) is described in Algorithm 4.

Compared with Algorithm 3, Algorithm 4 removes the step $\theta = MM(\overline{\theta}, 1)$, which converts the result into the correct domain $z_n^*$. This denotes that the exponentiation result is still in the Montgomery domain. The result of exponentiation is denoted by $\overline{\theta}$ to be distinguished from the one in Algorithm 3.

*4.1.2. Modified Montgomery Exponentiation 2.* The modified Montgomery exponentiation method 2 (MME2) is described in Algorithm 5.

Compared with Algorithm 3, Algorithm 5 removes the step $\overline{\gamma} = MM(\gamma)$ and $\theta = MM(\overline{\theta}, 1)$. The base number and result of Algorithm 5 are both in the Montgomery domain and are denoted by $\overline{\gamma}$ and $\overline{\theta}$, respectively, to be distinguished from the ones in Algorithm 3.

*4.1.3. Modified Montgomery Exponentiation 3.* The modified Montgomery exponentiation method 3 (MME3) is described in Algorithm 6.

Compared with Algorithm 3, Algorithm 6 removes the step $\overline{\gamma} = MM(\gamma)$. The base number of Algorithm 6 is in the Montgomery domain and is denoted by $\overline{\gamma}$ to be distinguished from the one in Algorithm 3.

Using these algorithms, encrypted data are converted to the Montgomery domain at the beginning of the process during the process. Then encrypted data are kept in the Montgomery domain for further computation. In the end, the results are reconverted back to the residue (non-Montgomery) domain. By reducing the conversions between the Montgomery domain and the residue domain, the aggregation operation can be accelerated.

*4.2. System Initialization.* In the proposed system model, we assume that there are $m$ SDs in WSN, which are denoted by $D_i$, $(1 \leq i \leq m)$. Each device $D_i$ generates an $l$-dimensional data vector $d_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,j}, \ldots, d_{i,l})$. Each $D_i$ gets an identity $ID_i$ and EA gets an identity $ID_{EA}$. The data in a region can be denoted by a matrix

$$D = \begin{bmatrix} d_{1,1} & \cdots & d_{1,l} \\ \vdots & \ddots & \vdots \\ d_{m,1} & \cdots & d_{m,l} \end{bmatrix}. \quad (3)$$

Given secure parameters $\kappa$ and $\kappa_1$, TA initializes the parameters of the additive homomorphic encryption algorithm and digital signature algorithm. The key generation procedure is shown as follows:

(i) Input: $\gamma$, $e$, $n$, where $n'$ is computed by the extended Euclidean algorithm.
Output: $\overline{\theta}$
(1) $\overline{\gamma} = \gamma \cdot r \bmod n$
(2) $\overline{\theta} = 1 \cdot r \bmod n$
(3) For $i = t - 1$ down to 0
(4) $\overline{\theta} = MM(\overline{\theta}, \overline{\theta})$
(5) if $e_i = 1$, then $\overline{\theta} = MM(\overline{\gamma}, \overline{\theta})$
(6) return $\overline{\theta}$

Algorithm 4: The modified Montgomery exponentiation 1 (MME1)

(i) Input: $\overline{\gamma}$, $e$, $n$, where $n'$ is computed by the extended Euclidean algorithm.
Output: $\overline{\theta}$
(1) $\overline{\theta} = 1 \cdot r \bmod n$
(2) For $i = t - 1$ down to 0
(3) $\overline{\theta} = MM(\overline{\theta}, \overline{\theta})$
(4) if $e_i = 1$, then $\overline{\theta} = MM(\overline{\gamma}, \overline{\theta})$
(5) return $\overline{\theta}$

Algorithm 5: The modified Montgomery exponentiation 2 (MME2)

(i) Input: $\overline{\gamma}$, $e$, $n$, where $n'$ is computed by the extended Euclidean algorithm.
Output: $\theta$
(1) $\overline{\theta} = 1 \cdot r \bmod n$
(2) For $i = t - 1$ down to 0
(3) $\overline{\theta} = MM(\overline{\theta}, \overline{\theta})$
(4) if $e_i = 1$, then $\overline{\theta} = MM(\overline{\gamma}, \overline{\theta})$
(5) $\theta = MM(\overline{\theta}, 1)$
(6) return $\theta$

Algorithm 6: The modified Montgomery exponentiation 3 (MME3)

Step 1: TA chooses prime numbers $p$, $q$ randomly, where $|p| = |q| = \kappa$. Let $n = pq$ and $\lambda = lcm(p - 1, q - 1)$. Choose $g$, with $g \in Z^*_{n^2}$, and the order of $g$ is a multiple of $n$. Then, TA generates the encryption key $(pk_{AHE}, sk_{AHE})$, where the encryption public key is $pk_{AHE} = (n, g)$ and decryption private key is $sk_{AHE} = (p, q, \lambda)$.

