Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4858722, 14 pages
https://doi.org/10.1155/2022/4858722

Research Article

Secure Data Aggregation Aided by Privacy Preserving in Internet of Things

X. Liu, X. Wang, K. Yu, X. Yang, W. Ma, G. Li, and X. Zhao

School of Computer Science, Qufu Normal University, Rizhao 276826, China

Correspondence should be addressed to K. Yu; kanyu@qfnu.edu.cn

Received 17 December 2021; Revised 20 February 2022; Accepted 24 February 2022; Published 17 March 2022

Academic Editor: Zhiguo Qu

With the rapid development of Internet of Things (IoT), more and more wireless smart devices are widely deployed in its typical applications. These devices acquire numerous data, and their security transmission is a challenge issue due to the openness and broadcast nature of wireless network. Security data aggregation (SDA) plays a significant role in IoT which can not only protect the data privacy but also reduce the network traffic among smart devices. In this paper, a multiple application SDA (MASDA) mechanism is proposed and it can ensure the data confidentiality without losing the integrity of data transmission. Firstly, we discuss an improved homomorphic message authentication code (iHMAC) which can verify the integrity of sensing data and eliminate the injection of false data. Secondly, a multiple application elliptic curve cryptography mechanism (iECC) is described. We introduce multiple types of data simultaneous encryption into an elliptic curve encryption system. This encryption mechanism aggregates the encrypted sensing data and prevents the sensing from being tampered and leaked in relay nodes. Finally, MASDA is a multiple application mechanism and provide both the confidentiality and integrity compared with other SDA schemes. It fuses the sensing data of different applications in a single data packet, and the base station can recover them one-by-one. The security analysis and the simulation experiments verify that MASDA has better performances in terms of the security, the computation cost, the communication overhead, and the data accuracy.

1. Introduction

The Internet of Things (IoT) is an intelligent system which can realize the sensing, the communication, and the decision-making through its underlying technology [1]. With the emerging of fifth generation (5G) communication, the reasonable implement of distributed devices has become an important research topic in IoT [2, 3]. The rapid development of sensor techniques, including the microprocessor and wireless communication, provides a feasible solution and promotes the concept of IoT to a more broad field in data communication, data exchange, and data integration [4]. As the driving force of IoT, the wireless sensor networks (WSNs) are widely used in typical IoT application scenarios, such as the industrial manufacturing, the smart transportation, the environmental monitoring, and the smart city [5, 6]. The integration of IoT and WSNs is one of the key schemes to push forward the third wave of the information technology revolution. However, new security threats become a huge obstacle to prevent some novel network techniques from being applied into IoT with WSNs. In an IoT system, sensor nodes of WSNs are usually deployed in open and hostile environments, attackers can more easily compromise the data security through the eavesdropping and the tampering attacks. What is worse, the constrained resources (e.g., the battery energy, the computation capacity, and the communication scope), it is a challenging issue to directly introduce the traditional security mechanisms into WSNs [7–9]. Therefore, security data aggregation (SDA) is proposed to guarantee the security of IoT, eliminate the redundancy of data acquisition, and improve the energy efficiency of smart nodes.

Although data aggregation (DA) is a common technique to reduce the data redundancy, improve the network efficiency, and prolong the lifetime of the network, it is vulnerable to various attacks in WSNs [10, 11]. Therefore, DA algorithms need to be designed together with the security protocol in order to protect the sensing data and improve

the network efficiency as well. The goal of SDA is to provide security for DA and prevent the private data from being eavesdropped on and leaked to unauthorized users in the process of data transmission among resource-limited sensor nodes [12]. As a result, the network efficiency and security can be assured. In the traditional sense, confidentiality and integrity are two important indexes for data transmission. The confidentiality requires that the attacker cannot understand the transmitted data and the integrity means that the receiver can distinguish the tampered data from untampered ones.

The confidentiality of DA can be implemented through two different mechanisms, the hop-by-hop encryption and the end-to-end encryption. The hop-by-hop DA mechanism encrypts the sensing data before sending them to the next hop and the ciphertexts are decrypted in the next hop in order to aggregate them with the sensing data of other sensor nodes. However, the hop-by-hop encryption leads to the nonignorable network delay. Moreover, the attacker may obtain the decrypted in the mediation node. The end-to-end DA mechanism directly aggregates the sensing data in the ciphertext, and the encryption is not necessary for relay nodes. In recent years, many SDA algorithms have been proposed to protect the data privacy of WSNs [13–17]. These mechanisms provide end-to-end data confidentiality which can improve data privacy and reduce the aggregation delay. However, most of these contributions focus on the confidentiality of data and ignore the integrity of DA. In addition, these mechanisms are difficult to work in heterogeneous WSNs with different sensor nodes (e.g., the temperature sensor, the humidity sensor, and the light sensor) in a practical multiple application sensor network [18, 19] and a malicious node can eavesdrop the keys and tamper the private data. Therefore, it is important to design a multiple application SDA mechanism with the aim of protecting confidentiality and integrity simultaneously.

Three major challenges need to be conquered in designing an SDA mechanism in multiple application scenarios. (i) In an IoT system with WSNs, sensor nodes are usually deployed in unattended even hostile environments, and each sensor node is equipped with an unchargeable battery and low computation processor. This implies that an SDA scheme does not play a huge burden on sensor resources. (ii) The security of data collection includes both confidentiality and integrity, which needs to design a novel encryption mechanism. (iii) Traditional encryption algorithms have high complexity and are difficult to support simultaneous aggregation of multiple application data (such as temperature and humidity).

According to the aforementioned issues, we propose a multiapplication secure data aggregation (MASDA) mechanism based on the elliptic curve encryption of compound-order finite groups, which can protect the data aggregation, reduce the energy consumption, and provide the simultaneous DA for multiple applications. The contributions are summarized as follows:

(i) A multiapplication SDA mechanism is proposed. It can provide confidentiality and integrity in the data transmission of IoT with WSNs

(ii) An elliptic curve encryption scheme based on compound-order finite groups is discussed, which can encrypt different application data which can be aggregated at the cluster head in the ciphertext. The communication overhead and computation complexity are reduced

(iii) A novel homomorphic message authentication is designed. It can verify the data integrity of sensing data and the injection of false data can be eliminated

The rest of this paper is organized as follows. In Section 2, we discuss the related work of data aggregation scheme. In Section 3, we describe the network model and attacks. In Section 4, we present the detailed descriptions of the proposed mechanism MASDA. In Section 5, we provide security analysis. After providing simulation experiment and analysis in Section 6, we give our conclusion of this paper and future work in Section 7.

2. Related Works

As the critical technique of IoT, SDA has received great attention in recent years. In this section, we classify the SDA into three categories, namely, the confidentiality mechanism, the integrity mechanism, and both confidentiality and integrity mechanisms.

2.1. Confidentiality. In [20], Alghamdi et al. investigated the reliable and secure end-to-end DA issue. Two data aggregation approaches were discussed, and the selective forwarding attack and the modification attack were taken into consideration in a homogeneous cluster-based WSN. The authors suggested that the secret sharing and signature had the potential ability to aggregate the data without understanding the contents of messages. Meanwhile, the base station can verify the aggregation result and retrieve the raw data from the aggregated data. However, these approaches can be adopted in a homogeneous network with single application data. They focus on the confidentiality of the network, and the data integrity is left to be investigated in the future. In [21], Zhong et al. proposed the latest privacy homomorphism scheme based on the elliptic curve encryption, which filtered out the false data and avoided unnecessary energy consumption. It ensured that the base station was able to verify the received data and recovery the original sensing data, which provided the arbitrary aggregation in a WSN. In [22], Gope and Sikdar discussed a lightweight and privacy-friendly shield-based spatial data aggregation mechanism which depended on the lightweight encryption to protect the data privacy, such as hash functions and the exclusive OR operations. Compared with other methods, the proposed scheme significantly reduced the computation overhead. In [23], Fang et al. proposed a data aggregation approach, called CSDA, based on the cluster privacy preserving. In CSDA, the slice-assemble technique was designed and the number of pieces was determined according to the network size so as to improve the aggregation flexibility and data privacy.

