Hindawi
Journal of Electrical and Computer Engineering
Volume 2017, Article ID 3956027, 9 pages
https://doi.org/10.1155/2017/3956027

Research Article

Multiorder Fusion Data Privacy-Preserving Scheme for Wireless Sensor Networks

Mingshan Xie,[1,2,3] Yong Bai,[1,2] Mengxing Huang,[1,2] and Zhuhua Hu[1,2]

[1]*College of Information Science & Technology, Hainan University, Haikou 570228, China*
[2]*State Key Laboratory of Marine Resource Utilization in South China Sea, Haikou 570228, China*
[3]*College of Network, Haikou College of Economics, Haikou 571127, China*

Correspondence should be addressed to Yong Bai; bai@hainu.edu.cn

Received 30 March 2017; Revised 29 July 2017; Accepted 7 September 2017; Published 16 October 2017

Academic Editor: Xiong Li

Privacy-preserving in wireless sensor networks is one of the key problems to be solved in practical applications. It is of great significance to solve the problem of data privacy protection for large-scale applications of wireless sensor networks. The characteristics of wireless sensor networks make data privacy protection technology face serious challenges. At present, the technology of data privacy protection in wireless sensor networks has become a hot research topic, mainly for data aggregation, data query, and access control of data privacy protection. In this paper, multiorder fusion data privacy-preserving scheme (MOFDAP) is proposed. Random interference code, random decomposition of function library, and cryptographic vector are introduced for our proposed scheme. In multiple stages and multiple aspects, the difficulty of cracking and crack costs are increased. The simulation results demonstrate that, compared with the typical Slice-Mix-AggRegaTe (SMART) algorithm, the algorithm proposed in this paper has a better data privacy-preserving ability when the traffic load is not very heavy.

1. Introduction

Nowadays, the application of Internet of things (IoT) is becoming more diverse. As an important part of the Internet of things, the wireless sensor network (WSN) has been widely used in all aspects of our lives (e.g., military surveillance, patient monitoring, forest monitoring, etc.). A wireless sensor network is a self-organizing network composed of a large number of sensor nodes. It has the characteristics of being resource constrained, distributed, multihop, wireless communication, and so on. WSN sensor nodes are placed in the public, untrusted, and even malicious intrusion environment and exchange the data using wireless communication. These make the data easy to intercept for WSN. WSN faces serious privacy data leakage risk. Especially in medical and military applications, data security requirements are very high. It is necessary to study the privacy protection of WSN.

It is the main task for wireless sensor networks to collect useful information, relying on a large number of nodes scattered in the environment, so that people can analyze and process the information. Data fusion technology is one of the key technologies for WSN. Many of the existing researches have increased the privacy protection function on the basis of data fusion. For instance, in [1], Conti et al. design a private data aggregation protocol that does not leak individual sensed values during the data aggregation process. It enhances the robustness, and the node computing complexity and data transmission are not large.

In [2–5], Wenbo et al. proposed the privacy-preserving algorithms based on data fusion. The SMART algorithm is widely used in these algorithms. In SMART, the data is divided into slices, and then the slices are sent to the randomly selected neighbor nodes; finally, the data fusion is carried out along the data fusion tree. In this paper, the method of decomposing data by random function is proposed. We make some improvements to SMART. We use the random function to decompose the data to increase the difficulty of cracking the data in the data aggregation point.

The purpose of privacy protection is to increase the cost and difficulty of acquisition for eavesdropper. The generation of interference information can greatly increase the cost and difficulty of eavesdropping.

Generally, the energy resources of the sampled sensors are limited. Energy resources are related to the lifetime of the whole WSN, so the energy consumption is usually the key problem of data fusion in WSNs. Now most of the wireless sensor networks take sleep strategies which randomly select a node as a sentinel node, to achieve compressive sampling, but now most of the compressed sampling schemes do not provide privacy protection. In fact, interference information can be generated by nodes randomly selected from the sleep nodes. In this paper, we propose a method to generate interference codes in the multiorder fusion data protection algorithm.

In the traditional password protection work, Girao et al. proposed a privacy-preserving solution for data fusion in [6, 7]. They use homomorphic encryption so that nodes can effectively fuse the data without decrypting data. In order to increase the defeat solution difficulty, this paper puts forward the strategy of password vector protection.

Now the eavesdropper's crack technology is richer. It is difficult to protect the security of wireless sensor networks with only one method. A variety of methods need to be organically combined so that layers of protection are formed. For this purpose, this paper proposes a privacy-preserving mechanism combined with active protection and passive protection and adopts multiorder fusion protection strategy for WSN, to increase the cost and difficulty of interception or eavesdropping in a nonlinear way.

The rest of this paper is organized as follows. Section 2 provides some related work. Section 3 describes the model of privacy-preserving data aggregation in WSN. Section 4 provides our multiorder fusion algorithms for private data aggregation. Section 5 evaluates the proposed schemes for cost and difficulty. We summarize our work and lay out future research direction in Section 6.

2. Related Work

In typical wireless sensor networks, sensor nodes are usually limited in resources and energy. Data aggregation is necessary for wireless sensor networks. Based on the management pattern of cluster structure, in [1], Conti et al. proposed a privacy-preserving algorithm for data fusion. In particular, neither the base station (BS) nor other nodes are able to compromise the privacy of an individual node's sensed value. Bista et al. proposed a new set of data fusion privacy protection solutions. The proposed scheme applies the additive property of complex numbers in [8, 9]. All of them have the advantages of computational complexity and small amount of data transmission.