Step 2: TA chooses an Elliptic curve group $\Gamma$ of an order $q_1$ with base point (generator) $G$, which is over the finite field $Z_{p_1}$ of integers modulo a prime $p_1$. The bit length of $q_1$ and $p_1$ should be set as the security parameter, that is, $|p_1| = |q_1| = \kappa_1$. For each $SD_i$ $(1 \le i \le m)$, TA chooses a secret key of digital signature $sk_{DS,i} \leftarrow Z_{q_1}$ randomly. TA sets the public key of the digital signature $pk_{DS,i} = sk_i \cdot G$. The signature key of $SD_i$ is $(pk_{DS,i}, sk_{DS,i})$. The signature keys of EA, CC, and TA are generated in the same way, which are denoted by $(pk_{DS,EA}, sk_{DS,EA})$, $(pk_{DS,CC}, sk_{DS,CC})$, and $(pk_{DS,TA}, sk_{DS,TA})$, respectively. The signature algorithm makes use of a hash function $H: \{0,1\}^* \longrightarrow Z_{q_1}$.

Step 3: Via a secure channel, TA sends the encryption public key $pk_{AHE}$ and the signature private key $sk_{DS,i}$ to $SD_i$ $(1 \le i \le m)$. It sends the encryption public key $pk_{AHE}$, the signature public key $pk_{DS,i}$, and the signature private key $sk_{DS,EA}$ to EA. It sends the decryption private key $sk_{AHE}$ and the signature public key $pk_{DS,EA}$ to CC.

After key generation, TA distributes the encryption keys and signing keys. The key distribution procedure is shown as follows:

Step 1: TA writes signature key pair $(pk_{DS,i}, sk_{DS,i})$ into the senor device $SD_i$ $(1 \le i \le m)$ before deploying the sensor device. TA writes the signature public key $pk_{DS,i}$ and the signature key pair $(pk_{DS,EA}, sk_{DS,EA})$ into EA before deploying the edge device. TA sends the signature public key $pk_{DS,TA}$ and $pk_{DS,EA}$ to CC through the Internet and give the signature key pair $(pk_{DS,CC}, sk_{DS,CC})$ to CC by a USB key device.

Step 2: Using the private key $sk_{DS,TA}$, TA computes a digital signature on $sk_{AHE}$ denoted by $\sigma_{AHE}$. Using CC's public key $pk_{DS,CC}$, TA generates a digital envelope on the Paillier private key $sk_{AHE}$ and the signature $\sigma_{AHE}$ denoted by $\sigma_{DE}$. TA sends the $\sigma_{DE}$ to CC through the Internet.

Step 3: After receiving the digital envelope $\sigma_{DE}$, CC decrypts it and gets the Paillier private key $sk_{AHE}$ and the signature $\sigma_{AHE}$. Using the public key $pk_{DS,TA}$, CC verifies the signature. If the verification is passed, the Pallier private key will be accepted.

*4.3. Data Report Generation.* Each sensor device $SD_i$, $(1 \le i \le m)$, performs the following phases to get a data report:

(i) *Generate*: The $SD_i$ firstly generates the raw data vector $d_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,j}, \ldots d_{i,l})$. Then $SD_i$ calculates the corresponding quadratic data vector $d_i^2 = (d_{i,1}^2, d_{i,2}^2, \ldots, d_{i,j}^2, \ldots d_{i,l}^2)$. Given a weight vector $w = (w_1, w_2, \ldots, w_i, \ldots w_m)$, $SD_i$ calculates the weighted data vector $d_{i,\text{wei}} = (d_{i,1,\text{wei}}, d_{i,2,\text{wei}}, \ldots, d_{i,j,\text{wei}}, \ldots d_{i,l,\text{wei}})$ by $d_{i,j,wei} = d_{i,j} w_j$.

(ii) *Encrypt*: After generating the $l$-dimensional data vectors $d_i$, $d_i^2$, and $d_{i,wei}$, sensor device $SD_i$ encrypts the data using the Paillier encryption algorithm. When calculating the ciphertexts, EPPSA uses MME1 to convert the results to the Montgomery domain. These result of $d_i$, $d_i^2$, and $d_{i,wei}$ are denoted by $\overline{c_i} = (\overline{c_{i,1}}, \ldots, \overline{c_{i,l}})$, $\overline{c_i^2} = \overline{c_{i,1}^2}, \ldots, \overline{c_{i,l}^2}$, and $\overline{c_{i,wei}} = \overline{c_{i,1,wei}}, \ldots, \overline{c_{i,l,wei}})$, respectively.

(iii) *Sign*: Timestamp is denoted by $TS$, and the identity of $SD_i$ is denoted by $ID_i$. $SD_i$ chooses an instance key $k_i \leftarrow Z_{q_1}$. Calculate $(r_{x,i}, r_{y,i}) = k_i G$ and

$sig_i = (H(\overline{c_{i,1}}\|\ldots\|\overline{c_{i,l}}\|\overline{c_{i,1}^2}\|\ldots\|\overline{c_{i,l}^2}\|\overline{c_{i,1,wei}}\|\ldots\|\overline{c_{i,l,wei}}\|TS\|\mathrm{ID}_i) + sk_{DS,i}r_{x,i})/k_i$. The signature is achieved by $\sigma_i = (sig_i \bmod q_1, r_{x,i} \bmod q_1)$.

(iv) *Send*: $\mathrm{SD}_i$ sends the data report $(\overline{c_i}, \overline{c_i^2}, \overline{c_{i,wei}}, \sigma_i, \text{timestamp}, ID_i)$ to EA.