These reviewed protocols focus on the confidentiality of data aggregation and protect data from being leaked to unauthorized nodes. However, they ignore the data integrity in data aggregation. Therefore, the aggregation data are prone to be tampered with by malicious nodes, and the cluster head or the base station may not collect the accurate data.

2.2. Integrity. In [24], Arazi proposed a complete and high compact MAC scheme based on the stream encryption. This contribution is to implement the hash conversion based on a stream cipher, in which the intensity of the hash is closely related to the underlying security of the password. In [25], Shen et al. combined the advantages of the aggregation signature with the ID-based cryptography and proposed an ID-based aggregation signature (IBAS) mechanism. IBAS consisted of six probabilistic polynomial time (PPT) algorithms: the setup algorithm, the key generation algorithm, the signing algorithm, the verification algorithm, the aggregation algorithm, and the aggregation verification algorithm. It also deduced the security of the Diffie-Hellman hypothesis based on the random oracle model, which proved that IBAS was able to ensure the integrity of data and reduce communication and storage costs.

These data aggregation protocols provide the integrity protection of private data and achieve the end-to-end security of DA. However, some issues are left to be further studied, such as low confidentiality and the excessive consumption of resources.

2.3. Both Confidentiality and Integrity. Boudia et al. [26] developed a secure aggregation scheme based on the stateful public key cryptography (SASPKC). SASPKC depended on the symmetric homomorphic encryption and the message authentication code to aggregate the ciphertext and generate the signature of aggregation data, respectively. It provided not only the end-to-end data confidentiality but also the data integrity. Shim and Park [27] have worked on a secure data aggregation protocol (Sen-SDA) for heterogeneous networks. Sen-SDA employed the additive homomorphic encryption to reduce the length of ciphertext and improve the end-to-end security. In order to provide hop-by-hop authentication, the pairless identity-based signature (IBS) technique and the binary fast search (BQS) were discussed to filter out the false data in a heterogeneous WSN. As a result, Sen-SDA can protect the private data and improve the efficiency of multisignature verification. Shim and Park [28] proposed a t times lattice-based homomorphic cryptosystem based on random soft noise techniques for smart grids. After the sensing data were aggregated, the smart meter (SM) generated a complete aggregation ciphertext and the trusted authority (TA) issued a time stamp and a new random number. These components meet the security requirements of integrity, confidentiality, and authenticity, which can defend the replay attack, the forwarding attack, and the quantum attack.

Although the existing protocols have discussed the security of DA, a few contributions have paid their attentions to support the integrity and the confidentiality simultaneously. Moreover, the current solutions either aggregate single application data or limit it to a certain type of data query. These facts inspire this study. We consider both the data confidentiality and the data integrity and combine the elliptic curve encryption with the homomorphic message authentication to concurrently aggregate the multiple sensing data, which can expand the application scenarios of SDA.

3. Network Model and Attacks

In this section, we present the network model and the attacks in an IoT with WSNs. The symbols used in the MASDA protocol are shown in Table 1.

3.1. Network Model. In this paper, we deploy a cluster network topology to aggregate the sensing data in a WSN. Three types of nodes are involved, the base stations (BS), the cluster head (CH), and the cluster member (CM). The sensor nodes are capable of sensing, calculating, and transmitting the collection data. A network is divided into several application groups according to the different functions of nodes, such as the temperature sensor group, the humidity sensor group, and the light sensor group. Nodes of different groups are randomly scattered in the network. CH receives the data from its CMs and aggregates them based on the application types. The aggregation results are eventually transmitted to BS. Figure 1 depicts the cluster model of data aggregation.

3.2. Attack Types. We take various attacks in IoT into consideration including the passive attack and the active attack. The specific attacks are described in Table 2.

3.2.1. Passive Attack. The passive attack usually listens to the communication channel and eavesdrops on the transmission data, which leads to the leaking of confidentiality even infers the key of the cryptosystem. The possible passive attack includes the ciphertext-only attack, the known-plaintext attack, and the selected plaintext attack. In the traditional sense, an encryption mechanism is an effective measure to protect the sensitive data from being attacked by the passive attack in an IoT with WSNs.

3.2.2. Active Attack. Active attack attempts to compromise the network security by changing the data in a network. It may tamper with the sensing data, replay, or discard a whole packet. An IoT may be damaged by one or more active attacks, such as the replay attack, the malleable attack, the unauthorized aggregation, and the forged attack. These negative impacts of these attacks can be degraded through verifying the identity of nodes and checking the integrity of the aggregated data.

4. MASDA Algorithm

In this section, we firstly introduce the idea of MASDA. Then, the elliptic curve encryption algorithm and the homomorphic message authentication are discussed. Finally, a numerical example is provided.

Table 1: The symbols used in MASDA.

Symbol	Description
CH	The cluster head
CM	The cluster member
BS	The base station
n	The order of a point
r	A random number
E	The set of elliptic curve points
C	The ciphertext
G	Pseudo random number generator
F	Pseudo random function
R	Message receiver
A	Message aggregator
F_p	Specified prime field
mac	Message authentication code
P_{pub}	The public key
P_{pri}	The private key

4.1. Idea of MASDA. Based on the cluster topology in Figure 1, CH is elected according to the residual energy and the distance between the sensor node and BS. Firstly, each CM encrypts its sensing data using the elliptic curve encryption algorithm. The different private keys are used for different application data. Secondly, CM generates a message authentication code using the shared key with CH after the data are encrypted. Then, CM sends the ciphertext and the message authentication code to CH. After CH receives the message from the CM, it verifies whether the message authentication code of ciphertext is valid. Finally, CH directly aggregates the ciphertext and transmits the aggregation data to BS which decrypts the aggregated ciphertext and recovers the raw data of different applications.

4.2. Elliptic Curve Encryption Algorithm. We adopt the privacy homomorphic encryption algorithm based on the elliptic curve encryption algorithm [29] in MASDA. The security of homomorphic encryption depends on the subgroup decision problem. In other words, assumed that an element belongs to the compound sequence group, $n = q_1 q_2$, it is infeasible to judge whether it belongs to the subgroup, q_1. This allows a hidden aggregation only relying on the ciphertexts encrypted with different keys. The algorithm supports the additive homomorphism and the multiplication homomorphism, and we will describe the additive homomorphic encryption in order to depict the design of MASDA.

4.2.1. Key Generation. The message receiver, R, generates a tuple, (q_1, q_2, E, n), according to the security parameter, $\tau \in Z$. E is a collection of elliptic curve points which forms a cyclic group. n is the order of E ($n = q_1 q_2$). Two points of order n are randomly selected (u and g) from E. Let $h = u^{q_2}$ and the order of h is q_1. The public key $P_{\text{pub}} = (n, E, g, h)$ is sent to the receiver and the private key $P_{\text{pri}} = q_1$.

4.2.2. Encryption. The sender, S, chooses an integer M ($M < q_2$). The length of M is close to the length of q_2. The value of data m should be less than or equal to M. After S receives the public key, it generates a random number $r(r \in \{0, 1, 2, \cdots, n-1\})$ and obtains the ciphertext $C = g^m + h^r$. g^m and h^r are the scalar multiplication of elliptic curve points, and the plus represents the addition of elliptic curve points.

4.2.3. Data Aggregation. Two ciphertexts received by aggregator A are expressed as follows:

$$C_1 = g^{m_1} + h^{r_1}, \tag{1}$$

$$C_2 = g^{m_2} + h^{r_2}. \tag{2}$$

A fuses these ciphertexts according to Equation (3). Then, it sends ciphertext C' to the receiver R.

$$C' = C_1 + C_2 = g^{(m_1+m_2)} + h^{(r_1+r_2)}. \tag{3}$$

4.2.4. Decryption. After receiving the data from A, R decrypts them using the private key, P_{pri}, according to $C'q_1 = g^{m_3 q_1} + h^{r_3 q_1}$ where $m_3 = m_1 + m_2$ and $r_3 = r_1 + r_2$. Equation (3) is transformed to

$$C'q_1 = g^{m_2 q_1} = (g^{q_1})^{m_3}. \tag{4}$$

R recovers the original data m_3 according to

$$m_3 = \log_{g^{q_1}} \left(C'q_1 \right). \tag{5}$$

Because the length of plaintext is less than or equal to M, the time complexity of decryption is $O(\sqrt{M})$ according to reference [30].