In [2–5], Wenbo et al. proposed the privacy-preserving data aggregation (PDA), which includes two algorithms, cluster-based private data aggregation (CPDA) and Slice-Mix-AggRegaTe (SMART). But the amount of calculation of CPDA is great, and data traffic of SMART is very large. SMART algorithm in [2] is the most closely related to the algorithm proposed in this paper. The SMART algorithm uses hop-by-hop data fusion mode and node-to-node encryption and decryption mode. It can prevent the invasion of external intruders, in the case of ensuring the accuracy of data fusion. It can guarantee the internal trusted node to obtain privacy data. The SMART algorithm is divided into three steps: slicing, mixing, and aggregating.

In [10–13], some features are added on the basis of SMART. The privacy-preserving algorithm based on fusion is extended. The energy consumption is reduced in document 10. Reference [11] focuses on improving the accuracy of data fusion. Reference [12] adds data integrity verification. The authors add fault tolerance in [13].

In [6, 7], Girao et al. proposed a privacy-preserving solution for data fusion. They adopted homomorphic encryption. This algorithm can make the nodes implement effective fusion of data, without the need of decrypting data. In order to increase the difficulty of crack, this paper puts forward the strategy of using password vector to protect the data.

In [14], the authors proposed a k-anonymization clustering method (k-ACM) that provides a k-anonymity framework with two levels of privacy for WSNs. Zhang et al. proposed a security privacy-preserving data aggregation model, which adopts a mixed data aggregation structure of tree and cluster for the wearable wireless sensors in [15]. In [16], Cao et al. proposed a privacy-preserving and auditing-supporting outsourcing data storage scheme by using encryption and digital watermarking. They used logistic map-based chaotic cryptography algorithm to preserve the privacy of outsourcing data.

Thus, it can be seen that there are many schemes about privacy protection. How to combine a variety of privacy protection strategies is also an urgent need to solve the problem. We propose the multiorder fusion data privacy-preserving (MOFDP) algorithm for wireless sensor networks by extending the privacy protection.

3. System Model

The MOFDP algorithm is based on the cluster structure of wireless sensor networks. It is assumed that there is a cluster head node N_0 which can be a sink node in each cluster structure. It is a high resource node, which can be responsible for data collection and integration. It can also be used as a query node to perform query tasks. In each cluster structure of WSN, other nodes can be the subnodes which are responsible for collecting data. Suppose that the set of subnodes is $\{\text{Node}_1, \text{Node}_2, \ldots, \text{Node}_{\text{NumNode}}\}$, where NumNode is the number of child nodes. The specific structure of the cluster of WSNs is shown in Figure 1.

The MOFDP algorithm achieves multiorder data privacy protection. In Figure 1, "—" denotes order-1 data privacy protection. "- · -" denotes order-2 data privacy protection. When NumNode = 4, there are four "—" and "- · -", respectively. "→" represents order-3 data privacy protection.

In the privacy-preserving algorithm of this paper, N_0 node is very important. In this paper, it is assumed that N_0 is very difficult to be captured. In this paper, the wireless sensor network takes the sleeping strategy. The sampling frequency of the whole network is dynamically adjusted by N_0. In each sampling process, all the subnodes are not sampled at the