*4.4. Statistical Aggregation.* After receiving the data $(\overline{c_i}, \overline{c_i^2}, \overline{c_{i,wei}}\ \sigma_i, TS, \mathrm{ID}_i)$ reported from $m$ sensor devices, EA performs the following steps to generate the statistical aggregation report:

(i) *Verify*: EA firstly calculates $(r'_{x,i}, r'_{y,i}) = \mathrm{G}/(sig_i \cdot H(\overline{c_{i,1}}\|\ldots\|\overline{c_{i,l}}\|\overline{c_{i,1}^2}\|\ldots\|\overline{c_{i,l}^2}\|\overline{c_{i,1,wei}}\|\ldots\|\overline{c_{i,l,wei}}\|\text{timestamp}\|\mathrm{ID}_i) + pk_{DS,i}/sig_i \cdot r_{x,i}$. Then, EA checks the validity of $(\overline{c_i}, \sigma_i t, \text{timestamp}, \mathrm{ID}_i)$ by verifying the equation $r'_{x,i} \bmod q_1 = r_{x,i} \bmod q_1$.

(ii) *Aggregate*: If the validity equation holds, EA executes the aggregation operations. EA firstly calculates the arithmetic sum, quadratic sum, and weighted sum of each dimension, which are denoted by $\overline{c_{j,\mathrm{sum}}}$, $\overline{c_{j,q\mathrm{sum}}}$, and $\overline{c_{j,w\mathrm{sum}}}$, $(1 \le j \le l)$, respectively. When calculating the sum, EA uses the $MM$ method (Algorithm 3) for modular multiplication. Then EA calculates arithmetic mean, quadratic mean, and weighted mean of each dimension, which are denoted by $\overline{c_{j,mea}}$, $\overline{c_{j,qmea}}$, and $\overline{c_{j,wmea}}$, $(1 \le j \le l)$, respectively. When calculating the mean by Equation 2, EA uses $MME2$ (Algorithm 5) for modular exponentiation. The result is denoted by $\overline{c_{EA}} = (\overline{c_{1,mea}}, \ldots, \overline{c_{l,mea}}, \overline{c_{1,qmea}}, \ldots, \overline{c_{l,qmea}}, \overline{c_{1,wmea}}, \ldots, \overline{c_{l,wmea}})$. The details of the aggregation are shown in Algorithm 7.

(iii) *Sign*: Timestamp is denoted by $TS$, and the identity of EA is denoted by $ID_{EA}$. EA chooses an instance key $k_{EA} \leftarrow Z_{q_1}$. Calculate $(r_{x,EA}, r_{y,EA}) = k_{EA}G$ and $sig_{EA} = (H(\overline{c_{1,mea}}\|\ldots\|\overline{c_{l,mea}}\|\overline{c_{1,qmea}}\|\ldots\|\overline{c_{l,qmea}}\|\overline{c_{1,wmea}}\|\ldots\|\overline{c_{l,wmea}}\|TS\|ID_{EA}) + sk_{EA}r_{x,EA})/k_{EA}$. The signature is achieved by $\sigma_{EA} = (sig_{EA} \bmod q_1, r_{x,EA} \bmod q_1)$.

(iv) *Send*: EA sends the data report $(\overline{c_{EA}}, \sigma_{EA}, TS, ID_{EA})$ to CC.

*4.5. Statistical Report Decryption.* After receiving the data report $(\overline{c_{EA}}, \sigma_{EA}, TS, ID_{EA})$ reported from EA, EA performs the following steps to decrypt the statistical aggregation report:

(i) *Verify*: EA firstly calculates $(r'_{x,EA}, r'_{y,EA}) = \mathrm{G}/(sig_{EA} \cdot H(\overline{c_{1,mea}}\|\ldots\|\overline{c_{l,mea}}\|\overline{c_{1,qmea}}\|\ldots\|\overline{c_{l,qmea}}\|\overline{c_{1,wmea}}\|\ldots\|\overline{c_{l,wmea}}\|TS\|ID_{EA}) + pk_{EA}/sig_{EA} \cdot r_{x,EA}$. The EA checks the validity of $(\overline{c_{EA}}, \sigma_{EA}, TS, ID_i)$ by verifying the equation $r'_{x,EA} \bmod q_1 = r_{x,EA} \bmod q_1$.

(ii) *Decrypt*: If the validity equation holds, CC executes the decryption operations using $d = D(c) = L(c^\lambda \bmod n^2)\mu \bmod n$. When calculating the decryption, CC uses the MME3 (Algorithm 6) for modular exponentiation. The result is $(d_{1,}\ mea, \ldots, d_{l,mea}, d_{1,qmea}, \ldots, d_{l,qmea}, d_{1,wmea}, \ldots, d_{l,wmea})$

(iii) *Reduce*: Considering that the decryption result of the mean may be a large integer with no sense, CC reduces each decryption result using Algorithm 2. CC takes elements in the decrypted data vector $(d_{1,mea}, \ldots, d_{l,mea}, d_{1,qmea}, \ldots, d_{l,qmea}, d_{1,wmea}, \ldots, d_{l,wmea})$ and the modulus as inputs and gets the reduced data vector $(D_{1,mea}, \ldots, D_{l,mea}, D_{1,qmea}, \ldots, D_{l,qmea}, D_{1,wmea}, \ldots, D_{l,wmea})$.