4.3. Improved Homomorphic Message Authentication Code. In [31], Kamal et al. proposed a homomorphic message authentication code (HMAC), which satisfied the homomorphic property. The message authentication code can not be calculated even if the attacker knows the original data. In order to meet the demands of IoT with WSNs, we improve the property of HMAC, called iHMAC, which includes three polynomial-time algorithms, the signature algorithm, the aggregation algorithm, and the verification algorithm. In iHMAC, BS constructs a pseudorandom number generator, $G : K_G \longrightarrow F_p$, and a pseudorandom function, $F : K_F \times id_i \longrightarrow F_p$, where F_p is the specified prime number field. The key pair is used to generate MAC, (k_{mac1}, k_{mac2}), where ($k_{\text{mac1}} \in K_G$) and ($k_{\text{mac2}} \in K_F$).

The ith CM calculate MAC according to its original data m as

$$v = G(k_{\text{mac1}}) \in F_p, \tag{6a}$$

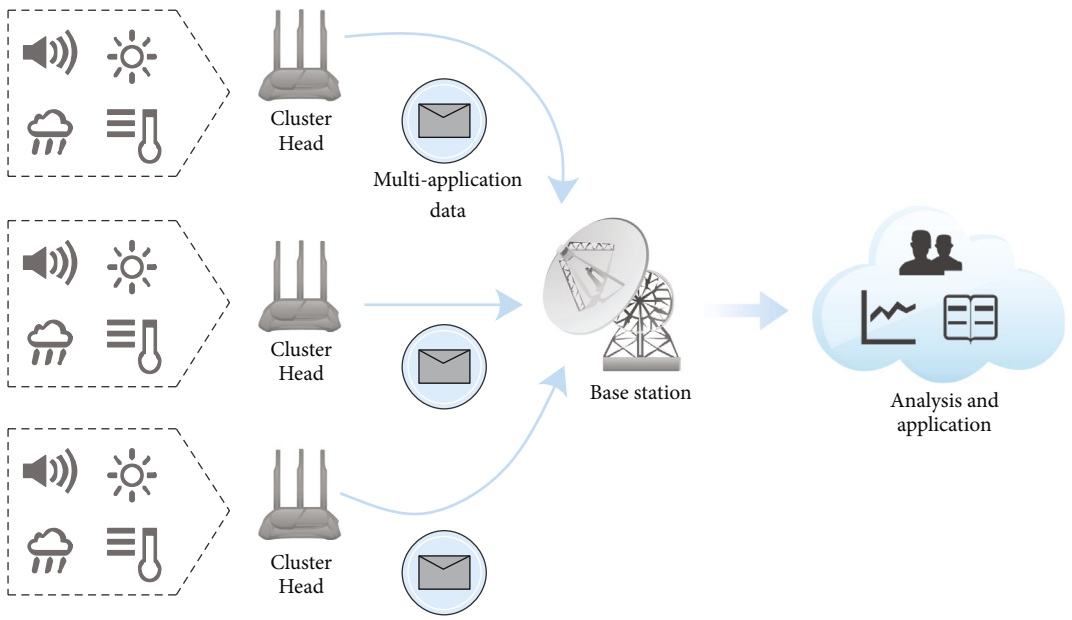

Figure 1: The cluster model.

Table 2: Attack type.

Attack name	Characteristic	Influence
Ciphertext-only attack	Malicious nodes try to obtain information by analyzing ciphertext.	Confidentiality
Known-plaintext attack	The attacker analyzes the known and the corresponding ciphertext to determine the secret information.	Confidentiality
Chosen plaintext attack	The attacker analyzes selected plaintext and its corresponding ciphertext.	Confidentiality
Replay attack	Repeated session requests cause valid packets to be transmitted repeatedly later.	Integrity
Malleability attack	Malicious nodes send grammatically correct meaningless ciphertext to deplete sensor nodes.	Slow down the network.
Unauthorized aggregation	Malicious nodes must know the encryption key and data authentication key to realize unauthorized aggregation.	Integrity
Forgery packet	The attacker forges the packet without knowing the encryption key.	Increase network energy consumption.

$$y = F(k_{mac2}, i) \in F_p, \tag{6b}$$

$$\text{mac} = v \cdot m + y \in F_p. \tag{6c}$$

The aggregation node receives the authentication codes of j nodes and the aggregation result of these codes is ma $c_{agg} = \sum_{i=1}^{j} \text{mac}_i$.

BS generates a verification code, mac_{BS}, according to the collected aggregation data, m_{agg}, and the keys, (k_{mac1}, k_{mac2}), as

$$v_{BS} = G(k_{mac1}) \in F_p, \tag{7a}$$

$$y_{BS} = \sum_{i=1}^{j} F(k_{mac2}, i) \in F_p, \tag{7b}$$

$$\text{mac}_{BS} = v_{BS} \cdot m_{agg} + y_{BS} \in F_p. \tag{7c}$$

It proves that the data are integral if $\text{mac}_{agg} = \text{mac}_{BS}$. Otherwise, the data are trend to be tampered.

4.4. Confidentiality and Integrity Mechanism of MASDA. MASDA is designed to be worked in multiple application scenarios, and the traditional elliptic curve encryption needs to be improved so that it can ensure the confidentiality as well as the integrity.

4.4.1. Key Generation. BS generates a tuple, $(q_1, q_2, \cdots, q_{k+1}, E, n)$, according to the security parameter $\tau \in Z$. E is a collection of elliptic curve points which form a cyclic group. n is the order of E and satisfied that $n = q_1 q_2 \cdots q_{k+1}$. $k + 2$ points of order n are randomly selected, $(u_1, u_2, \cdots, u_{k+1}, g)$ from E. The value of h is determined according to

$$h = u_{k+1}^{\beta}, \tag{8}$$

where the order of h is q_{k+1} and β is

$$\beta = \prod_{i=1}^{k} q_i (i = 1, 2, \cdots, k). \tag{9}$$

Then, the k keys are allocated to k application group nodes. P_z is the key of the zth$(1 \le z \le k)$ application group node.

$$P_z = u_z^{\alpha}, \tag{10}$$

where α is

$$\alpha = \prod_{i=1}^{k+1} q_i (z = 1, 2, \cdots, k). \tag{11}$$

The public key deployed for the zth application group node is $P_{\text{pubz}}(n, E, g, h, P_z)$, and the private key group, $P_{\text{pri}} = (q_1, q_2, \cdots, q_{k+1})$, is retained by BS. In the integrity verification, BS uses the keys, $(k_{\text{mac1}}, k_{\text{mac2}})$, to generate MAC in a sensor node in advance.

4.4.2. Encryption. In order to ensure that the length of T_z is close to the length of q_z, T_z should satisfy with $T_z < q_z$ and the data, m, should be less than or equal to T_z. After a CM of the zth application group receives the public key, it creates a random number, $r(r \in \{0, 1, 2, \cdots, n-1\})$ and encrypts m with the public key to ciphertext, C, according to

$$C = P_{\text{pubz}}^{m} + h^{r}. \tag{12}$$

The CM with identity i also calculates a verification code depending on the ciphertext C using the following equation; CM sends data packects to the aggregation node (CH).

$$v = G(k_{\text{mac1}}) \in F_p, \tag{13a}$$

$$y = F(k_{\text{mac2}}, i) \in F_p, \tag{13b}$$

$$\text{mac} = v \cdot C + y \in F_p. \tag{13c}$$

4.4.3. Aggregation. After CH receives data from j nodes in a cluster, it first verifies whether the data are integral according to their authentication codes. CH obtains a new MAC, mac_{CH}, corresponding to the data of every CM using Equation (13) and compares mac_{CH} with the authentication code sent by a node. If the verification successes, the data of nodes will be aggregated. Otherwise, the data will be discarded.