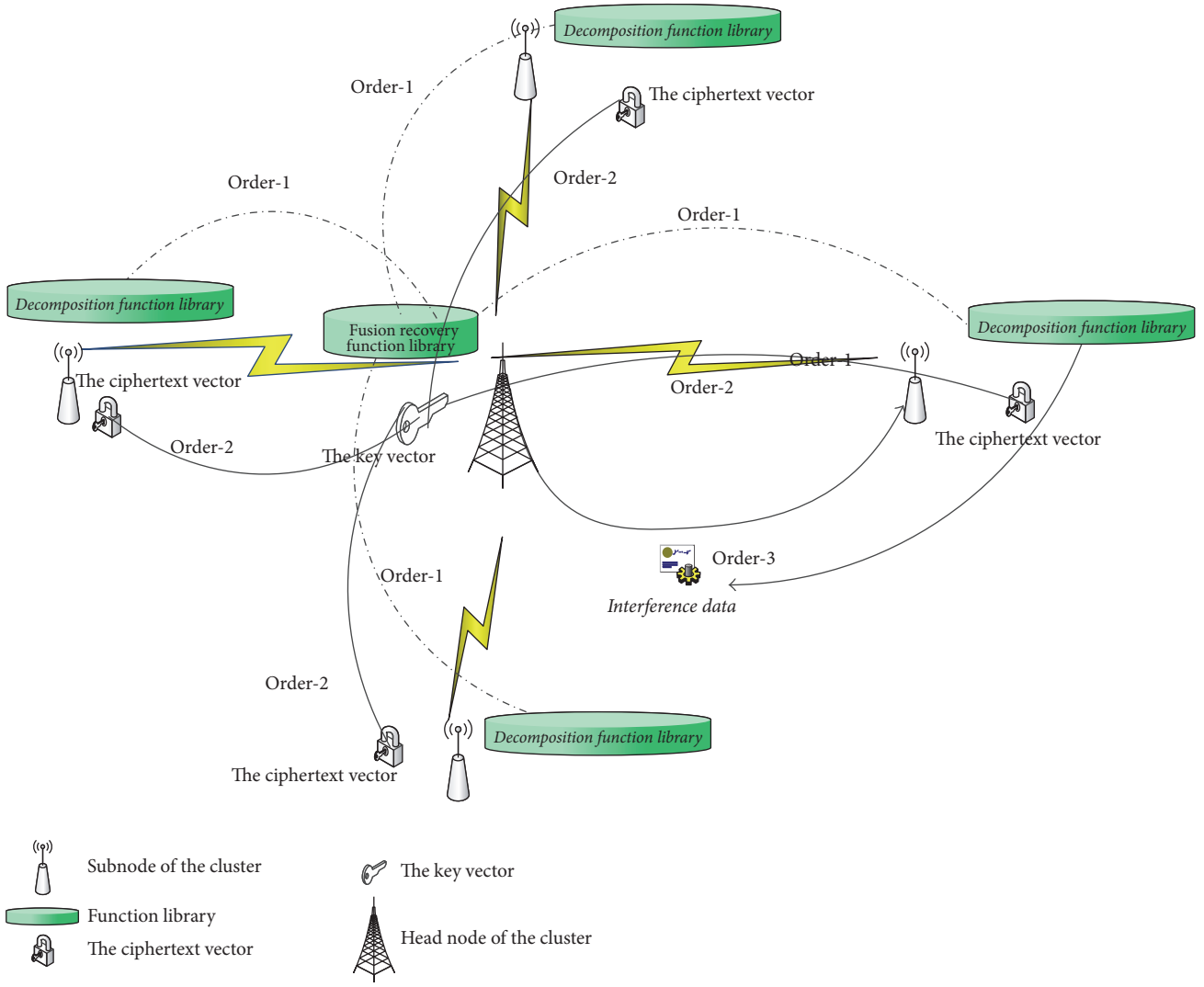

Figure 1: MOFDP model based on cluster structure.

same time, but only one random subnode is sampled. Any node that takes the sample is determined by the N_0 node according to the algorithm to generate random values. The fusion recovery function library and the key vector both are in the head node N_0. The detailed algorithm will be described in the rest of this paper.

4. The Protection Mechanisms of Multiorder Fusion Protection

In this paper, we use three order models to protect data privacy. The first step is the use of interference code protection. In the second phase, the data is decomposed by a random function. In the third stage, we use cryptographic vectors to secure data.

4.1. Construction of Decomposition Function Library. In [2], the slice-mixed aggregation (SMART) uses the segmentation and reorganization technology to complete the privacy-preserving data aggregation. The basic idea of SMART is that the sensor node randomly divides the original data into many data slices; the hop-by-hop encryption mechanism is adopted, and the exchanged data slices are randomly selected by the neighbor nodes; it performs the summation operation for all the received data slices and uploads the results to the base station; the base station can obtain all the received data and get the accurate SUM aggregation results.

We do not simply use the fixed method to segment the data in this paper. The decomposition function is randomly extracted from the decomposition function library, and then the collected data are decomposed and sent in separate time periods. It increases the difficulty of cracking data, because the decomposition function is random, and the decomposition function library and the fusion recovery function library must be obtained at the same time in order to recover the signal. Due to the implementation of transmitting data in the interval time, the number of elements in the solution

vector of decomposition function is not uniform. It is difficult to determine the number of signals to be intercepted for cracking data; thus, the difficulty of the crack is increased.

Set $D_i(t)$ as the data collected by the ith subnode at t moment. We define the data decomposition function as

$$\begin{aligned}\mathrm{Dec}_q\left(D_i(t)\right) &= \mathrm{DD}_i \\ &= \left\{d_{(1,i)}(t), d_{(2,i)}(t), \ldots, d_{(N,i)}(t)\right\},\end{aligned} \tag{1}$$

where DD_i is the solution vector. $D_{(j,i)}(t)$ $(j = 1, 2, \ldots, N)$ denotes the elements of DD_i. The number N of components of the solution vector is determined by the decomposition function. The q is the index number of the decomposition function $\mathrm{Dec}(*)$.

The data decomposition function library of subnodes is denoted as

$$\mathrm{Decs} = \left\{q \in \mathbb{Z} \mid \mathrm{Dec}_q\left(D_i(t)\right)\right\}. \tag{2}$$

The data fusion recovery function is interpreted as

$$\begin{aligned}\mathrm{Cec}_p\left(\mathrm{DD}_i\right) &= D_i(t) \\ &= \mathrm{Cec}_p\left(\left\{d_{(1,i)}(t), d_{(2,i)}(t), \ldots, d_{(N,i)}(t)\right\}\right) \\ &= D_i(t),\end{aligned} \tag{3}$$

where p is the index number of the fusion recovery function $\mathrm{Cec}(*)$.

The fusion recovery function library of sink node is defined as

$$\mathrm{Cecs} = \left\{p \in \mathbb{Z} \mid \mathrm{Cec}_p\left(\mathrm{DD}_i\right)\right\}. \tag{4}$$

A decomposition function corresponds to a fusion recovery function. The fusion recovery function is the inverse operation of the decomposition function. The relationship between them is $\mathrm{Dec}_u(\mathrm{Cec}_u(\mathrm{DD}_i)) = \mathrm{Cec}_u(\mathrm{Dec}_u(D_i(t))) = D_i(t)$, $u \in \mathbb{Z}$. The total number of functions in the decomposition function library corresponds to the total number of functions in the fusion recovery function library.

4.2. End-to-End Encryption Used Key Vectors. In many practical applications, wireless sensor networks must consider privacy protection to ensure that the data collected by each node can only be accessed by the authorized users. Only the authorized nodes can transmit the data to the sink node to ensure the correctness and integrity of the data. In this paper, the idea of encryption is end-to-end encryption.