(iv) *Post-Process*: The reduced data vector includes arithmetic mean, mean of square, and weighted mean of each dimension. For each dimension, CC calculates the quadratic mean $D_{j,Qmea}$ by equation (3) and variance $D_{j,var}$ by equation (4). Finally, CC gets the result of arithmetic mean, quadratic mean, weighted mean, and variance of each dimension, denoted by $(D_{1,mea}, \ldots, D_{l,mea}, D_{1,Qmea}, \ldots, D_{l,Qmea}, D_{1,wmea}, \ldots, D_{l,wmea}, D_{1,var}, \ldots, D_{l,var})$.

$$D_{j,Qmea} = \sqrt{D_{j,qmea}}, \tag{4}$$

$$D_{j,var} = D_{j,qmea} - \left(D_{j,mea}\right)^2, \tag{5}$$

## 5. Security Analysis

In this section, we analyze the security properties of the proposed EPPSA scheme, following the security requirements and design goals given in Section 2.

**Lemma 1.** *The result of encryption in the Montgomery domain is a valid format of the ciphertext.*

*Proof 1.* The residue (non-Montgomery) system is a commutative ring denoted by $R_n$, and the Montgomery domain is a commutative ring denoted by $R_n^M$. The rings $R_n$ and $R_n^M$ are isomorphic by the isomorphism $h$: $R_n \longrightarrow R_n^M$ defined by $h(a) = a \cdot 2^t \bmod n$ and $h^{-1}$: $R_n^M \longrightarrow R_n$ defined by $h(a) = a \cdot 2^{-t} \bmod n$. Due to the isomorphism, the result of encryption in the Montgomery domain is a valid format of the ciphertext. □

*5.1. Resistance to Eavesdropping Attack*

**Theorem 1.** *WSN devices' private data and statistics cannot be obtained by an adversary $\alpha$ even if it is eavesdropped during transmitting.*

*Proof 2.* In the EPPSA scheme, the data are encrypted by the Paillier cryptosystem. According to Lamma 1, the result of encryption $c_{i,j} = g^{d_{i,j}} r^n \bmod n$ in the Montgomery domain is

(i) Input: Vectors $\overline{c_i} = (\overline{c_{i,1}}, \ldots, \overline{c_{i,l}}, \overline{c_{i,1}^2}, \ldots, \overline{c_{i,l}^2}, \overline{c_{i,1,wei}}, \ldots, \overline{c_{i,l,wei}})$, where $1 \le i \le m$.
Output: Ciphertext of arithmetic mean, mean of square, and weighted mean in the Montgomery domain of each dimension, denoted by $\overline{c_{j,mea}}$, $\overline{c_{j,qmea}}$, and $\overline{c_{j,wmea}}$, $(1 \le j \le l)$.
(1) $\overline{c_{j,\mathrm{sum}}} = \overline{c_{i,1}}$, $\overline{c_{j,q\mathrm{sum}}} = \overline{c_{i,1}^2}$, $\overline{c_{j,w\mathrm{sum}}} = \overline{c_{i,1,wei}}$, $(1 \le j \le l)$
(2) for $j = 1$ up to $j = l$
(3) for $i = 2$ up to $i = m$
(4) $\overline{c_{j,\mathrm{sum}}} = MM(\overline{c_{j,\mathrm{sum}}}, \overline{c_{i,j}})$,
(5) $\overline{c_{j,mea}} = \mathrm{MME2}(\overline{c_{j,\mathrm{sum}}}, m^{-1} \bmod n, n^2)$
(6) for $j = 1$ up to $j = l$
(ii) for $i = 2$ up to $i = m$
(8) $\overline{c_{j,q\mathrm{sum}}} = MM(\overline{c_{j,q\mathrm{sum}}}, \overline{tc_{i,j}^2})$,
(9) $\overline{c_{j,qmea}} = \mathrm{MME2}(\overline{c_{j,q\mathrm{sum}}}, m^{-1} \bmod n, n^2)$
(10) for $j = 1$ up to $j = l$
(11) for $i = 2$ up to $i = m$
(12) $\overline{c_{j,w\mathrm{sum}}} = MM(\overline{c_{j,w\mathrm{sum}}}, \overline{c_{i,j,wei}})$,
(13) $\overline{c_{j,wmea}} = \mathrm{MME2}(\overline{c_{j,w\mathrm{sum}}}, m^{-1} \bmod n, n^2)$
(14) return $\overline{c_{EA}} = (\overline{c_{1,mea}}, \ldots, \overline{c_{l,mea}}, \overline{c_{1,qmea}}, \ldots, \overline{c_{l,qmea}}, \overline{c_{1,wmea}}, \ldots, \overline{c_{l,wmea}})$

Algorithm 7: Statistical aggregation

a valid format of the ciphertext. Meanwhile, the private key $sk_{AHE}$ is transmitted to CC in digital envelope. The $sk_{AHE}$ is encrypted by CC's public key $pk_{DS,CC}$ so that $\alpha$ cannot get it. Since Paillier cryptosystem is provably secure against the chosen plaintext attack based on the decisional Diffie—Hellman problem, $\alpha$ cannot guess the plaintext in a nonnegligible probability without the private key $sk_{AHE}$. Similarly, $\alpha$ cannot obtain statistics by eavesdropping on the transmission between EA and CC. In a word, the data and statistics in transmission are semantically secure. □