The encrypted data and MAC are separately aggregated to produce the aggregation ciphertext, C_{agg}, and the aggregation MAC, mac_{agg}, as shown in

$$C_{\text{agg}} = \sum_{l=1}^{j} C_l = \sum_{l=1}^{j} P^{\sum_{l=1}^{j} m_l} + h^{\sum_{l=1}^{j} r_l}, \tag{14}$$

$$\text{mac}_{\text{agg}} = \sum_{l=1}^{j} \text{mac}_l, \tag{15}$$

where m_l represents the original data of the lth node, r_l denotes the random number of the lth node, and mac_l is the MAC of the lth node. The aggregation ciphertext and message code, $C_{\text{agg}} \mid \text{mac}_{\text{agg}}$, will be transmitted to BS for decryption and verification in the next step.

4.4.4. Decryption and Verification. After receiving $C_{\text{agg}} \mid \text{mac}_{\text{agg}}$, BS firstly verifies the integrity of the data.

$$v_{\text{BS}} = G(k_{\text{mac1}}) \in F_p, \tag{16a}$$

$$y_{\text{BS}} = \sum_{i=1}^{j} F(k_{\text{mac2}}, i) \in F_p, \tag{16b}$$

$$\text{mac}_{\text{BS}} = v_{\text{BS}} \cdot C_{\text{agg}} + y_{\text{BS}} \in F_p. \tag{16c}$$

Then, BS compares mac_{BS} with mac_{agg}. If $\text{mac}_{\text{agg}} = \text{mac}_{\text{BS}}$, BS decrypts the ciphertext of aggregation data; otherwise, BS regards the date as a tampered or incomplete one.

After the data is authenticated, BS encrypts the ciphertext using the private key P_{pri}. For different data of k applications, different private keys are used to decrypt them. If BS wants to recover the data of the zth application, they are decrypted according to

$$C_{\text{agg}}\gamma = \left(\sum_{l=1}^{j} P^{\sum_{l=1}^{j} m_l} + h^{\sum_{l=1}^{j} r_l} \right) \gamma, \tag{17}$$

where

$$\gamma = \prod_{d=1, d \neq z}^{k+1} q_d. \tag{18}$$

Then, Equation (17) is transformed to $C_{\text{agg}}^{\gamma} = (\sum_{l=1}^{j} m_l)\gamma$. Let $\sum_{l=1}^{j} p^{\sum_{l=1}^{j} m_l} = \widehat{P}$, the aggregation result m_{agg}^{z} of the zth application is $m_{\text{agg}}^{z} = \log_{\widehat{P}\gamma} C_{\text{agg}}\gamma$. BS can recover the aggregated information of all application groups according to the abovementioned steps, and the goals of secure data transmission and integrity verification are achieved.

4.5. A Numerical Example of MASDA. In this section, we will use a numerical example to illustrate the working mechanism of MASDA. Suppose that two clusters $((\text{Cluster}_1$ and $\text{Cluster}_2))$ for two applications are deployed. The public key of node_t is (n, E, g, h, P_t) and the public key of node_h is (n, E, g, h, P_h). The aggregation nodes in Cluster_1 and

Cluster_2 are DA_1 and DA_2, respectively. Cluster_1 is farther from BS compared with Cluster_2. There are two different types of nodes in each cluster, the temperature sensor node node_t and the humidity sensor node node_h.

4.5.1. Key Generation. In key generation step, the orders of P_t, P_h, and h are 11, 13 and 17, then $n = 11 \times 23 \times 17 = 2431$.

4.5.2. Encryption. Two nodes in Cluster_1 are encrypted according to the following method, where the random number is randomly generated by the nodes. $m_{t1} = 1$ is the sensing data collected by node_{t1} with the random number 4. The data are encrypted using Equation (19) and the message verification code MAC_{t1} is generated based on C_{t1}.

$$C_{t1} = P_t^1 + h^4. \tag{19}$$

The sensing data collected by node_{h1} is $m_{h1} = 3$ with the random number 6. The ciphertext is C_{h1} according to Equation (20), and the message verification code MAC_{h1} is calculated based on C_h.

$$C_{h1} = P_h^3 + h^6. \tag{20}$$

4.5.3. Aggregation. When DA_1 receives the data from two nodes, it verifies the MAC and aggregates the data to C_{DA1} according to Equation (21). The message verification codes are also fused to MAC_{agg1} according to Equation (22).

$$C_{\text{DA1}} = C_{t1} + C_{h1} = P_t^1 + P_h^3 + h^{10}, \tag{21}$$

$$\text{MAC}_{\text{agg1}} = \text{MAC}_{t1} + \text{MAC}_{h1}. \tag{22}$$

Then, DA_1 sends the aggregation data C_{DA1} to DA_2. In Cluster_2, two nodes ($M_{t2} = 4$, $M_{h2} = 2$) receive the aggregation data and the random numbers are 2 and 7 in these nodes, respectively. After the same encryption process is executed as Cluster_1, the aggregation data C_{agg} are obtained using Equation (23) in DA_2 and the message verification code of aggregation data is $\text{MAC}_{\text{agg}} = \text{MAC}_{\text{agg1}} + \text{MAC}_{\text{agg2}}$.

$$C_{\text{agg}} = C_{\text{DA1}} + C_{\text{DA2}} = P_t^5 + P_h^5 + h^{19}. \tag{23}$$

Because the order of h is 17, $h^{17} = \infty$, where ∞ is the generator in elliptic curve encryption. Therefore, Equation (23) can be rewritten as $C_{\text{agg}} = P_t^5 + P_h^5 + h^2$. Finally, Cluste r_2 sends C_{agg} and MAC_{agg} to BS.

4.5.4. Decryption and Verification. After BS receives C_{agg} and MAC_{agg}, it verifies the MAC and decrypts the data if the verification is successful. The private key of temperature data is $P_t = 17 \times 13 = 221$, and BS can decrypt the temperature data m_h according to

$$M_h = log_{P_t} C_{agg}^{221} = log_{P_t} \left(P_t^5 + P_h^5 + h^2\right)^{221} = 5, \tag{24}$$

where $P_h^{5\times221} = P_h^{1105} = \infty, h^{2\times221} = h^{442} = \infty$. Similarly, BS can also use the private key of humidity data to recover the original humidity data.

5. Security Analysis

In this section, we will show the resistance of MASDA against the passive and the active attacks.

5.1. Ability to Resist Passive Attacks. The passive attacks include the ciphertext analysis, the known-plaintext attack, and the selected plaintext attacks. The elliptic curve encryption of MASDA relies on the factorization of large integers, and it is robust to the ciphertext analysis. For the known-plaintext attack and the selected plaintext attack, the encryption encryption of MASDA is related to the random number and the ciphertext is probabilistic. Therefore, MASDA can defend against the known plaintext attack and the selected plaintext attack.The above attack methods are all for the analysis of ciphertext and plaintext. According to the characteristics of MASDA plaintext and ciphertext, we can draw the following conclusion: MASDA achieves end-to-end confidentiality having indistinguishable ciphertext in the presence of a probabilistic polynomial time adversary.

Since random numbers are added during encryption, MASDA generating different ciphertexts for the same plaintext. The attackers cannot deduce the plaintext from the eavesdropped ciphertext. By comparing the ciphertext, the attackers also cannot deduce any important information. Furthermore, the ciphertext is secured by appending the MAC. Even if the attackers breaks the signature and gets the ciphertext, they cannot decrypt it because the key is in the BS. Due to we assume the BS is a powerful and trusted device, the attackers cannot obtain the secret to decrypt the ciphertext. Therefore, MASDA has the ability to resist plaintext and ciphertext analysis.

In addition, we analyze the brute force cracking of the key. We compare the scheme with two other asymmetric homomorphic encryption mechanisms based on elliptic curve encryption mechanism: EC-OU [32] and EC-EG [33]. The security of EC-OU is based on the intractability of factoring. EC-EG security is based on the elliptic curve discrete logarithm problem. In elliptic curve encryption algorithm, the security of encryption mechanism mainly depends on the protection of key by users. The more bits of the key, the harder it is to crack the ciphertext. In MASDA, the length of the private key is $k \mid q \mid$, where k is the number of application types and $\mid q \mid$ is the length of the key in iECC. Table 3 compares the key strength of MASDA with EC-OU and EC-EG. As shown in the table, the key strength of EC-OU and EC-EG depends on the length of the key. However, the key strength of MASDA depends on the number of applications and the length of the key. As the number of applications increases, the key strength of MASDA also increases. The key strength of EC-OU and EC-EG does not change with the number of applications.