The sensed data being passed to nonleaf aggregators are revealed for the sake of middle-way aggregation in the hop-by-hop aggregation protocols in [17]. Compared with the hop-by-hop encryption mechanism, the intermediate nodes in the end-to-end encryption mechanism can save the cost of encryption and decryption and reduce the time delay. The end-to-end encryption mechanism is as follows: the sensor node and the sink node share the key encryption, and then the sink node implements the decryption process. The end-to-end encryption mechanism requires data aggregation on encrypted data. Homomorphic encryption can be used to achieve the sum or product operation on the ciphertext, which can effectively support data aggregation in the encrypted data.

```
(1) While no error
(2) TSID = Rand(1, NumNode);
(3) TFID = Rand(1, NumFunction);
(4) Send MES to Node(TSID);
(5) Endwhile;
```

Algorithm 1: Sink node randomly generated number.

The ith sampling node:

$$\begin{aligned}\mathrm{cd}_1 &= \mathrm{Encode}\left(d_{(1,i)}(t), k_{(i,1)}, M\right) = d_{(1,i)}(t) + k_{(i,1)} \oplus M; \\ \mathrm{cd}_2 &= \mathrm{Encode}\left(d_{(2,i)}(t), k_{(i,2)}, M\right) = d_{(2,i)}(t) + k_{(i,2)} \oplus M; \\ &\vdots \\ \mathrm{cd}_N &= \mathrm{Encode}\left(d_{(N,i)}(t), k_{(i,N)}, M\right) = d_{(N,i)}(t) + k_{(i,N)} \oplus M,\end{aligned} \tag{5}$$

where $\oplus$ is denoted as the mode operation and M is the system parameter. $\mathrm{CD}_i = \{\mathrm{cd}_1, \mathrm{cd}_2, \ldots, \mathrm{cd}_N\}$ is ciphertext vector. The key vector $K_i = \{k_{(i,1)}, k_{(i,2)}, \ldots, k_{(i,N)}\}$ is shared by the ith subnode and the head node.

$\mathrm{DD}_i = \{d_{(1,i)}(t), d_{(2,i)}(t), \ldots, d_{(N,i)}(t)\}$ is the plaintext vector of sampling node.. The relationship then follows as

Ciphertext fusion value of head node: $\mathrm{CD}_{1\cdots N} = (\mathrm{Cec}_p(\mathrm{CD}_i)) \oplus M$;

Decryption of head node: $\mathrm{Decode}(\mathrm{CD}_{1\cdots N}, \mathrm{Cec}_p(K_i), M) = \mathrm{CD}_{1\cdots N} - \mathrm{Cec}_p(K_i) \oplus M$.

4.3. Sink Node Randomly Generated Number. Because the sleeping strategy is used in the wireless sensor network in this paper, under the condition that the probability of abnormal event is not high, all nodes are not sampled at the same time, in each sampling process. The sink node randomly generates the really sampling node number by the algorithm and notifies its sampling. This truly sampled node is called a sentinel node.

TSID is denoted as the truly sampling node number. TFID is in terms of the number of decomposition function. Active signal group MES = (TSID, TFID, CONDITION), where CONDITION is the condition to activate node sampling. NumNode is the total number of subnodes. NumFunction is the total number of functions in the decomposition function library. The algorithm of randomly generating number is Algorithm 1.

4.4. Generation of Random Interference Data. It is possible to randomly select a node to generate interference data. In this way, the probability that the eavesdropper intercepts the data in the channel and the information of the source is determined by the number of interfering data nodes. Assuming that the number of the real sampling nodes is

(1) While no error
(2) $t = 0$;
(3) RSC = 0;
(4) IF Receive(MES);
(5) RSC = RSC + 1;
(6) $D_{\mathrm{TSID}}(t)$ = SAMPLE;
(7) $D_{\mathrm{TSID}} = \mathrm{Dec}_{\mathrm{TFID}}(D_{\mathrm{TSID}}(t))$
(8) $\mathrm{cd}_1 = \mathrm{Encode}(d_{(1,i)}(t), k_{(i,1)}, M)$;
(9) $\mathrm{cd}_2 = \mathrm{Encode}(d_{(2,i)}(t), k_{(i,2)}, M)$;
(10) ...
(11) $\mathrm{cd}_N = \mathrm{Encode}(d_{(N,i)}(t), k_{(i,N)}, M)$;
(12) For $tt = 1$ to N
(13) SEND(TSID, TFID, cd_{tt}) to sink;
(14) $tt = tt + 1$;
(15) Endfor
(16) ELSE IF RSC >= 2||!SAMPLING
(17) Then NOISYFD;
(18) RSC = 0;
(19) SEND FD to sink
(20) ENDIF
(21) ENDIF
(22) $t = t + 1$
(23) Endwhile

Algorithm 2: Subnode really sample and generate interference data.

$\mathrm{Num}_{\mathrm{datanodes}}$, the number of nodes generating the interference data is $\mathrm{Num}_{\mathrm{datainter}}$, and the probability p of the eavesdropper intercepting the false data is

$$p \approx \frac{\mathrm{Num}_{\mathrm{datainter}}}{\mathrm{Num}_{\mathrm{datanodes}}}. \quad (6)$$

In case there are one node generating the interference data and one real sampling node, the probability p of the eavesdropper intercepting the false data approximately equals 50%. If there are two nodes generating the interference data and one real sampling data, the probability p approximately equals 33.33%.

The selection problem of nodes generating interference data must be considered. If the head node notifies the subnodes to generate random interference codes, this increases the communication cost and instability. The method of this paper is shown in Algorithm 2. In $n - 1$ nonsampling nodes, each node's sampling threshold is used to decide whether to transmit random values which can be used as the interference data. Since the real sampling nodes are random, the nodes that generate the interference data by Algorithm 2 are random too.