### 5.2. Resistance to Replay Attack

**Theorem 2.** *If a replayed data report is transmitted to EA, or a statistical report to CC, it can be detected.*

*Proof 3.* If an adversary $\alpha$ replays the data report $(\overline{c_{re}}, \sigma_{\mathrm{new}}, TS_{\mathrm{new}})$ to aggregator EA, it needs to forge a new timestamp donated by $\mathrm{timestamp}_{\mathrm{new}}$. Since the timestamp is new, $\alpha$ has to forge a new signature $\sigma_{\mathrm{new}}$ of the replayed ciphertext $\overline{c_{re}}$. The security of the ECDSA system is based on the computational intractability of the discrete logarithm problem (DLP). The signature key pair $(pk_{DS,i}, sk_{DS,i})$ is written to $\mathrm{SD}_i$ directly when system initialization. Thus, $\alpha$ cannot guess the correct signature of the replayed report in a nonnegligible probability without the private key $sk_{DS,i}$. Similarly, the replay attack of the statistical report to CC can be detected for the same reason. In a word, the EPPSA scheme is resistant to replay attack. □

### 5.3. Resistance to Manipulation Attack

**Theorem 3.** *If an adversary α manipulates the data report from WSN senor device or statistics from EG, it can be detected.*

*Proof 4.* It is assumed that an adversary $\alpha$ manipulates the encrypted data $\overline{c_i} = (\overline{c_{i,1}}, \ldots, \overline{c_{i,l}}, \overline{c_{i,1}^2}, \ldots, \overline{c_{i,l}^2}, \overline{c_{i,1,wei}}, \ldots, \overline{c_{i,l,wei}})$s during the transmission to aggregator EA. When receiving the data report, EA calculates the hash value $H(\overline{c_{i,1}}\| \ldots \|\overline{c_{i,l}}\|\overline{c_{i,1}^2}\| \ldots \|\overline{c_{i,l}^2}\|\overline{c_{i,1,wei}}\| \ldots \|\overline{c_{i,l,wei}}\|TS\|\mathrm{ID}_i)$ and checks the signature $\sigma_i$ by verifying the equation $r_{x,i}' \bmod q_1 = r_{x,i} \bmod q_1$. If $\overline{c_i}$ is manipulated, the hash value will be incorrect and the signature will not be validated. Similarly, when receiving statistical report from EA, CC calculates the hash value $H(\overline{c_{1,mea}}\| \ldots \|\overline{c_{l,mea}}\|\overline{c_{1,qmea}}\| \ldots \|\overline{c_{l,qmea}}\|\overline{c_{1,wmea}}\| \ldots \|\overline{c_{l,wmea}}\|TS\|\mathrm{ID}_{EA})$ and checks the signature $\sigma_i$ by verifying the equation $r_{x,EA}' \bmod q_1 = r_{x,EA} \bmod q_1$. The statistical report is considered invalid if it is manipulated by $\alpha$. In a word, the integrity of data and statistics can be satisfied. □

### 5.4. Resistance to Internal Attack

**Theorem 4.** *If EA is an internal attacker which is curious about WSN devices' privacy data, it still cannot obtain the actual data of the devices.*

*Proof 5.* According to Lemma 1, the encrypted data from WSN sensor devices $\overline{c_i} = (\overline{c_{i,1}}, \ldots, \overline{c_{i,l}}, \overline{c_{i,1}^2}, \ldots, \overline{c_{i,l}^2}, \overline{c_{i,1,wei}}, \ldots, \overline{c_{i,l,wei}})$ in the Montgomery domain is a valid format of the ciphertext. The aggregator EA does not have the private key to decrypt the ciphertext. □

**Theorem 5.** *If CC is an internal attacker which is curious about the total number of WSN devices, it still cannot obtain the actual number of the devices.*

*Proof 6.* CC obtains the arithmetic mean, quadratic mean, and weighted mean by decryption and reduction. Also, CC calculates the variance by Equation (3), which uses plaintext of arithmetic mean and quadratic mean. In a word, CC does

Table 1: Aggregation computation complexity of MMDA and EPPSA.

| Scheme | Computation complexity |
|---|---|
| EPPSA | $(m-1)\cdot T_{MM}$ |
| [28] | $(m-1)\cdot T_{OMM}$ |
| [31] | $(m-1)\cdot T_{OMM}$ |
| [33] | $(m-1)T_{MM}+T_{ME}$ |