5.2. Ability to Resist Active Attacks. The active attacks mainly include the replay attacks, the malleability, and the

Table 3: EC-OU, EC-EG, and MASDA key strength comparison.

Scheme	Key strength (bits)
EC-OU	$\|q\|$
EC-EG	$\|q\|$
MASDA	$k\|q\|$

unauthorized aggregation. These active attacks can compromise data integrity or generate false data. MASDA adopts iHMAC to defend against these attacks. The node uses the pseudorandom number key, and the sent data to generate the MAC and sends the result to the aggregation nodes. After receiving the MAC, the aggregation nodes perform an aggregation operation on the MAC and send the aggregated MAC to the BS. The BS checks whether the data is under active attack by calculating the MAC. Data integrity and false data screening are ensured if the calculated MAC is the same as the received MAC. Therefore, it is difficult for an attacker to launch an attack for the following reasons:

(i) It is difficult to generate a MAC unless the key is known

(ii) The keys generated by each node are different

(iii) The key is not shared with other nodes

We will analyze in detail MASDA's defense against active attacks.

5.2.1. Replay Attacks. Replay attacks destroy data freshness. Data freshness is used to measure whether the collected data is recent. If the collected data is within the query time period, the result is recent. Data freshness is violated if an adversary submits legitimate data to a queryer before or after the query time period. MASDA can not be resistant to replay attacks. However, in situations where data freshness is critical, there are two methods that can be used as an additional means of protection.

(i) Node adds timestamp to packet. BS asks the nodes to collect data for a time period t. The sending node records the time of sending the data and adds it to the data packet as a timestamp. After BS receives the data packet, it first checks the timestamp. If the timestamp on the packet is within t, BS will receive the packet. Otherwise, regardless of whether the data is correct, the BS will not receive this data packet

(ii) BS uses different keys in different time periods. In this way, BS sends a new key for generating the MAC to the node at regular intervals. If an attacker tries to interfere with the data collection process by exploiting old data data_{old}. The attacker will send the data_{old} into WSNs. When the data_{old} arrives at the aggregation node or BS, the aggregation node uses the new key to verify whether the MAC in the data_{old} is correct. If it is not correct, the data packet will be discarded

Table 4: The parameters of encryption.

Parameter	Value
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{amp}	0.0013 pJ/bit/m^4
E_i	0.5 J

5.2.2. Malleability Attack. Malleability is an undesirable property of cryptosystems that negatively affects the data integrity of ciphertexts. In this attack, one ciphertext is modified into another so that decryption produces the associated plaintext. We will show that MASDA has the ability to resist such attacks.

We assume that the attacker actively obtains the information in the wireless signal and attemps to make the BS decrypt a different plaintext by modifying certain bytes in the data. When the modified data arrives at BS, BS generates a MAC $\text{mac}_{\text{modify}}$ using the modified data. Then, BS compares the $\text{mac}_{\text{modify}}$ with the original $\text{mac}_{\text{original}}$. Since iHMAC adopts linear calculation, different inputs will lead to different outputs. Therefore, $\text{mac}_{\text{modify}}$ is different from $\text{mac}_{\text{original}}$. Then, BS can verify whether the data has been modified by comparing the $\text{mac}_{\text{modify}}$ with the $\text{mac}_{\text{original}}$. The attackers can successfully change the ciphertext if and only if they can forge a valid MAC for the ciphertext. According to HMAC [31], this is a difficult work for an attacker.

5.2.3. Unauthorized Aggregation. Unauthorized aggregation is a particular weakness of homomorphic encryption schemes. The main idea of this attack is to collect multiple correct ciphertexts into a single forged but valid ciphertext to deceive the BS. If the CH can only perform data aggregation, anyone can mislead the BS by dropping some packets to forge a wrong aggregation result.

In MASDA, each CH not only performs the aggregation operation but also generates a signature on the aggregation result. Thus, the BS can check the authenticity and integrity of the aggregated data sent by CHs. For the unauthorized aggregation, the adversary must destroy at least one sensor node before it can obtain the keys of iECC and iHMAC. Our encryption mechanism is built on the asymmetric cryptosystems of elliptic curve encryption. Since it is very difficult and infeasible to know the details of the curve, attackers are unable to perform unauthorized aggregation.

5.2.4. Forged Packet Attack. Forged packet attacks disrupt the data aggregation process by injecting fake data into the network, causing the results to deviate from the true value, making such attacks impossible for the base station to determine. As a result, network resources are wasted and become useless as a result. To detect spurious data injection or forged packets, MASDA appends a MAC to each ciphertext and aggregated ciphertext. MAC allows each sensor node to ensure the origin of the messages it generates. In MASDA, iECC is used for data encryption, and then, iHMAC is used

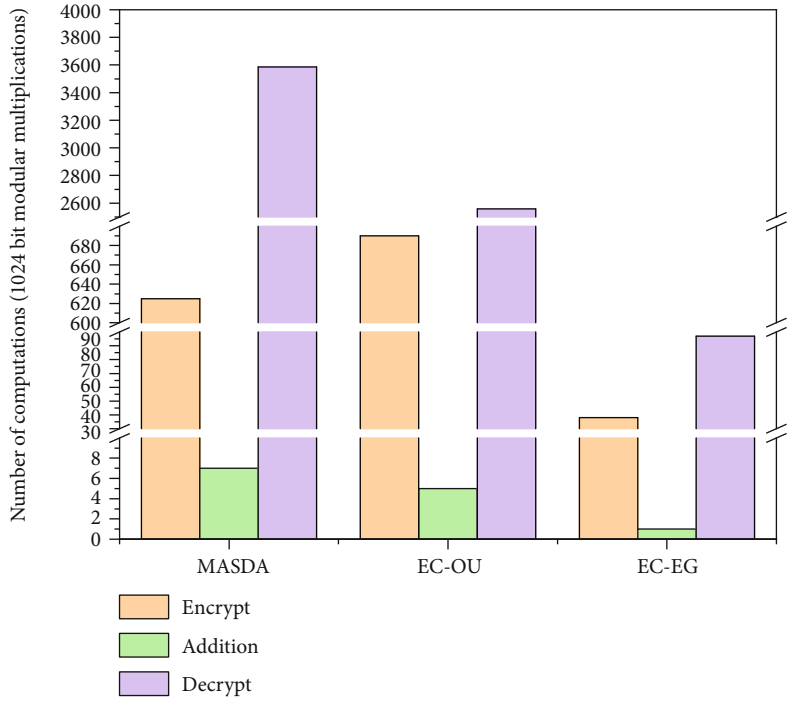

Figure 2: Comparison of computation overhead.

to prevent false data injection. After receiving the data, the aggregation node first verifies whether the data is reliable. If the verification is successful, the aggregation ciphertext and the aggregation MAC value are sent to BS, and the responsible node discards the ciphertext. After receiving the message, the BS first checks the integrity and decrypts it if it passes the verification. Otherwise, BS considers that the information has been tampered with or incomplete. Therefore, the scheme can effectively resist forged packet attacks.

6. Simulation Experiment and Analysis

In this section, we provide the experimental simulation of MASDA. We employ the OMNET++ as the simulation platform and the parameters are shown in Table 4. Four types of nodes are deployed in a WSN for four applications. MASDA are evaluated in terms of the computation overhead, the communication cost, and the aggregation accuracy.

6.1. Computation Overhead. The homomorphic encryption mechanism based on symmetric key encryption consumes fewer resources. However, the mechanism based on asymmetric homomorphic encryption is more secure. EC-OU, EC-EG, and MASDA all include the encryption, the aggregation, and the decryption operations. However, they are based on different mathematical foundations. we follow the evaluation method defined in [27], first calculate the number of $|q|$-bit modular multiplications and then convert different calculation operations into basic unit numbers (1024-bit modular multiplication) and regard it as the evaluation standard. Their computation overheads are shown in Figure 2.

Figure 2 shows that EC-EG is the best one compared with EC-OU and MASDA in computation overhead because of the smaller modulus. Noticed that it is at the price of lower encryption strength. EC-OU and MASDA choose the same modulus, so their computation overheads are the same level. However, neither EC-EG nor EC-OU can support the data aggregation of different applications, while MASDA can be applied to multiple application scenarios. In terms of decryption, MASDA needs more computation overhead than EC-OU and EC-EG. Generally speaking, BS has the unlimited resources and the more computation overhead of MASDA in decryption does not pay a huge burden on the overall lifetime of WSNs.