5. Data Privacy-Preserving Algorithm Based on Multiorder Fusion Protection

In order to ensure that the wireless sensor networks can securely collect data and secure the data collected by wireless sensor networks (user privacy information, especially, is not stolen), MOFDP algorithm has increased the difficulty of cracking from three aspects to achieve the 3-order stereo protection, based on the idea of increasing the difficulty of hacking.

In the aspect of data decomposition and reception, the random selection function in the function library is used to decompose and fuse data. In data encryption, the use of key vector for the end-to-end encryption has increased the difficulty of breaking. In addition, data is protected by sending interference data. Finally, multiorder data privacy protection in wireless sensor networks is achieved.

5.1. The Algorithm of Data Sampling and Sending for Subnode. In the process of sampling, each node is sampled once and then accumulated once. RSC is denoted as a cumulative variable.

The sampling time is t. SAMPLE is denoted as sampling, quantization, and coding to obtain sampled data. $\mathrm{Dec}_{\mathrm{TFID}}(D_{\mathrm{TSID}}(t))$ denotes that the decomposition function is called by TFID from the decomposition function library. In this algorithm the ciphertext is sent out in separate time periods. The symbol tt is accumulator for each time period, which is determined by decomposition function. NOISYFD denotes that false interference data are randomly generated. FD is in terms of false data.

5.2. The Algorithm of Data Fusion Recovering for Head Node. See Algorithm 3.

6. Discussion and Simulation Experiment

6.1. Protection Cost. In this paper, the real sampling node number and the interference node number are randomly generated and distributed uniformly. Thus, the energy balance of wireless sensor networks is guaranteed, and the service lifetime is effectively prolonged.

Most of the computational work of MOFDP is focused on head node. Because the head node is a high resource node, the calculation of energy consumption and communication energy consumption has been adequately supported and guaranteed.

6.2. Crack Cost. Since the aim is to extract the encryption function and the decryption function randomly from the library function in this paper's algorithm, it is necessary to break all the functions to ensure the success rate of crack function. The cost of cracking function is in terms of $\mathrm{cost}_{\mathrm{getfun}}$. The cost of cracking all functions in the library is $\mathrm{NumFuntion} \times \mathrm{cost}_{\mathrm{getfun}}$.

Because of the function decomposition, the cost of collecting data is increased. If the eavesdropper synthesized the data, it is needed that he has to break the communication links between the nodes and spend a certain amount of time gathering the data. The risk that the eavesdropper is found is increased. The cost of collecting one piece of data is $\mathrm{cost}_{\mathrm{getdata}}$. Since MOFDP uses the random function extraction, the eavesdropper has to take the maximum number of function decomposition components in the function library as the

```
(1) While no error
(2)     IF Receive(TSID) == TSID and Receive(TFID) == TFID;
(3)         CD_{1...N} = (Cec_TFID(CD_TSID)) mode M;
(4)         Decode(CD_{1...N}, Cec_TFID(K_i), M);
(5)     Else reject interference data and not receiving
(6)     Endif
(7) End while
```

Algorithm 3: Data fusion recovering of head node.

number of times to collect data every time. The cost of collecting data is

$$\mathrm{MAX}\left(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\right) \times \mathrm{cost}_{\mathrm{getdata}}. \tag{7}$$

The MOFDP algorithm decomposes the data into many components, which form a solution vector. In order to encrypt data for each component, the key vector is introduced. The number of components of the key vector is equal to the number of decomposition components. The number of key vectors forms a vector: $\{N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\}$. The cost of breaking a key is set as $\mathrm{cost}_{\mathrm{key}}$. In the process of eavesdropping, the cost of cracking passwords is

$$\mathrm{MAX}\left(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\right) \times \mathrm{cost}_{\mathrm{key}} \tag{8}$$

Due to the random interference data, the cost of identifying the authenticity is required. MOFDP uses random nodes to send interference data, so distinguishing the authenticity of the data requires the existence of the characteristics of each interference data of nodes to be analyzed. The cost of setting up one interference data is $\mathrm{cost}_{\mathrm{jarm}}$. The cost of identifying the authenticity is defined as $\mathrm{NumNode} \times \mathrm{cost}_{\mathrm{jarm}}$.

MOFDP preventive measures have three stages of protection. As long as the prevention of any stage cannot break, the right real data cannot be obtained. The total cracking cost of MOFDP is defined as

$$\begin{aligned}\mathrm{COST}_{\mathrm{MOFDP}} &= \mathrm{NumFuntion} \times \mathrm{cost}_{\mathrm{getfun}} \\ &\quad + \mathrm{MAX}\left(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\right) \\ &\quad \times \mathrm{cost}_{\mathrm{getdata}} + \mathrm{NumNode} \times \mathrm{cost}_{\mathrm{jarm}} \\ &\quad + \mathrm{MAX}\left(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\right) \\ &\quad \times \mathrm{cost}_{\mathrm{key}} \\ &= \mathrm{MAX}\left(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}}\right) \\ &\quad \times \left(\mathrm{cost}_{\mathrm{key}} + \mathrm{cost}_{\mathrm{getdata}}\right) \\ &\quad + \mathrm{NumFuntion} \times \mathrm{cost}_{\mathrm{getfun}} \\ &\quad + \mathrm{NumNode} \times \mathrm{cost}_{\mathrm{jarm}}.\end{aligned} \tag{9}$$

The classic SMART algorithm also uses the idea of data segmentation. It decomposes the data into J slices and then encrypts and transmits $(J-1)$ data. The cracking cost of SMART approximately is defined as

$$\mathrm{COST}_{\mathrm{SMART}} = (J-1) \times \left(\mathrm{cost}_{\mathrm{key}} + \mathrm{cost}_{\mathrm{getdata}}\right). \tag{10}$$

$\mathrm{MAX}(N_1, N_2, \ldots, N_{\mathrm{NumFuntion}})$ and $(J-1)$ are all representatives of the number of decomposed data into components. They can be approximately equal.