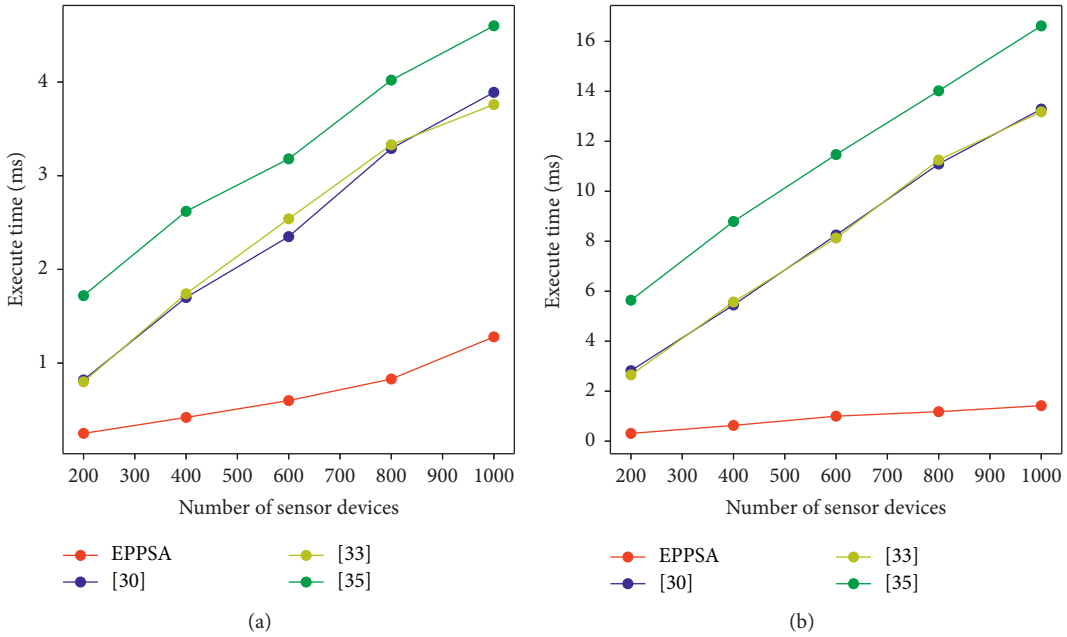


Figure 2: Comparison of aggregation computation costs: (a) comparison of aggregation computation cost on 1024 bits and (b) comparison of aggregation computation cost on 2048 bits.

not get any information on the total number of WSN devices when calculating statistics. □

## 6. Performance Evaluation and Comparison

In this section, our scheme is evaluated in terms of computation costs and communication costs. The performance results are compared with the scheme proposed in references [28, 31, 33, 36].

*6.1. Computation Cost.* Assume that there are $m$ sensor devices $SD_i$ in the system and each of them reports an $l$-dimensional data vector for both our EPPSA scheme and schemes in [28, 31, 33]. For the fairness of comparison, these schemes are assumed to get moduli with the same bit length.

In our EPPSA scheme, the modified Montgomery exponentiations (Algorithms 4, 5, and 6) are used to keep the result of exponentiation in the Montgomery domain. That means the aggregation in EA only needs Montgomery multiplications. Let $T_{MM}$ and $T_{OMM}$ be the time cost of a Montgomery multiplication operation and an ordinary modular multiplication operation, respectively. And time cost of a Montgomery exponentiation is denoted by $T_{ME}$. In our proposed EPPSA scheme, benefitting from the modified Montgomery exponentiations, $(m-1)\cdot T_{MM}$ is needed. In [28], the aggregation of each dimension is calculated by $(m-1)\cdot T_{OMM}$. In [31], the aggregation of each dimension is calculated by $(m-1)\cdot T_{OMM}$. In [33], the cost of aggregation is $(m-1)\cdot T_{MM}+T_{ME}$. A comparative summary of computation cost for $m$ SDs aggregation is listed in Table 1.

To evaluate the performance, we execute the experiments on a Laptop with Windows 10 OS, Intel® Core™ i5-700U 2.50 GHz and 16 GB RAM. And we utilize the OpenSSL library (OpenSSL 1.1.1 h) to provide basic cryptographic primitives. For the evaluation of the EPPSA scheme, we set the $n$ to be 512 and 1024 bits in the Paillier Cryptosystem, and the $n^2$ to be 1024 and 2048 bits. As the number of dimensions changes, we get the comparison of aggregation computation costs of 1024 bits in Figure 2(a), and the computation costs of 2048 bits in Figure 2(b). In summary, Figure 2 clearly shows that compared with schemes in [28, 31, 33], EPPSA has the smallest computation cost. For example, compared with [28], the EPPSA scheme gets 62.5% aggregation performance improvement on 1024 bits.