6.2. Communication Cost. The communication cost in WSNs is closely related to the length of the ciphertext. For MASDA, the different applications of sensor nodes will affect the length of the ciphertext. The length of ciphertext is $(k+1) \times |q|$ bits in MASDA, and k is the number of application types in a node. Therefore, the length of ciphertext increases with the increase of application types. In [34] the length of ciphertext in EC-OU is $3 \times |q| + 2$ ($|q| = 341$ bits), and that of EC-EG is $2 \times |q| + 2$ ($|q| = 163$ bits). We choose $|q| = 163$ bits and $|q| = 341$ bits as the modules of MASDA and the length of EC-OU, EC-EG, and MASDA are shown in Figure 3.

Figure 3 shows that the length of ciphertext in EC-EG is the smallest one. With the increasing of k, the length of ciphertext in MASDA also increases. Meanwhile, the ciphertext length of MASDA is smaller gradually than EC-OU when $|q| = 163$ bits and $k < 5$ of MASDA. If the applications of nodes are greater than two in the condition of $|q| = 341$ bits, the ciphertext of MASDA is longer than

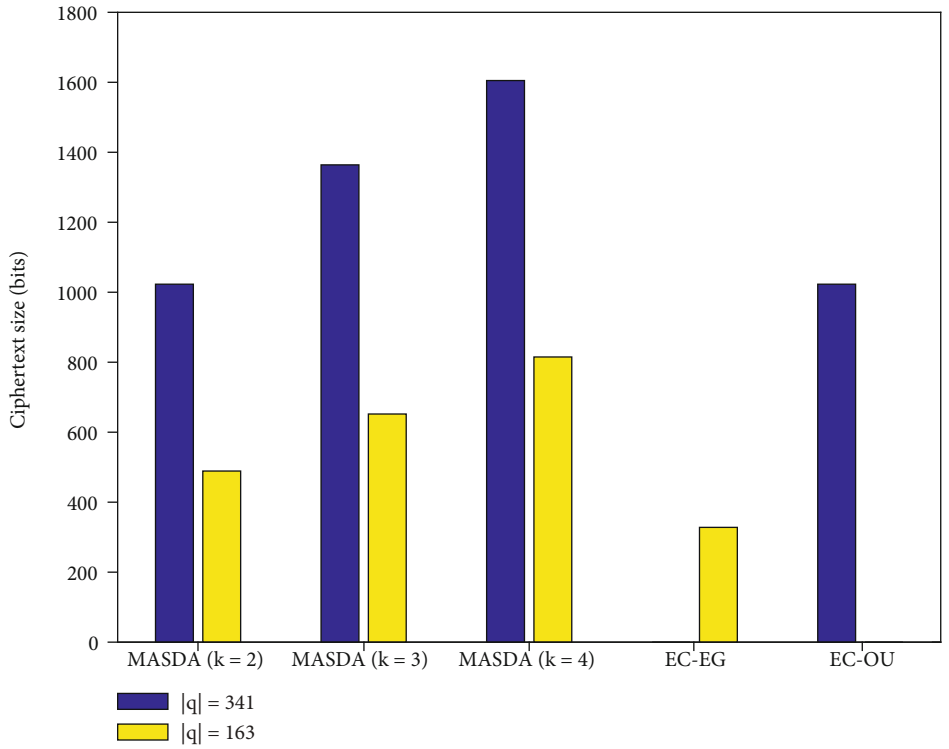

Figure 3: Comparison of ciphertext.

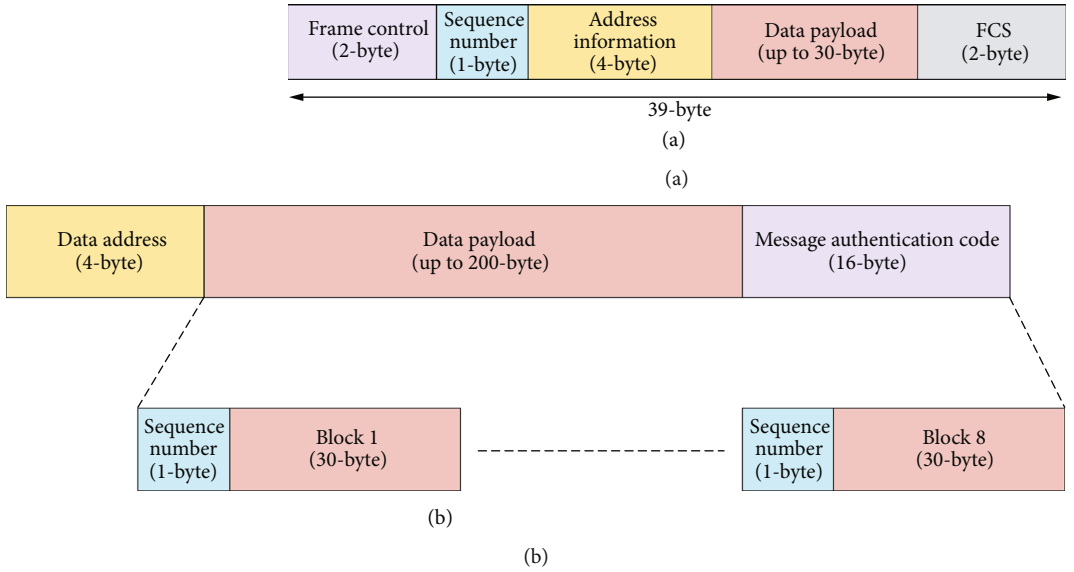

Figure 4: (a) The media access layer frame format of MASDA. (b) MASDA packet format ($k = 4$); the data packet is divided into 8 data slices (≤30 bytes) and a serial number is assigned.

EC-EG and EC-OU. However, neither EC-EG nor EC-OU can support the notable attributes of MASDA, such as the multiple application data aggregation and recovery of aggregated data in BS. By analyzing the results, it can also be observed that there is a trade-off between the security and the length of ciphertext in MASDA. If the stronger security is the first demand, a larger modulus is recommended. If the lifetime of WSNs is the major goal, a smaller modulus is practicable.

The energy cost is also correlated with the amount of data transmission. We compared MASDA with EC-EG and EC-OU in data transmission. Different number of sensor nodes (120, 150, and 180) are randomly scattered in a 100mÃ—100 m square area with BS in the center. Each node belongs to a cluster and CH is determined according to LEACH protocol. The probability of a node being selected as CH is 0.05. The simulation result of each mechanism is the average of 10 simulation rounds. In MASDA, CH is

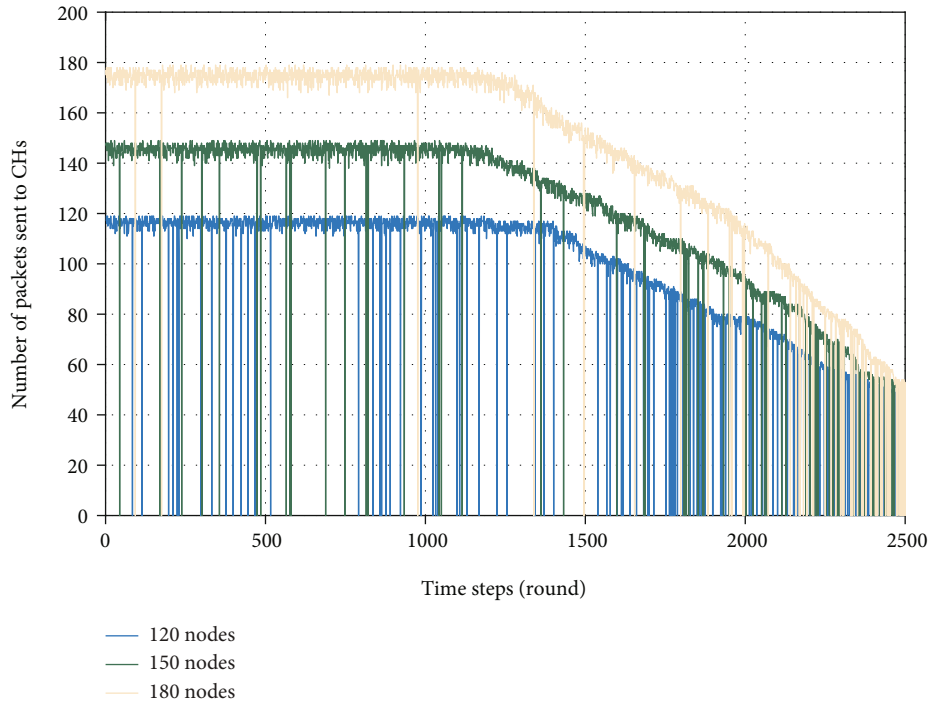

Figure 5: Node survival.