Subtract formula (9) and formula (10):

$$\begin{aligned}&\mathrm{COST}_{\mathrm{MOFDP}} - \mathrm{COST}_{\mathrm{SMART}} \\ &\quad = \mathrm{NumFuntion} \times \mathrm{cost}_{\mathrm{getfun}} + \mathrm{NumNode} \\ &\qquad \times \mathrm{cost}_{\mathrm{jarm}}.\end{aligned} \tag{11}$$

The actual work found that NumNode is far greater than NumFuntion. But the value of $\mathrm{cost}_{\mathrm{getfun}}$ and $\mathrm{cost}_{\mathrm{jarm}}$ is not small; these two cannot be ignored. It can be seen that the crack cost of MOFDP algorithm is much larger than the cost of the SMART algorithm, mainly because of heavy workload of cracking the function in the library and identifying the true and false data.

6.3. Crack Probability and Simulation Experiment. In the case of random interference data in wireless sensor networks, the probability of obtaining correct data changes with the number of random interference nodes. The true sampled node is also a random node. Set JM to indicate the correct data when sending interference data. The probability is

$$P(\mathrm{JM}) = \frac{1}{1 + N_{\mathrm{jarm}}} \times \frac{1}{\mathrm{NumNode}}, \tag{12}$$

where N_{jarm} denotes the number of joining random interference nodes. In this paper the value of N_{jarm} is set as 1, and NumNode is the total number of subnodes.

In this paper, we use the idea of data fragmentation and synthesis to increase the security of true sampling data. The decomposition function library and the fusion recovery function library are used. The vector of numbers of solution vector components of decomposition function in the function library is $\{N_1, N_2, \ldots, N_q, \ldots, N_{\mathrm{NumFunction}}\}$, where N_q represents the number of components which is the result that the qth decomposition function $\mathrm{Dec}_q(D_i(t))$ decomposes the $D_i(t)$. Since the decomposition function and the fusion recovery function are one-to-one reciprocal operations, decomposition function can be cracked, but also can break the fusion recovery function. The probability that the decomposition function can be cracked is denoted as

$$[\text{Decs}, P] = \begin{bmatrix} \text{Dec}_1(D_i(t)) & \text{Dec}_2(D_i(t)) & \cdots & \text{Dec}_q(D_i(t)) & \cdots & \text{Dec}_{\text{NumFunction}}(D_i(t)) \\ P(\text{FunDec}_1) & P(\text{FunDec}_2) & \cdots & P(\text{FunDec}_q) & \cdots & P(\text{FunDec}_{\text{NumFunction}}) \\ \text{pcf}_1 & \text{pcf}_2 & \cdots & \text{pcf}_q & \cdots & \text{pcf}_{\text{NumFunction}} \end{bmatrix}, \tag{13}$$

where Decs(q) is the decomposition function library. NumFunction is the capacity of the function library, or, alternatively, the total number of decomposition functions. The channel is set up between the sampling node and the sink node to transmit the data in separate time. The eavesdropper cannot intercept the correct data at once. They have to intercept the component many times in the channel. The more the components that get decomposed, the smaller the probability of intercepting the correct data. In case the decomposition function is cracked, the probability in which the listener can intercept one component is $P(\text{GetCom}_i \mid \text{FunDec})$, where i is denoted as the ith component. The probability value is determined by the technology and conditions of the eavesdropper. Here we set it as p_{compon}.

$$P(\text{GetCom}_i \mid \text{FunDec}) = p_{\text{compon}}. \tag{14}$$

In this paper, we investigate the probability of capturing each component as equal probability. The probability that the eavesdropper intercepts all components after they break down the decomposition function is $P(\text{GetComs} \mid \text{FunDec})$. There are two steps for our method in this paper: the first step is to extract a function from the decomposition function library. The probability of extracting the function from the decomposition function library obeys uniform distribution. The probability of obtaining the function to implement decomposition is $P(\text{Getchosfun})$; then

$$P(\text{Getchosfun}) = \frac{1}{\text{NumFunction}}. \tag{15}$$

The second step is to decompose the real sampling data with the random function. According to the full probability formula, the probability of information exposure in the procedure of function decomposition protection is denoted as

$$\begin{aligned} P(\text{GetComs}) &= P(\text{Getchosfun}) \times \Bigg(P(\text{FunDec}_1) \\ &\times \prod_{i=1}^{i=N_1} P(\text{GetCom}_i \mid \text{FunDec}_1) + P(\text{FunDec}_2) \\ &\times \prod_{i=1}^{i=N_2} P(\text{GetCom}_i \mid \text{FunDec}_2) + \cdots \\ &+ P(\text{FunDec}_q) \times \prod_{i=1}^{i=N_q} P(\text{GetCom}_i \mid \text{FunDec}_q) \\ &+ \cdots + P(\text{FunDec}_1) \\ &\times \prod_{i=1}^{i=N_{\text{NumFunction}}} P(\text{GetCom}_i \mid \text{FunDec}_{\text{NumFunction}}) \Bigg) \\ &= \frac{1}{\text{NumFunction}} \times \left(\text{pcf}_1 \times p_{\text{compon}}^{N_1} + \text{pcf}_2 \right. \\ &\times p_{\text{compon}}^{N_2} + \cdots + \text{pcf}_q \times p_{\text{compon}}^{N_q} + \cdots \\ &\left. + \text{pcf}_{\text{NumFunction}} \times p_{\text{compon}}^{N_{\text{NumFunction}}} \right). \end{aligned} \tag{16}$$