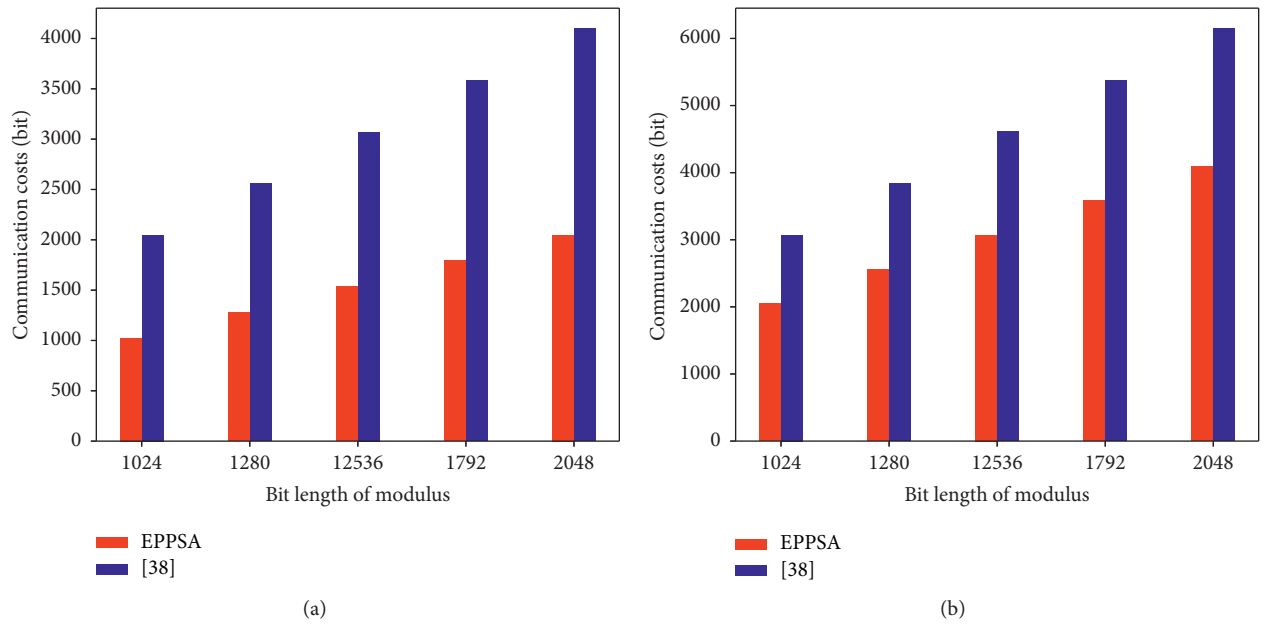


Figure 3: Comparison of communication cost: (a) comparison of communication cost on arithmetic mean and (b) comparison of communication cost on variance.

*6.2. Communication Cost.* Among the previous edge-aided aggregation schemes, the scheme in [36] is the only one that offers statistical functions. Therefore, we compare the communication costs of the EPPSA scheme with the scheme in [36]. We consider the communication costs of arithmetic mean and variance for fairness. For the sake of instruction, we denote the bit length of the modulus by $L$.

In [36], EA needs to transmit the aggregated ciphertext of summation and counter to CC for arithmetic mean, in which the communication cost is $2L$. In our EPPSA scheme, EA needs to send aggregated ciphertext of arithmetic mean to CC, in which the communication cost is $L$. In [36], EA needs to transmit the aggregated ciphertext of summation, quadratic summation, and counter to CC for variance, in which the communication cost is $3L$. In our EPPSA scheme, EA needs to send aggregated ciphertext of arithmetic mean and quadratic mean to CC, in which the communication cost is $2L$. Figure 3 shows the communication cost comparison of EPPSA and [36] in different bit lengths. It can be demonstrated that the communication cost of the EPPSA scheme decreases by 50% on arithmetic mean and 33% on variance.

## 7. Conclusion

In this article, we present an efficient privacy-preserving statistical aggregation scheme for edge computing-enhanced WSNs. The EPPSA scheme adopts the Paillier encryption scheme and ECDSA signature algorithm to guarantee data confidentiality, authentication, and data integrity. Compared with the existing multidimensional and multifunctional data aggregation schemes, the EPPSA scheme improves the efficiency of aggregation and decreases the communication load. Furthermore, the EPPSA scheme improves privacy protection by hiding the total number of devices in the data report. The EPPSA scheme can be applied in various WSN scenarios, such as smart factory, health care, and environmental monitoring.

## Data Availability

The data used in the experiments will be available upon request.

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the Key Research and Development Program of Shandong Province (the Major Scientific and Technological Innovation Project of Shandong Province) under Grant no. 2020CXGC010114.

## References

[1] M. Assim and A. Al-Omary, "Design and Implementation of Smart home Using WSN and IoT Technologies," in *Proceedings of the 2020 International Conference On Innovation And Intelligence For Informatics, Computing And Technologies (3ICT)*, pp. 1–6, Sakheer, Bahrain, December2020.

[2] H. Gao, C. Liu, and Y. Yin, "A hybrid approach to trust node assessment and management for VANETs cooperative data communication: historical interaction perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[3] P. Vijayakumar, M. Azees, V. Chang, J. Deborah, and B. Balusamy, "Computationally efficient privacy preserving authentication and key distribution techniques for vehicular

ad hoc networks," *Cluster Computing*, vol. 20, no. 3, pp. 2439–2450, 2017.

[4] P. Vijayakumar, V. Chang, L. J. Deborah, B. Balusamy, and P. G. Shynu, "Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks," *Future Generation Computer Systems*, vol. 78, pp. 943–955, 2018.

[5] X. Ge, J. Yu, H. Zhang, J. Bai, J. Fan, and N. N. Xiong, "SPPS: a search pattern privacy system for approximate shortest distance query of encrypted graphs in IIoT," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, 2021.

[6] B. Wang, X. Gu, and S. Yan, "STCS: A practical solar radiation based temperature correction scheme in meteorological WSN," *International Journal of Sensor Networks*, vol. 28, no. 1, pp. 22–33, 2018.

[7] A. Zainuri, R. Yuwono, S. R. Arief, and M. Ghadafi, "Performance of temperature and humidity sensors with WSN mesh topology," *Advanced Science Letters*, vol. 25, no. 1, pp. 70–74, 2019.

[8] N. Sivakumar, "Minimizing transmission loss using inspired ant colony optimization and Markov chain Monte Carlo in underwater WSN environment," *Journal of Ocean Engineering and Science*, vol. 4, no. 4, pp. 317–327, 2019.