allowed to aggregate the data being forwarded by other CHs in order to fuse the data of different clusters to a single ciphertext. EC-EG and EC-OU can only aggregate a single application. CSMA is adopted as the media access control mechanism. The upper bound of retransmission is five times and the initial energy of node is E_i. Then, we can evaluate the energy cost of MASDA. Two energy models are usually adopted in data transmission, the free space model and the multipath attenuation model as shown in [35]

$$E = \begin{cases} l * \left(E_{\text{elec}} + \varepsilon_{\text{fs}} d^2\right), d < d_0, \\ l * \left(E_{\text{elec}} + \varepsilon_{\text{amp}} d^4\right), d \geq d_{0.} \end{cases} \quad (25)$$

The maximum frame of media access control layer is 39 bytes, and its structure is depicted in Figure 4(a). Noticed that the bit error rates are various if different sizes of data packets are adopted. Therefore, it is necessary to divide the aggregation ciphertext and mark the sequence number in each slice. When $k = 4$ and $|q| = 341$, the length ciphertext of MASDA is almost 200 bytes and the application layer data packet (ciphertext and message authentication code) of MASDA can be divided into 8 data packets before sending. Each packets is less than 30 bytes, and the serial number is assigned. After receiving these small data blocks, BS can reconstruct the original data according to the assigned sequence numbers. Figure 4(b) shows the slicing pattern of ciphertext (EC-EG and EC-OU also divided their data into the same slices as MASDA.)

Figure 5 shows the survival of nodes in the network with different numbers of sensor nodes. As the number of surviving nodes decreases with the increase of time, the number of packets sent in the whole network decreases gradually. The frame format of media access layer of EC-OU and EC-EG is the same as that of MASDA, so the node survival of the three mechanisms is similar. In addition to the packet format, we also control all sensor nodes to be in the survival state, so as to fairly compare the transmission volume of the three mechanisms.

Figure 6 shows the total data transfer volume for different mechanisms. We obtain the same ciphertext length by adjusting the modulus $|q|$. When $k = 2$, the length of EC-EG ciphertext is 326 bits. To achieve the same length of ciphertext, we set $|q| = 107$ in MASDA. When $k > 2$, we set $|q| = 163$ in MASDA. At this time, the key strength of MASDA is much higher than EC-EG. In addition, we also simulate the data transmission volume of MASDA under the same ciphertext length as EC-OU ($n = 1024$) (when $k = 2$,$|q| = 341$; when $k = 3$, $|q| = 257$; and when $k = 4$, $|q| = 203$). The specific analysis is shown in Figure 6. When $k = 2$, the total amount of data transmission of EC-OU is the largest. However, the key strength of MASDA is higher than that of EC-OU and EC-EG. When $k > 2$, our scheme is significantly better than EC-EG and slightly lower than EC-OU. This is because the key length of MASDA exceeds 163 bits, and the cost of key generation will undoubtedly increase. The results show that our scheme achieves a good compromise in terms of energy consumption and safety. According to the simulation results, we can choose the desired mechanism according to the amount of data transmission and the security level of different applications.

6.3. Aggregation Accuracy. In some traditional data aggregation (DA) mechanisms, the sensing data may be changed due to the compression algorithm and the data accuracy is an important indicator to evaluate the security of DA. EC-EG, EC-OU, and MASDA aggregate the sensing data only

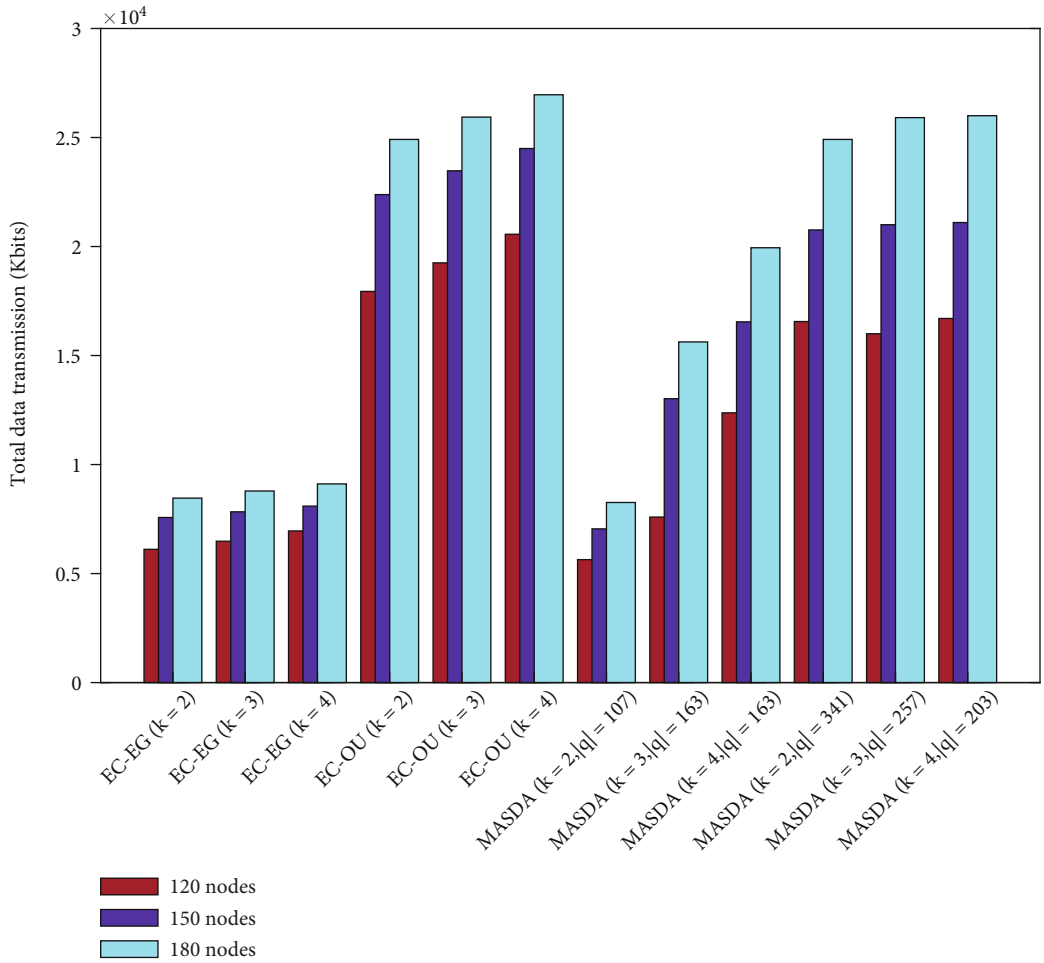

Figure 6: Total data transfer volume of different scenarios.

Table 5: Data accuracy.

Time interval (seconds)	0.5	1	1.5	2	2.5
EC-OU	0.65	0.83	0.86	0.84	0.91
EC-EG	0.67	0.83	0.85	0.86	0.88
MASDA	0.67	0.85	0.87	0.87	0.88

depending on ciphertexts and the raw sensing data are not be compressed during the data aggregation. Therefore, SDA mechanisms based on homomorphic encryption can provide better performance in accuracy. Meanwhile, the bit error rate is another factor which affects the accuracy of DA. In an ideal situation, we generally assume that there is no the data conflict and the packet loss in a network and the accuracy of DA may reach 100%. However, the data conflict and the packet loss are inevitable and the data accuracy is various in different mechanisms. In [36], Li et al. proposed a metric for data accuracy and defined it as "the ratio of the actual sum of the original data to the sum of the data received by BS." We adopted this definition and deployed a simulation network with 120 sensors and the bit error rate is 5%. We compared the data accuracy of EC-EG, EC-OU, and MASDA as shown in Table 5.