In this paper, the key vector is promoted in the algorithm. The ith node and the sink node share the key vector $K_i = \{k_{(i,1)}, k_{(i,2)}, \ldots, k_{(i,N)}\}$. The probability of decryption key is

$$\begin{aligned} P(\text{Getkeys}) &= \frac{K_i}{\sum_{j=1}^{j=\text{NumFunction}} N_j} \times \frac{1}{K_i} \\ &= \frac{1}{\sum_{j=1}^{j=\text{NumFunction}} N_j}. \end{aligned} \tag{17}$$

The information exposure probability of our algorithm in this paper is

$$\begin{aligned} P &= P(\text{Getkeys}) \times P(\text{JM}) \times P(\text{GetComs}) \\ &= \frac{1}{\sum_{j=1}^{j=\text{NumFunction}} N_j} \times \frac{1}{1 + N_{\text{jarm}}} \times \frac{1}{\text{NumNode}} \\ &\times \frac{1}{\text{NumFunction}} \times \left(\text{pcf}_1 \times p_{\text{compon}}^{N_1} + \text{pcf}_2 \right. \\ &\times p_{\text{compon}}^{N_2} + \cdots + \text{pcf}_q \times p_{\text{compon}}^{N_q} + \cdots \\ &\left. + \text{pcf}_{\text{NumFunction}} \times p_{\text{compon}}^{N_{\text{NumFunction}}} \right) \\ &= \frac{1}{\sum_{j=1}^{j=\text{NumFunction}} N_j} \times \frac{1}{1 + N_{\text{jarm}}} \times \frac{1}{\text{NumNode}} \\ &\times \frac{1}{\text{NumFunction}} \times \sum_{q=1}^{q=\text{NumFunction}} \text{pcf}_q \times p_{\text{compon}}^{N_q}. \end{aligned} \tag{18}$$

If N_{jarm} is set to be 1, $N_{\text{jarm}} = 1$, formula (18) can be simplified as

$$P = \frac{\sum_{q=1}^{q=\text{NumFunction}} \text{pcf}_q \times p_{\text{compon}}^{N_q}}{2 \times \text{NumFunction} \times \text{NumNode} \times \sum_{j=1}^{j=\text{NumFunction}} N_j}. \quad (19)$$

The information exposure probability and the probability of cracking the function or intercepting the data have positive correlation.

The SMART algorithm and MOFDP algorithm both have the idea of chip integration. At the time of each data aggregation, SMART uses the same function to decompose and merge, while MOFDP uses decomposition and fusion of functions that are randomly extracted from the function library.

In order to further analyze and compare, based on the classical SMART algorithm, the MOFDP algorithm is implemented in this paper. In this section, we set the probability that the link level privacy is broken q to 0.03 from 0.01. We compare the percentage that private data is disclosed in the case of different number of slices J, the number of decomposition functions NumFun, and the number of components of the vector N.

For convenience of comparison, the probability of the crack function pcf is set to a constant. In this simulation experiment, pcf takes 0.01.

In the SMART algorithm, J stands for data slices, but only the $J - 1$ data slices are encrypted. In the MOFDP algorithm, N stands for data fragmentation, and all N data slices are encrypted. NumFun stands for function library capacity. Compare the data exposure rate in the $J - 1$ case of the SMART algorithm with the N of the MOFDP algorithm.

When J takes 4 in SMART, NumFun takes 3, N takes up to 3 in the MOFDP, and the comparison results between them are shown in Figure 2.

When J takes 3 in SMART, NumFun takes 2, N takes up to 2 in the MOFDP, and the comparison results between them are shown in Figure 3.

It is shown that the information exposure rate of MOFDP algorithm is obviously lower than that in the SMART algorithm when the q is greater than 0.022. With the increase of q, the gap between the SMART algorithm and the algorithm of this paper is getting bigger and bigger. With the increase of q, the information exposure rate of SMART algorithm increases greatly, but the information exposure rate of MOFDP algorithm increases slowly. As can be seen from Figure 3, the number of small pieces of this algorithm in this paper is very obvious advantages. The probability of information exposure of MOFDP is very low. The growth rate of SMART algorithm is raising curve, but the curve about MOFDP algorithm is very smooth.

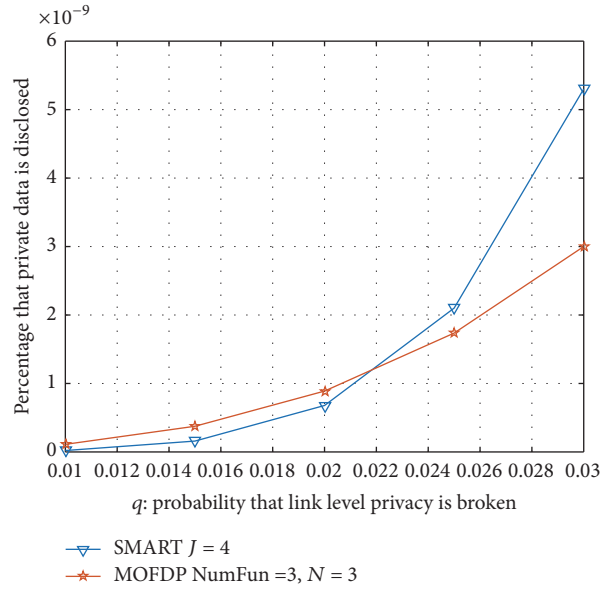

Figure 2: Comparison results when $J = 4$, NumFun = 3, $N = 3$.