[9] Y. Yin, Z. Cao, Y. Xu, H. Gao, R. Li, and Z. Mai, "QoS prediction for service recommendation with features learning in mobile edge computing environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1136–1145, 2020.

[10] Y. Huang, H. Xu, H. Gao, R. Li, and Z. Mai, "SSUR: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 670–681, 2021.

[11] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 274, 2019.

[12] X. Ma, H. Xu, H. Gao, and M. Bian, "Real-time Multiple-Workflow Scheduling in Cloud Environments," *IEEE Transactions on Network and Service Management(TNSM)*, vol. 18, 2021.

[13] X. Gao, J. Yu, Y. Chang, H. Wang, and J. Fan, "Checking only when it is necessary: enabling integrity auditing based on the keyword with sensitive information privacy for encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[14] D. Mishra, D. Dharminder, P. Yadav, Y. S. Rao, P. Vijayakumar, and N. Kumar, "A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party," *Journal of Information Security and Applications*, vol. 55, Article ID 102648, 2020.

[15] X. Liu, J. Yu, F. Li, W. Lv, Y. Wang, and X. Cheng, "Data aggregation in wireless sensor networks: from the perspective of security," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6495–6513, 2020.

[16] R. Li, C. Sturtivant, J. Yu, and X. Cheng, "A novel secure and efficient data aggregation scheme for IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1551–1560, 2018.

[17] H. Gao, Y. Zhang, H. Miao, R. J. D. Barroso, and X. Yang, "SDTIOA: Modeling the Timed Privacy Requirements of IoT Service Composition: A User Interaction Perspective for Automatic Transformation from BPEL to Timed Automata," *Mobile Networks and Applications*, vol. 26, 2021.

[18] H. Gao, X. Qin, R. J. D. Barroso, W. Hussain, Y. Xu, and Y. Yin, "Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.

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

[20] H. Bao and L. Chen, "A lightweight privacy-preserving scheme with data integrity for smart grid communications," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1094–1110, 2016.

[21] K. Alharbi and X. Lin, "Lpda: a lightweight privacy-preserving data aggregation scheme for smart grid," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Huangshan, China, October2012.

[22] Z. Sui, M. Niedermeier, and H. de Meer, "RESA: A robust and efficient secure aggregation scheme in smart grids," in *Proceedings of the Proceedings of the 10th International Conference on Critical Information Infrastructures Security*, pp. 171–182, Cham, May2015.

[23] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "EDAT: efficient data aggregation without TTP for privacy-assured smart metering," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, May2016.

[24] Y. Liu, W. Guo, C. I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1767–1774, 2018.

[25] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712–719, 2018.

[26] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[27] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1369–1381, 2017.

[28] P. Zeng, B. Pan, K. K. R. Choo, and H. Liu, "MMDA: multidimensional and multidirectional data aggregation for edge computing-enhanced IoT," *Journal of Systems Architecture*, vol. 106, Article ID 101713, 2020.

[29] X. Liu, Y. Zhang, B. Wang, and H. Wang, "An anonymous data aggregation scheme for smart grid systems," *Security and Communication Networks*, vol. 7, no. 3, pp. 602–610, 2014.

[30] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[31] X. Wang, Y. Liu, and K. K. R. Choo, "fault-tolerant multi-subset aggregation scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4065–4072, 2020.

[32] A. Mohammadali and M. S. Haghighi, "A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid," *IEEE*

*Transactions on Smart Grid*, vol. 12, no. 6, pp. 5212–5220, 2021.

[33] Y. Zhang, J. Chen, H. Zhou, and L. Dang, "A privacy-preserving data aggregation scheme with efficient batch verification in smart grid," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 2, pp. 617–636, 2021.

[34] Y. Guo, N. Wang, Z. Y. Xu, and K. Wu, "The internet of things-based decision support system for information processing in intelligent manufacturing using data mining technology," *Mechanical Systems and Signal Processing*, vol. 142, Article ID 106630, 2020.

[35] P. Zhang, J. Wang, K. Guo, F. Wu, and G. Min, "Multi-functional secure data aggregation schemes for WSNs," *Ad Hoc Networks*, vol. 69, pp. 86–99, 2018.

[36] C. Peng, M. Luo, P. Vijayakumar, D. He, O. Said, and A. Tolba, "Multi-functional and multi-dimensional secure data aggregation schemes in WSNs," *IEEE Internet of Things Journal*, vol. 9, 2021.

[37] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Berlin, Heidelberg, April1999.

[38] *ISO/IEC 18033-6:2019, IT Security Techniques*, Encryption Algorithms Homomorphic Encryption (British Standard), 2019, https://www.iso.org/standard/67740.html.

[39] M. Shah, W. Zhang, H. Hu, and N. Yu, "Paillier cryptosystem based mean value computation for encrypted domain image processing operations," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 3, pp. 1–21, 2019.

[40] S. Gueron, "Efficient software implementations of modular exponentiation," *Journal of Cryptographic Engineering*, vol. 2, no. 1, pp. 31–43, 2012.

[41] R. K. Kodali, "Implementation of ECDSA in WSN," in *Proceedings of the 2013 International Conference On Control Communication And Computing (ICCC)*, pp. 310–314, IEEE, Thiruvananthapuram, India, December2013.