It can be seen that MASDA is the best one in these mechanisms. However, their differences are not so remarkable. After the 240th round, the data accuracy tends to be stable and the data accuracy is mainly affected by the channel noise at this time. The reason is that EC-EG, EC-OU, and MASDA transmit data packets in the same way. At the same time, because the channel congestion in the transmission process decrease with the increase of time, the accuracy of aggregated data is also improved. Noticed that MASDA has other prominent advantages compared with EC-EG and EC-OU. It can be applied to multiple applications and provide confidentiality and integrity simultaneously. This indicates that we expand the application scope of SDA without losing the security and effectiveness in terms of the encryption strength, the energy consumption, and the data accuracy.

7. Conclusion

In this paper, we discussed a the multiple applications secure data aggregation mechanism (MASDA) and its applcation in IoT. This scheme can encrypt the different application data in a single ciphertext and aggregate the encrypted data in a relay node in order to reduce the overhead and ensure the

data integrity through the homomorphic message authentication code. MASDA has desired confidentiality and integrity and the security analysis and simulation experiments show that it can maintain the higher security, the longer lifetime, and the better accuracy. Although our scheme may provide a solution for security aggregation in WSNs, there are still many meaningful topics to be studied in the future. We should verify the impact of differen packet loss rates on the aggregation accuracy and design a more robust homomorphic encryption scheme. The decrease in communication cost is also a huge challenge in subsequent studies.

Data Availability

The datasets generated or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by NSF of China under Grants 61672321, 61832012, 61771289, and 61373027, Nature Science Foundation of Shandong Province under Grants ZR2021MF075, ZR2021QF050; and Shandong Graduate Education Quality Improvement Plan under Grants SDYY17138, SDYKC21097.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[2] O. Sadio, I. Ngom, and C. Lishou, "Design and prototyping of a software defined vehicular networking," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 842–850, 2020.

[3] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1–19, 2015.

[4] K. Yu, B. Yan, J. Yu, H. Chen, and A. Dong, "Methods of improving secrecy transmission capacity in wireless random networks," *Ad Hoc Networks*, vol. 117, article 102492, 2021.

[5] P. L. Díez, I. Gabilondo, E. Alarcón, and F. Moll, "Mechanical energy harvesting taxonomy for industrial environments: application to the railway industry," *IEEE Transactions on Intelli- gent Transportation Systems*, vol. 21, no. 7, pp. 2696–2706, 2019.

[6] B. Montrucchio, E. Giusto, M. G. Vakili, S. Quer, R. Ferrero, and C. Fornaro, "A densely-deployed, high sampling rate, open-source air pollution monitoring WSN," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15786–15799, 2020.

[7] K. Yu, J. Yu, X. Cheng, D. Yu, and A. Dong, "Efficient link scheduling solutions for the internet of things under Rayleigh fading," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2508–2521, 2021.

[8] L. Zhou, C. Ge, S. Hu, and C. Su, "Energy-efficient and privacy-preserving data aggregation algorithm for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3948–3957, 2020.

[9] K. Yu, J. Yu, and A. Dong, "Cooperative communication and mobility for securing URLLC of future wireless network," *IEEE Transactions on Vehicular Technology*, 2022.

[10] J. Huang and B.-H. Soong, "Cost-aware stochastic compressive data gathering for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1525–1533, 2018.

[11] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks: filtering out the attacker's impact," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 681–694, 2014.

[12] M. Kaur and T. Su, "Data aggregation algorithms for wireless sensor network: a review," *Ad Hoc Networks*, vol. 100, article 102083, 2020.

[13] 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.

[14] D. Xiao, M. Li, M. Wang, J. Liang, and R. Liu, "Low-cost and high-efficiency privacy-protection scheme for distributed compressive video sensing in wireless multimedia sensor networks," *Journal of Network and Computer Applications*, vol. 161, article 102654, 2020.

[15] C. Zhao, W. Zhang, Y. Yang, and S. Yao, "Treelet-based clustered compressive data aggregation for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4257–4267, 2014.

[16] A. Sreenivasulu and P. C. Reddy, "NLDA non-linear regression model for preserving data privacy in wireless sensor networks," *Digital Communications and Networks*, vol. 6, no. 1, pp. 101–107, 2020.

[17] H. Wang, G. Han, C. Zhu, S. Chan, and W. Zhang, "TCSLP: a trace cost based source location privacy protection scheme in WSNs for smart cities," *Future Generation Computer Systems*, vol. 107, pp. 965–974, 2020.

[18] S. Papadopoulos, A. Kiayias, and D. Papadias, "Exact in-network aggregation with integrity and confidentiality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1760–1773, 2012.

[19] J. Xu, G. Yang, Z. Chen, and Q. Wang, "A survey on the privacy-preserving data aggregation in wireless sensor networks," *China Communications*, vol. 12, no. 5, pp. 162–180, 2015.

[20] W. Y. Alghamdi, H. Wu, and S. S. Kanhere, "Reliable and secure end-to-end data aggregation using secret sharing in wsns," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, San Francisco, CA, USA, 2017.

[21] H. Zhong, L. Shao, J. Cui, and Y. Xu, "An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 1–12, 2018.

[22] P. Gope and B. Sikdar, "Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1554–1566, 2018.

[23] W. Fang, X. Z. Wen, J. Xu, and J. Z. Zhu, "CSDA: a novel cluster-based secure data aggregation scheme for wsns," *Cluster Computing*, vol. 22, Supplement 3, pp. 5233–5244, 2019.

[24] B. Arazi, "Message authentication in computationally constrained environments," *IEEE Transactions on Mobile Computing*, vol. 8, no. 7, pp. 968–974, 2009.

[25] L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao, "A secure and efficient id-based aggregate signature scheme for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 546–554, 2017.

[26] O. Boudia, S. M. Senouci, and M. Feham, "A novel secure aggregation scheme for wireless sensor networks using stateful public key cryptography," *Ad Hoc Networks*, vol. 32, pp. 98–113, 2015.

[27] K. A. Shim and C. M. Park, "A secure data aggregation scheme based on appropriate cryptographic primitives in heterogeneous wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2128–2139, 2015.

[28] J. Qian, Z. Cao, X. Dong, J. Shen, Z. Liu, and Y. Ye, "Two secure and efficient lightweight data aggregation schemes for smart grid," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2625–2637, 2020.

[29] S. Viswanathan and A. Kannan, "Elliptic key cryptography with Beta Gamma functions for secure routing in wireless sensor networks," *Wireless Networks*, vol. 25, no. 8, pp. 4903–4914, 2019.

[30] R. Gallant, R. Lambert, and S. Vanstone, "Improving the parallelized pollard lambda search on anomalous binary curves," *Mathematics of Computation*, vol. 69, no. 232, pp. 1699–1706, 1999.

[31] A. Kamal, H. Dahshan, and A. D. Elbayoumy, "A new homomorphic message authentication code scheme for network coding," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 520–524, San Jose, CA, USA, 2020.

[32] J. Cui, L. Shao, H. Zhong, Y. Xu, and L. Liu, "Data aggregation with end-to-end confidentiality and integrity for large-scale wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 1022–1037, 2018.

[33] D. Vinodha, E. A. Mary, and D. Mohana, "A novel multi functional multi parameter concealed cluster based data aggregation scheme for wireless sensor networks (NMFMP-CDA)," *Wireless Networks*, vol. 27, no. 2, pp. 1111–1128, 2021.

[34] S. Ozdemir and Y. Xiao, "Integrity protecting hierarchical concealed data aggregation for wireless sensor networks," *Computer Networks*, vol. 55, no. 8, pp. 1735–1746, 2011.

[35] X. Du, Z. Zhou, Y. Zhang, and T. Rahman, "Energy-efficient sensory data gathering based on compressed sensing in iot networks," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–16, 2020.

[36] H. Li, K. Lin, and K. Li, "Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks," *Computer Communications*, vol. 34, no. 4, pp. 591–597, 2011.