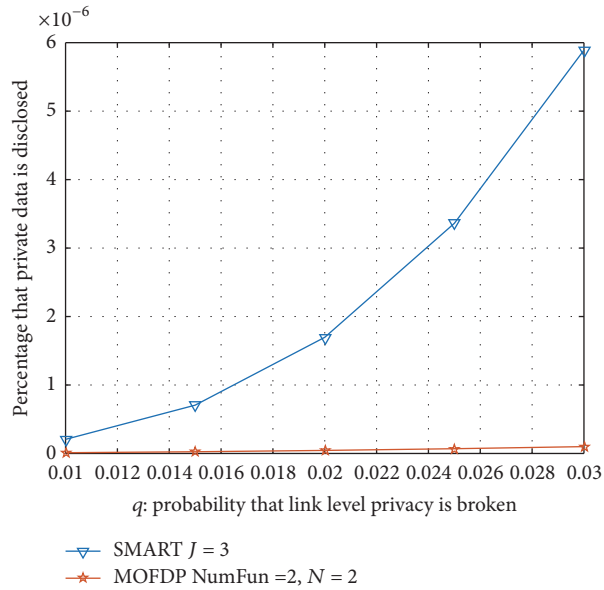

Figure 3: Comparison results when $J = 3$, NumFun = 2, $N = 2$.

7. Conclusions

With the rise of the Internet of things, the privacy protection of wireless sensor networks has become more important. In this paper, a multiorder data privacy protection algorithm is proposed based on the idea of SMART algorithm. In this paper, we introduce the interference code protection and adopt the idea of multiorder fusion to implement our proposed scheme. We target the difficulty and cost of cracking to improve effective privacy protection. The simulation results show that the proposed scheme has a better privacy protection function under low traffic.

Conflicts of Interest

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

Acknowledgments

This work was financially supported by the Project of Natural Science Foundation of Hainan Province in China (Grant no. 20166232 and Grant no. 617033), the Innovative Research Projects for Graduate Students of Hainan higher education institutions (Grant no. Hyb2017-06), the National Natural Science Foundation of China (Grant no. 61561017 and Grant no. 61462022), and the Open Project of State Key Laboratory of Marine Resource Utilization in South China Sea (Grant no. 2016013B).

References

[1] M. Conti, L. Zhang, S. Roy, R. Di Pietro, S. Jajodia, and L. V. Mancini, "Privacy-preserving robust data aggregation in wireless sensor networks," *Security and Communication Networks*, vol. 2, no. 2, pp. 195–213, 2009.

[2] H. E. Wenbo, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2045–2053, Anchorage, AK, USA, May 2007.

[3] W.-B. He, N. Hoang, X. Liu, K. Nahrstedt, and T. Abdelzaher, "iPDA: an integrity-protecting private data aggregation scheme for wireless sensor networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[4] W. He, X. Liu, H. Nguyen, and K. Nahrstedt, "A Cluster-Based Protocol to Enforce Integrity and Preserve Privacy in Data Aggregation," in *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS Workshops)*, pp. 14–19, Montreal, Canada, June 2009.

[5] M. M. Groat, W. Hey, and S. Forrest, "KIPDA: K-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the IEEE INFOCOM 2011*, pp. 2024–2032, April 2011.

[6] J. Girao, D. Westhoff, and M. Schneider, "CDA: concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, vol. 5, pp. 3044–3049, IEEE, May 2005.

[7] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 109–117, IEEE Computer Society, July 2005.

[8] R. Bista, K.-J. Jo, and J.-W. Chang, "A new approach to secure aggregation of private data in wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC 2009*, pp. 394–399, December 2009.

[9] R. Bista, K. Hee-Dae, and W. Chang J, "A new private data aggregation scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Computer and Information Technology, Cit 2010*, pp. 273–280, Bradford, UK, June 29, 2010.

[10] G. Yang, A.-Q. Wang, Z.-Y. Chen, J. Xu, and H.-Y. Wang, "An energy-saving privacy-preserving data aggregation algorithm," *Chinese Journal of Computers*, vol. 34, no. 5, pp. 792–800, 2011.

[11] L. Sen and Y. Geng, "Research on Precision Aggregation Privacy-preserving Algorithm in Wireless Sensor Networks," *Computer Technology and Development*, no. 9, pp. 139–142, 2013.

[12] S. Lu-sheng and Q. Xiao-lin, "Privacy-preserving data aggregation algorithm with integrity verification," *Computer Science*, vol. 40, no. 11, pp. 197–202, 2013.

[13] W. Tao-chun, L. Yong-long, and Z. Kai-zhong, "Fault-tolerant and privacy-preserving data aggregation algorithm in sensor networks," *Application Research of Computers*, vol. 31, no. 5, pp. 1499–1502, 2014.

[14] H. Bah and A. Lev, "k-anonymity based framework for privacy preserving data collection in wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 2, pp. 241–271, 2010.

[15] C. Zhang, C. Li, and J. Zhang, "A secure privacy-preserving data aggregation model in wearable wireless sensor networks," *Journal of Electrical and Computer Engineering*, vol. 2015, Article ID 104286, 9 pages, 2015.

[16] X. Cao, Z. Fu, and X. Sun, "A privacy-preserving outsourcing data storage scheme with fragile digital watermarking-based data auditing," *Journal of Electrical and Computer Engineering*, vol. 2016, Article ID 3219042, 7 pages, 2016.

[17] E. Mlaih and S. A. Aly, "Secure hop-by-hop aggregation of end-to-end concealed data in wireless sensor networks," in *Proceedings of the 2008 IEEE INFOCOM Workshops*, pp. 1–6, April 2008.